

درس : بینایی کامپیوتر

نام و نام خانوادگی: فاطمه توکلی

تمرین شماره ۰۲

۱) الگوریتم HoughLinesP به این صورت است که یک پیکسل تصادفی از تصویر لبه های ورودی، انتخاب می شود و انباشتگر Hough به روزرسانی می شود. اگر این به روزرسانی منجر به رای کافی برای یک خط معین  $L$  شود (یعنی اگر مقدار در ماتریس انباشتگر مربوط به  $L$  بالاتر از یک آستانه باشد)، جستجو با «راه رفتن» در هر دو جهت  $L$  انجام می شود تا بزرگترین قطعه خط در این راستا را پیدا کنیم، اگر به یک مرز تصویر برسیم، یا شکاف کافی بزرگ تشخیص داده شود، این جستجو متوقف می شود. تمام پیکسل های روی این خط از لیست پیکسل های ورودی حذف می شود و رای ها این خط در انباشتگر هم حذف می شود. به این ترتیب الگوریتم کارآمدتر و سریع تر می شود. و نقاط انتهایی خط، خروجی الگوریتم است.

انباشتگر: تعداد رای ها برای پارامترها مقدار انباشتگر هستند.

پارامترهای ورودی:

Image: تصویر منبع ۸ بیتی، تک کاناله، باینری

rho: وضوح فاصله انباشتگر بر حسب پیکسل

theta: وضوح یا تفکیک زاویه انباشتگر بر حسب رادیان

Threshold:

پارامتر آستانه انباشتگر، فقط آن خطوطی برگردانده می شوند که رای کافی ( $> \text{threshold}$ ) را بدست آورند.

minLineLength: حداقل طول خط. بخش های خط کوتاه تر از آن رد می شوند

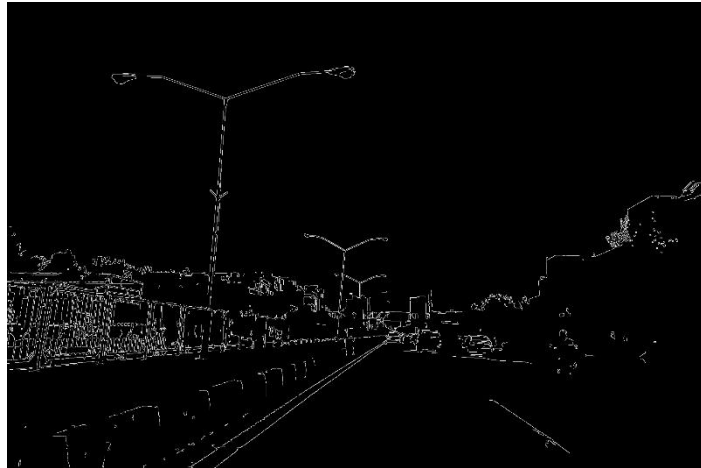
maxLineGap: حداکثر فاصله مجاز بین نقاط روی یک خط برای حفظ پیوستگی

پارامتر خروجی:

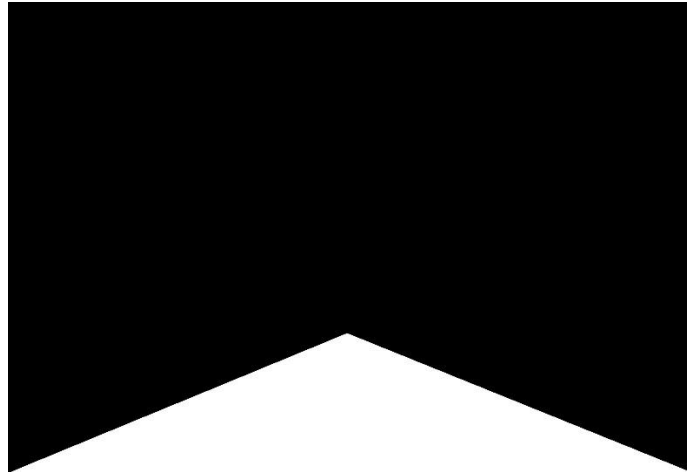
Lines:

بردار خروجی از خطها. هر خط به وسیله بردار ۴ عنصری  $(x1, y1, x2, y2)$  نمایش داده می شود، که  $(x1, y1)$  و  $(x2, y2)$  نقاط انتهایی خط تشخیص داده شده است.

•



• ماسک ایجاد شده:



•



- نتیجه نهایی به صورت زیر می باشد:



۳) در پیوست نتیجه آمده است

روشی که برای بهبود میتوان بکار برد این است که خط های سری قبل را ذخیره کرده و در صورت دیده شدن ناپیوستگی در خطوط تشخیص داده شده با خطوط قبلی پر کنیم.

۴) **Active countour**، **snack** ها را با ویژگی های تصاویر تطبیق می دهد. از تصاویر دو بعدی تک و چند کاناله پشتیبانی می کند. مارها می توانند متناوب (برای تقسیم بندی) یا دارای انتهای ثابت و/یا آزاد باشند.

طول مار خروجی برابر با مرز ورودی است. از آنجایی که تعداد نقاط ثابت است، باید مطمئن شد که مار اولیه دارای نقاط کافی برای ثبت جزئیات کانتور نهایی است.

پارامترهای ورودی:

image : (N, M) or (N, M, 3) ndarray

تصویر ورودی

snake : (N, 2) ndarray

مختصات اولیه مار(مرز اولیه). برای شرایط مرزی دوره ای، نقاط پایانی نباید تکراری باشد.

Alpha : float, optional

پارامتر شکل طول مار. مقادیر بالاتر باعث می شود مار سریعتر منقبض شود.

Beta : float, optional

پارامتر شکل صافی مار. مقادیر بالاتر مار را صاف تر می کند.

W\_line : float, optional

جذب به روشنایی را کنترل می کند. از مقادیر منفی برای جذب به سمت مناطق تاریک استفاده کنید.

W\_edge : float, optional

جذب لبه ها را کنترل می کند. از مقادیر منفی برای دفع لبه های مار استفاده کنید.

Gamma : float, optional

پارامتر صریح زمان گام.

Max\_px\_move : float, optional

حداکثر فاصله پیکسل برای حرکت در هر تکرار.

Max\_num\_iter : int, optional

حداکثر تکرار برای بهینه سازی شکل مار.

Convergence : float, optional

معیارهای همگرایی

Boundary\_condition : string, optional

شرایط مرزی برای کانتور. می تواند یکی از 'periodic'، 'free'، 'fixed'، 'free-fixed' یا 'fixed-free' باشد. 'periodic' دو انتهای مار را بهم متصل می کند. 'fixed' نقاط انتهایی را در جای خود نگه می دارد و 'free' اجازه حرکت آزاد به نقاط انتهایی را می دهد. 'free' و 'fixed' میتوانند در 'free-fixed' و 'fixed-free' با هم ترکیب شوند.

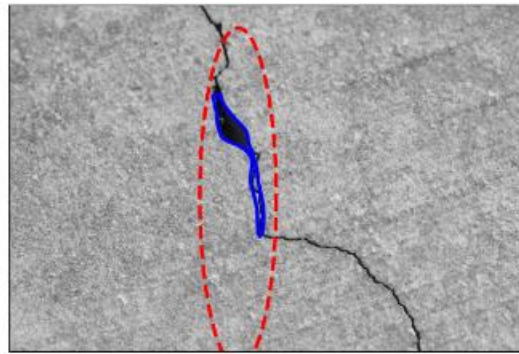
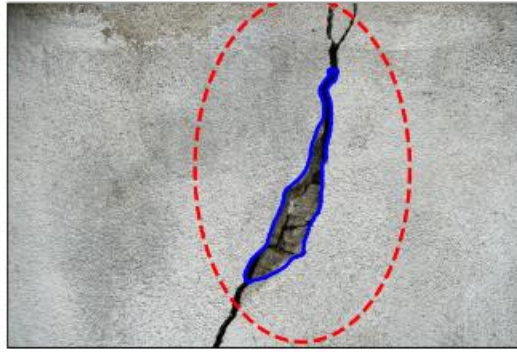
Coordinates : {'rc'}, optional

این گزینه فقط برای سازگاری باقی می ماند و هیچ تاثیری ندارد. مختصات باید در قالب ردیف-ستون تنظیم شوند.

پارامترهای خروجی:

snake : (N, 2) ndarray

مار بهینه شده، همان شکل پارامتر ورودی.



پایان