

به نام خدا

نام درس: مبانی بینایی

نام استاد: دکتر محمدی

نام و نام خانوادگی: نیوشا یقینی

شماره دانشجویی: 98522346

گزارش تمرین: HW3

تاریخ: 1403/01/25

سوال 1:

این سوال تئوری است.

(a) برای تصویری مانند $I(x, y)$ ، بردار گرادیان $\nabla I(x, y)$ را محاسبه کنید. (نوشتن روابط کافی است.)

بردار گرادیان یک تصویر $I(x, y)$ نشان دهنده نرخ تغییر شدت (روشنایی) در هر نقطه از تصویر است. بردار گرادیان در واقع یک میدان برداری است که از مشتقات جزئی x و y تشکیل شده است و اساساً جهت و میزان تغییر شدت را در هر مکان پیکسل توصیف می کند.

فرمول گرادیان به شرح زیر است.

• گرادیان تابع دوبعدی f به صورت زیر تعریف می شود:

$$\nabla f(x, y) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

بنابراین در مرحله اول برای محاسبه گرادیان باید مشتق x و y را به ازای هر پیکسل بدست آوریم.

فرمول مشتق گیری تصویر به شرح زیر است.

Finite diff.	$\frac{\partial I[x_m, y_n]}{\partial x} \approx \frac{I[x_m + \epsilon, y_n] - I[x_m - \epsilon, y_n]}{\epsilon}$	$\frac{\partial I[x_m, y_n]}{\partial y} \approx \frac{I[x_m, y_n + \epsilon] - I[x_m, y_n - \epsilon]}{\epsilon}$						
Kernel	<table><tr><td>-1</td><td>0</td><td>1</td></tr></table>	-1	0	1	<table><tr><td>-1</td></tr><tr><td>0</td></tr><tr><td>1</td></tr></table>	-1	0	1
-1	0	1						
-1								
0								
1								

بنابراین به تبع از فرمول بالا بردار گرادیان $\nabla I(x, y)$ برای هر پیکسل در لوکیشن (x, y) بصورت زیر تعریف می شود:

$$\nabla I(x, y) = \left[\frac{I(x+1, y) - I(x-1, y)}{2}, \frac{I(x, y+1) - I(x, y-1)}{2} \right]$$

(b) چرا محاسبه این بردار مفید می باشد؟

محاسبه بردار گرادیان یک تصویر برای کارهای مختلف پردازش تصویر و بینایی کامپیوتری مفید است. در ادامه چند دلیل برای مفید بودن آن نام می بریم:

تشخیص لبه:

اندازه گرادیان نشان دهنده نرخ تغییر شدت در تصویر است. تغییرات شدید در شدت روشنایی معمولاً در لبه های تصویر رخ می دهد. اگر بردار گرادیان محاسبه شود، می توان این لبه ها را با شناسایی مناطقی که اندازه گرادیان در آن ها زیاد است، تشخیص داد.

استخراج ویژگی:

بردار گرادیان اطلاعات ارزشمندی در مورد ساختار محلی تصویر ارائه می دهد. ویژگی هایی مانند گوشه ها، اتصالات و مرزهای بافت را می توان از اطلاعات گرادیان استخراج کرد.

تشخیص اشیاء:

در وظایف تشخیص و شناسایی شی، شناسایی ویژگی های متمایز بسیار مهم است. بردار گرادیان به استخراج ویژگی هایی کمک می کند که برای تمایز بین اشیاء مهم هستند.

جریان نوری:

با تخمین جریان نوری می توان حرکت اجسام در فریم های متوالی یک دنباله ویدیو را ردیابی کرد. از بردار گرادیان برای محاسبه جریان نوری با ردیابی تغییرات شدت از یک فریم به فریم دیگر، استفاده می شود.

بهبود تصویر:

اطلاعات گرادیان را می توان برای تکنیک های بهبود تصویر مانند شارپ کردن استفاده کرد، که در آن از بزرگی گرادیان برای تأکید بر لبه ها و جزئیات در تصویر استفاده می شود.

فشرده سازی تصویر:

اطلاعات گرادیان را می توان برای الگوریتم های فشرده سازی تصویر مورد سوء استفاده قرار داد. برای مثال، در تکنیک های فشرده سازی مبتنی بر گرادیان، گرادیان ها کوانتیزه و کدگذاری می شوند تا محتوای تصویر را به طور کارآمد نشان دهند.

ثبت تصویر:

تکنیک‌های مبتنی بر گرادیان اغلب در ثبت تصویر استفاده می‌شوند، جایی که هدف، تراز کردن چندین تصویر از یک صحنه است که از دیدگاه‌های مختلف یا در زمان‌های مختلف گرفته شده‌اند. اطلاعات گرادیان به تخمین پارامترهای تبدیل مورد نیاز برای ثبت کمک می‌کند.

دید استریو:

در کارهای بینایی استریو یا تخمین عمق، اطلاعات گرادیان می‌تواند برای محاسبه اختلاف بین نقاط متناظر در جفت‌های تصویر استریو استفاده شود، که سپس برای تخمین عمق اشیاء در صحنه استفاده می‌شود.

به طور کلی، بردار گرادیان اطلاعات مهمی در مورد ساختار محلی و ویژگی‌های یک تصویر ارائه می‌کند و آن را به ابزاری اساسی برای پردازش تصویر مختلف و برنامه‌های بینایی کامپیوتری تبدیل می‌کند.

(c) *اندازه گرادیان تعریف شده روی صفحه تصویر (x, y) محاسبه کنید. (نوشتن روابط کافی است.)*

بزرگی گرادیان که اغلب به صورت $|\nabla|$ نشان داده می‌شود، قدرت تغییر شدت را در هر مکان پیکسل در تصویر، نشان می‌دهد.

فرمول اندازه گرادیان به شرح زیر است.

$$M(x, y) = \|\nabla f\| = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2} \approx |g_x| + |g_y| \quad \bullet \text{ اندازه گرادیان}$$

این فرمول بزرگی گرادیان را در هر مکان پیکسل در تصویر ارائه می‌دهد که نشان دهنده قدرت تغییر شدت در آن نقطه است.

بنابراین به تبع از فرمول بالا اندازه گرادیان برای هر پیکسل بصورت زیر تعریف می‌شود:

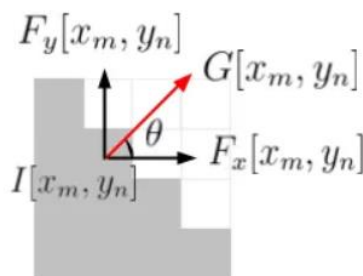
$$|\nabla I(x, y)| = \sqrt{\left(\frac{I(x+1, y) - I(x-1, y)}{2}\right)^2 + \left(\frac{I(x, y+1) - I(x, y-1)}{2}\right)^2}$$

(d) جهت گرادیان تعریف شده روی صفحه تصویر (x, y) را محاسبه کنید. (نوشتن روابط کافی است.)

جهت گرادیان که اغلب با θ نشان داده می‌شود، جهت‌گیری تغییر شدت روشنایی را در هر مکان پیکسل در تصویر نشان می‌دهد.

فرمول جهت گرادیان به شرح زیر است.

• جهت گرادیان $\alpha(x, y) = \text{dir}(\nabla f) = \text{atan2}(g_y, g_x)$



این فرمول زاویه (جهت) بردار گرادیان را در هر پیکسل تصویر، بر حسب رادیان نسبت به محور x مثبت محاسبه می‌کند و درواقع جهت‌گیری تغییر شدت روشنایی را در آن نقطه نشان می‌دهد. نتیجه معمولاً در محدوده $[-\pi, \pi]$ است.

بنابراین به تبع از فرمول بالا اندازه گرادیان برای هر پیکسل بصورت زیر تعریف می‌شود:

$$\theta(x, y) = \text{atan2}\left(\frac{I(x, y + 1) - I(x, y - 1)}{2}, \frac{I(x + 1, y) - I(x - 1, y)}{2}\right)$$

(e) نحوه استفاده از بردار گرادیان را در آشکارساز لبه **Canny** توضیح دهید. (مراحل اصلی آن و مزایای آن نسبت به رویکردهای جایگزین را بنویسید.)

لبه یاب **Canny** یکی از پرکاربردترین و موفق‌ترین روشهای لبه‌یابی است، که از اطلاعات گرادیان برای شناسایی لبه‌ها در یک تصویر استفاده می‌کند. این شامل چندین مرحله است و بردار گرادیان نقش مهمی در عملکرد آن ایفا می‌کند.

این لبه یاب از 4 گام اساسی تشکیل می‌شود:

- هموار کردن تصویر با استفاده از فیلتر گاوسی:
صاف کردن و هموارسازی تصویر اولین گام است که باعث کاهش نویز در تصویر می‌شود. معمولاً با اعمال یک فیلتر گاوسی روی تصویر به دست می‌آید. صاف کردن به حذف نویز با فرکانس بالا کمک می‌کند و منجر به تشخیص کاذب لبه می‌شود.
- محاسبه گرادیان:
پس از صاف کردن تصویر، گرادیان شدت روشنایی تصویر محاسبه می‌شود تا مقدار و جهت تغییر شدت در هر مکان پیکسل مشخص شود. (معمولاً از عملگر Sobel برای محاسبه گرادیان های افقی و عمودی استفاده می‌شود).
- حذف مقادیر غیربیشینه:
در این مرحله از الگوریتم با حذف اندازه های غیرحداکثری لبه های بالقوه، شناسایی می‌شوند. این شامل اسکن در امتداد جهت گرادیان و حفظ تنها حداکثرهای محلی در بزرگی گرادیان است. پیکسل‌هایی که حداکثر محلی نیستند، حذف می‌شوند (بر روی صفر تنظیم می‌شوند).
- آستانه گذاری دو مرحله‌ای:
در نهایت، پیکسل های لبه، بر اساس آستانه گذاری (Thresholding) انتخاب می‌شوند. دو مقدار آستانه استفاده می‌شود: یک آستانه بالا (T-high) و یک آستانه پایین (T-low). پیکسل هایی با اندازه گرادیان بالاتر (T-high) به عنوان پیکسل های "لبه قوی" در نظر گرفته می‌شوند، در حالی که پیکسل های بین (T-low) و (T-high) به عنوان پیکسل های "لبه ضعیف" در نظر گرفته می‌شوند. پیکسل های "لبه ضعیف" تنها در صورتی به عنوان لبه های اصلی شناخته خواهند شد که به پیکسل های لبه قوی متصل باشند.

مزایای آشکارساز لبه Canny نسبت به روش های جایگزین:

- مکان یابی دقیق لبه:
آشکارساز لبه Canny به دلیل توانایی خود در بومی سازی دقیق لبه ها در یک تصویر، به لطف مرحله "حذف مقادیر غیربیشینه" شناخته شده است.
- مقاوم در برابر نویز:
با اجرای هموارسازی گاوسی به عنوان مرحله اولیه، آشکارساز لبه Canny در برابر نویز مقاوم است و نقشه های لبه تمیزتری را در مقایسه با روش های دیگر تولید می‌کند.

- آستانه قابل تنظیم:

استفاده از دو مقدار آستانه در مرحله "آستانه گذاری دو مرحله‌ای" امکان انعطاف پذیری در تشخیص لبه را فراهم می‌کند.

- لبه‌های عرض تک پیکسل:

آشکارساز لبه Canny معمولاً لبه‌های عرض تک پیکسلی تولید می‌کند که در بسیاری از کاربردها مانند تشخیص اشیا و استخراج ویژگی مطلوب است.

به این معنا که لبه‌های شناسایی شده توسط الگوریتم معمولاً دارای عرض تنها یک پیکسل در نقشه لبه به دست آمده هستند. به عبارت دیگر، لبه‌ها با یک خط باریک از پیکسل‌ها نشان داده می‌شوند، جایی که هر پیکسل مربوط به محل دقیق لبه در تصویر اصلی است.

(f) از عملگر لاپلاسین می‌توان برای تشخیص لبه استفاده کرد. اما غالباً در عمل برای تشخیص لبه از همان عملگرهای Sobel و Canny استفاده می‌شود. چرا عملگر لاپلاسین عملگر خوبی برای تشخیص لبه نیست؟ (3 دلیل ذکر کنید.)

در حالی که عملگر Laplacian را میتوان برای تشخیص لبه استفاده کرد، در عمل در مقایسه با عملگرهایی مانند Sobel و Canny به طور معمول استفاده نمی‌شود. چند دلیل برای این وجود دارد:

- حساسیت به نویز:

عملگر لاپلاسین به نویز در تصویر بسیار حساس است. از آنجایی که مشتق دوم شدت تصویر را محاسبه می‌کند، نویز را تقویت می‌کند که منجر به تشخیص لبه‌های نادرست و یک نقشه لبه نویزدار می‌شود. در مقابل، عملگرهایی مانند Sobel و Canny معمولاً یک مرحله هموارسازی (مانند فیلتر گاوسی) را برای کاهش نویز قبل از تشخیص لبه در نظر می‌گیرند که باعث می‌شود در تصاویر نویزدار بهتر عمل کنند.

- پاسخ‌های دو لبه (Double Edge Responses):

"پاسخ‌های دو لبه"، که به عنوان "پاسخ‌های لبه‌های چندگانه" یا "دو برابر شدن لبه" نیز شناخته می‌شوند، زمانی رخ می‌دهند که یک الگوریتم تشخیص لبه بیش از یک یال را برای یک لبه موجود در تصویر اصلی شناسایی کند. این پدیده به ویژه در روش‌های تشخیص لبه که از عملگرهای مبتنی بر مشتق دوم استفاده می‌کنند، رایج است. این اتفاق به فرآیند تشخیص لبه پیچیدگی می‌افزاید و ممکن است به مراحل پس از پردازش اضافی نیاز داشته باشد.

عملگر لاپلاسین نیز مشتق دوم شدت تصویر را محاسبه می‌کند که نواحی تغییر شدت سریع را برجسته می‌کند. در تشخیص لبه، عملگر لاپلاسین اغلب برای یافتن نقاطی که شدت آن به سرعت تغییر می‌کند، استفاده می‌شود که نشان

دهنده وجود لبه ها است. با این حال، عملگر لاپلاسین یک آشکارساز صفر-تقاطع است، به این معنی که تلاقی های صفر را در مشتق دوم، مربوط به مناطقی که شدت از افزایش به کاهش یا بالعکس (شیب مثبت و شیب منفی) تغییر می کند، تشخیص می دهد. در نتیجه، ممکن است لبه ها را در هر دو طرف یک لبه تشخیص دهد که منجر به پاسخ های دو لبه می شود.

- مکان یابی ضعیف:

عملگر لاپلاسین تمایل به محلی سازی ضعیف لبه ها دارد، به خصوص برای لبه های ضخیم یا لبه هایی که تغییرات شدت روشنائی تدریجی دارند. این می تواند منجر به تار شدن یا نادقیق بودن نقشه های لبه شود و محلی سازی دقیق لبه ها در تصویر را دشوار کند. در مقایسه، اپراتورهایی مانند Sobel و Canny اغلب با محاسبه دقیق بزرگی و جهت گرادیان، محلی سازی بهتر لبه را ارائه می دهند و امکان تشخیص دقیق تر لبه را فراهم می کنند.

سوال 2:

این سوال عملی است و در اینجا گزارش آن آورده شده است.

(a) تصویر **1.jpg** و **2.jpg** را از پوشه‌ی **Q2** بخوانید. تبدیل فوری هر تصویر را محاسبه کنید و دامنه و فاز هر تصویر را نمایش دهید.

ابتدا تصاویر را میخوانیم.

تصویر 1:



تصویر 2:



یادآوری تبدیل فوری:

تبدیل فوری یک سیگنال ورودی را به صورت مجموعی از جملات سینوسی تجزیه می‌کند. فرمول های تبدیل فوری نیز به شرح زیر هستند.

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{+j2\pi(ux/M + vy/N)}$$

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

$$Magnitude = |F(u, v)| = \sqrt{Re^2(u, v) + Im^2(u, v)}$$

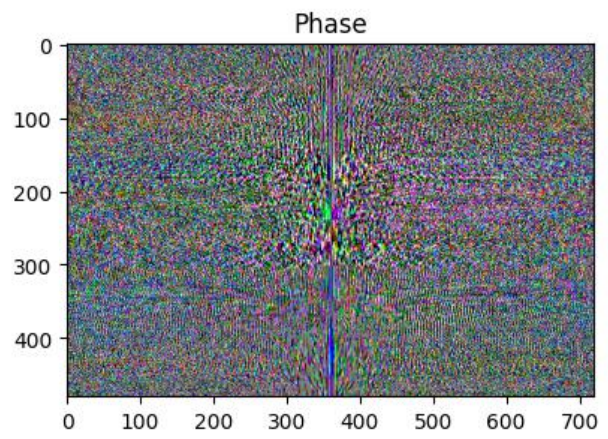
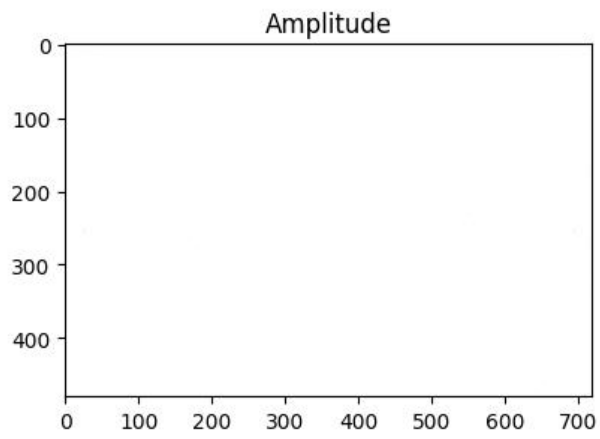
$$Phase = \varphi(u, v) = atan2(Im(u, v), Re(u, v))$$

برای محاسبه تبدیل فوری یک تصویر و نمایش دامنه و فاز آن، می‌توان از الگوریتم تبدیل فوری سریع (FFT) ارائه شده توسط کتابخانه‌هایی مانند NumPy و OpenCV استفاده کرد. همچنین می‌توان از تابع `fftpack` که در کتابخانه `scipy` ارائه شده نیز استفاده کرد. (هر دو مورد پیاده سازی شده اند).

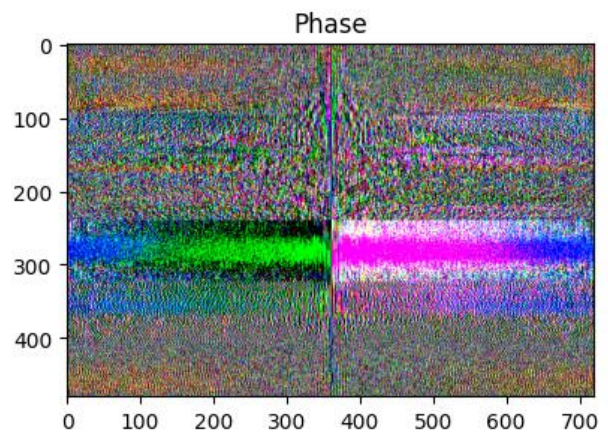
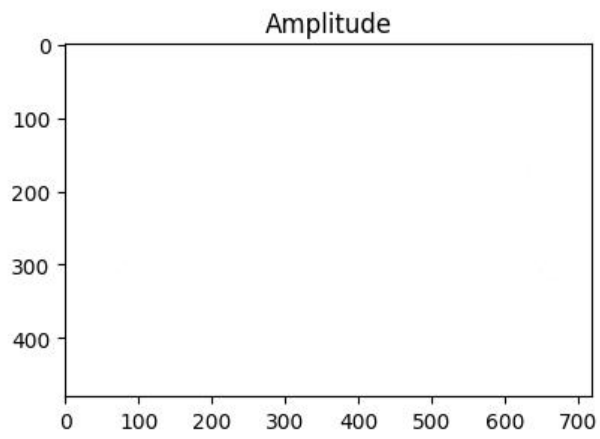
• مرحله 1:

در مرحله اول با استفاده از دو روش بالا از تصویر رنگی تبدیل فوری گرفتیم و در قسمت دامنه چیزی نمایش داده نشد، این به آن دلیل است که طیف دامنه نشان دهنده بزرگی ضرایب تبدیل فوری است که محتوای فرکانس تصویر را نشان می‌دهد. هنگامی که تبدیل فوری یک تصویر رنگی را محاسبه می‌کنیم، اساساً تبدیل فوری را برای هر کانال رنگی به طور جداگانه محاسبه می‌کنیم، که منجر به ایجاد سه طیف دامنه جداگانه (یکی برای هر کانال رنگی) می‌شود. و دامنه ممکن است به خوبی نمایش داده نشود زیرا اساساً بزرگی ترکیبی سه کانال رنگی را نشان می‌دهد. این می‌تواند به مقادیری منجر شود که برای نمایش صحیح در محدوده رنگی معمولی 8 بیتی (0-255) بسیار بزرگ هستند. برای پرداختن به این موضوع، می‌توانید قبل از محاسبه تبدیل فوری و تجسم طیف دامنه آن، طیف دامنه هر کانال رنگی را به طور جداگانه تجسم کنید یا تصویر رنگی را به مقیاس خاکستری تبدیل کنید.

نتیجه تصویر 1:



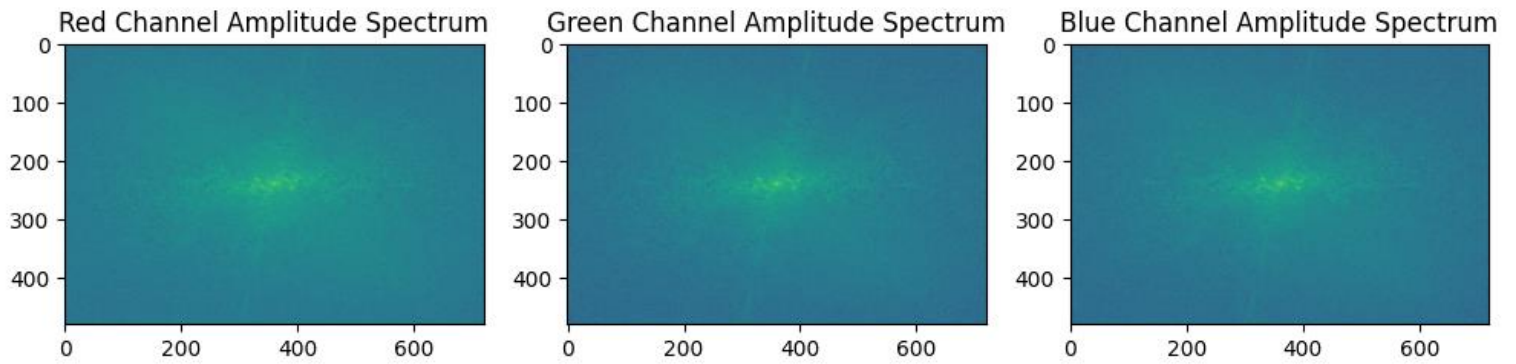
نتیجه تصویر 2:



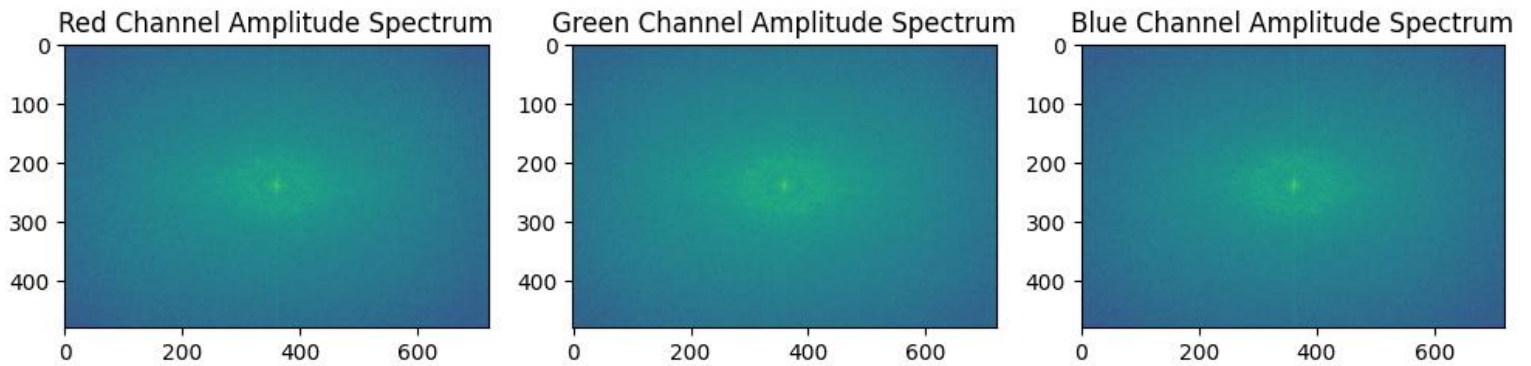
• مرحله 2:

در مرحله بعد تبدیل فوریه را برای هر یک از کانال های رنگی تصویر محاسبه کردیم.

نتیجه تصویر 1:



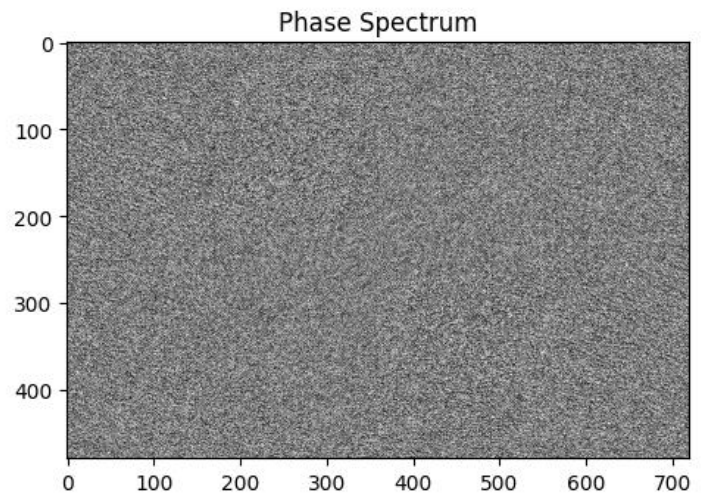
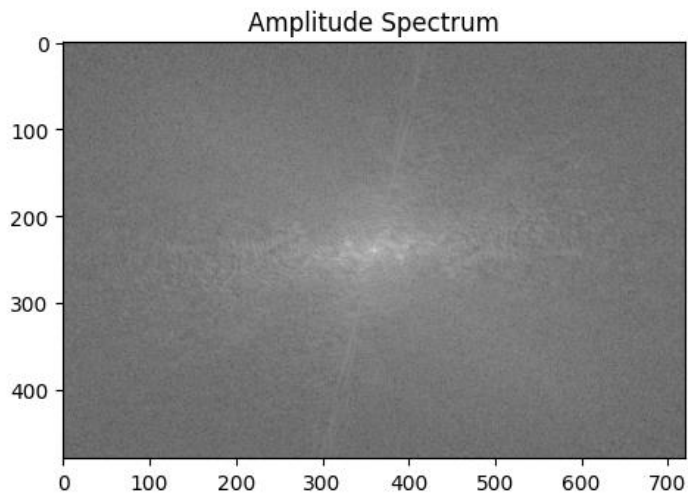
نتیجه تصویر 2:



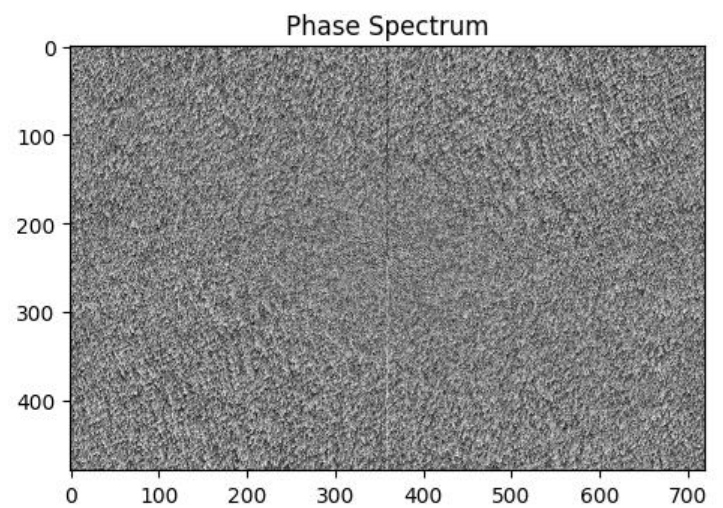
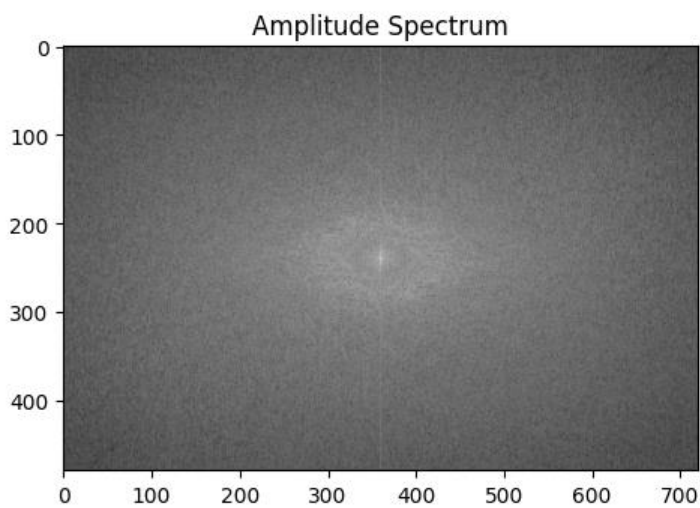
• مرحله 3:

و در انتها نیز تبدیل فوریه را برای تصویر در مقیاس خاکستری (gray scale) انجام دادیم.

نتیجه تصویر 1:



نتیجه تصویر 2:



تحلیل نتیجه:

تجزیه و تحلیل دامنه و پاسخ های فازی تصویر در مقیاس خاکستری به درک محتوای فرکانس و تغییرات فضایی در تصویر کمک می کند.

در زمینه تحلیل فوریه و تجسم طیف دامنه، "نقاط نور" به طور معمول فرکانس های پایین را نشان می دهند، در حالی که "مناطق تاریک" با فرکانس های بالا مطابقت دارند.

همچنین دقت شود که فرکانس 0 را در نمایش دامنه در وسط تصویر قرار می دهیم.

فرکانس های پایین:

اجزای فرکانس پایین در یک تصویر با الگوها یا ساختارهای صاف و به آرامی متغیر مطابقت دارند.

این مؤلفه های فرکانس پایین، ویژگی های کلی تصویر، مانند روشنایی کلی، گرادیان های در مقیاس بزرگ، و خطوط گسترده را ثبت می کنند. در حوزه فوریه، فرکانس های پایین با ضرایب نزدیک به مرکز طیف نمایش داده می شوند. [3]

فرکانس های بالا:

اجزای فرکانس بالا در یک تصویر نشان دهنده نوسانات یا تغییرات سریع یا جزئیات دقیق هستند. این اجزای فرکانس بالا ویژگی های محلی، لبه های تیز، جزئیات بافت و الگوهای ظریف را ثبت می کنند. در حوزه فوریه، فرکانس های بالا با ضرایب دورتر از مرکز طیف نشان داده می شوند. [3]

بنابراین با توجه به توضیحات بالا برای تصویر 1، یعنی گورخر است، بنظر می رسد که تغییرات شدت روشنایی افقی بصورت smooth یا غیر شدید هستند، و در بقیه ی نقاط به نسبت حالت افقی در میانه تصویر تغییرات شدت بیشتری دارند.

و در مورد تصویر 2، یعنی پلنگ، به نظر می رسد که تغییرات شدت ابتدا کم اما دوباره زیاد می شود، و دوباره کم، به نوعی انگار تغییرات روی خود پلنگ کم و در مرز پلنگ و اطراف زیاد و دوباره کم می شود.

نکته: cell های اضافه ای در کدها برای نمایش تصاویر بالا گذاشته شده است.

(b) جای فاز دو تصویر را عوض کنید و تبدیل فوریه معکوس بگیرید. سپس تصاویر حاصل را نمایش دهید. از این آزمایش چه نتیجه ای می گیرید؟

برای انجام این قسمت، مراحل زیر را انجام می دهیم:

- محاسبه تبدیل فوریه 2 تصویر و بدست آوردن فاز و دامنه آنها
- برای تصویر 1، دامنه خودش و فاز تصویر 2 را ترکیب می کنیم و به تابع معکوس فوریه گیر می دهیم.
- برای تصویر 2، دامنه خودش و فاز تصویر 1 را ترکیب می کنیم و به تابع معکوس فوریه گیر می دهیم.

تغییر فاز دو تصویر در حالی که دامنه آنها را بدون تغییر نگه داریم، منجر به اثراتی در پردازش تصویر می شود. که در ادامه به آن ها اشاره کرده ایم:

انتقال بافت:

با جابجایی اطلاعات فاز بین دو تصویر، امکان انتقال ویژگی های بافت یک تصویر به تصویر دیگر وجود دارد. این می تواند در برنامه هایی مانند انتقال سبک، که در آن بافت یک تصویر به محتوای تصویر دیگر اعمال می شود، مفید باشد.

فیوژن تصویر:

تعویض فاز بین تصاویر می تواند برای تکنیک های ترکیب تصویر استفاده شود. با ترکیب دامنه یک تصویر با فاز تصویر دیگر، می توانید تصویر جدیدی ایجاد کنید که جزئیات ساختاری هر دو تصویر ورودی را حفظ کند.

جلوه های بصری:

تغییر فاز تصاویر می تواند منجر به جلوه های هنری یا اغوجاج بصری شود. به عنوان مثال، تعویض فاز تصاویر طبیعی با نویز تصادفی می تواند تصاویر انتزاعی یا سورئالیستی ایجاد کند.

استحکام در برابر فشرده سازی:

در برخی موارد، اصلاح فاز یک تصویر با حفظ دامنه آن می تواند تصویر را در تکنیک های فشرده سازی قوی تر کند. این به این دلیل است که ادراک انسان نسبت به تغییرات در ساختار تصویر (قدرت) حساس تر از تغییرات فاز تصویر است.

source 1



amplitude source 1



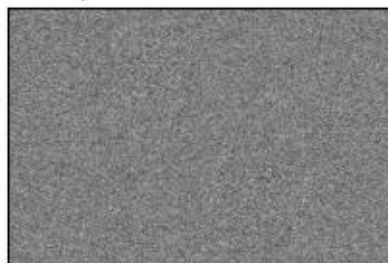
phase source 2



new image 1



phase source 1



source 2



amplitude source 2



new image 2



سوال 3:

این سوال عملی است و در اینجا گزارش آن آورده شده است.

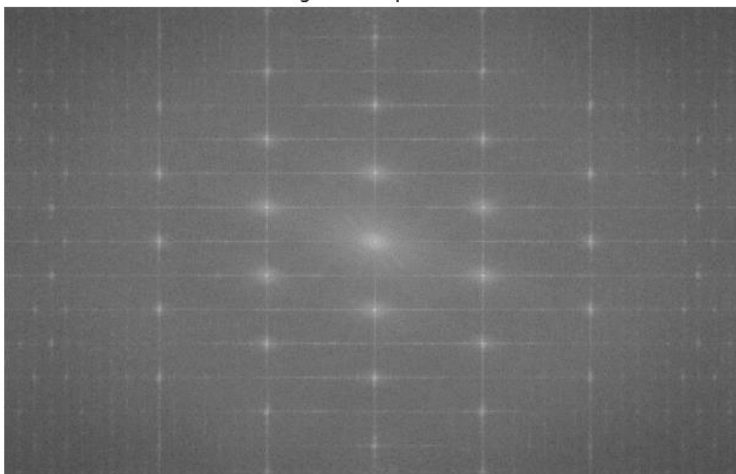
(a) ابتدا تصویر *saffron.jpg* را بخوانید و نویز تصویر را با تبدیل *FFT* حذف کنید. در حذف نویز از تمام مراحل خروجی گرفته و رسم کنید و علت کار خود را نیز توضیح دهید.

برای حذف نویز با تبدیل فوریه (FFT) مراحل زیر باید طی شوند:

- اعمال FFT:

FFT تصویر ورودی را محاسبه می‌کنیم تا دامنه فرکانس و فاز آن را بدست آوریم.

Magnitude Spectrum

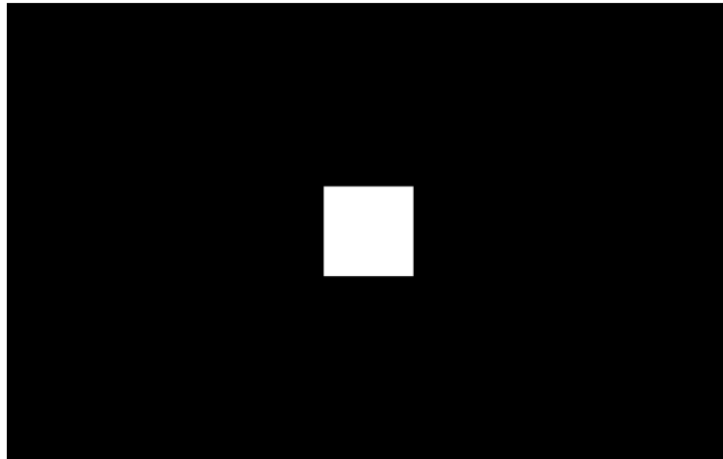


- فیلتر کردن:

فرکانس های نویز را در حوزه فرکانس شناسایی و حذف می‌کنیم. برای اینکار یک **threshold** در نظر می‌گیریم و به کمک آن فرکانس های بالا (که احتمال نویز بودن آنها زیاد است) را حذف می‌کنیم در حالی که اجزای فرکانس پایین حفظ می‌شود.

نکته: انتخاب **threshold** نامناسب باعث کاهش کیفیت تصویر نیز ممکن است بشود، بنابراین اگر عدد خیلی کوچکی انتخاب شود بخش های زیادی از خود تصویر را هم از دست می‌دهیم، و اگر عدد خیلی بزرگی انتخاب شود، نویز ها را کاور نمی‌کند. (عدد های 50، 60، 70، 80، 90، 100، 120 و 150 برای **threshold** تست شدند و 100 انتخاب شد.)

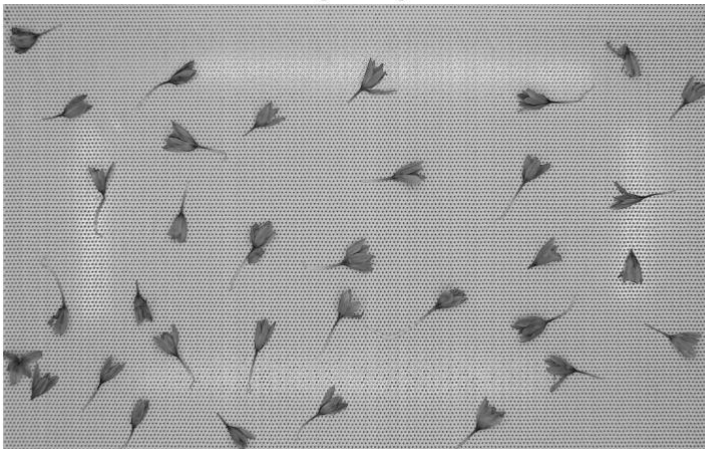
Filter Mask



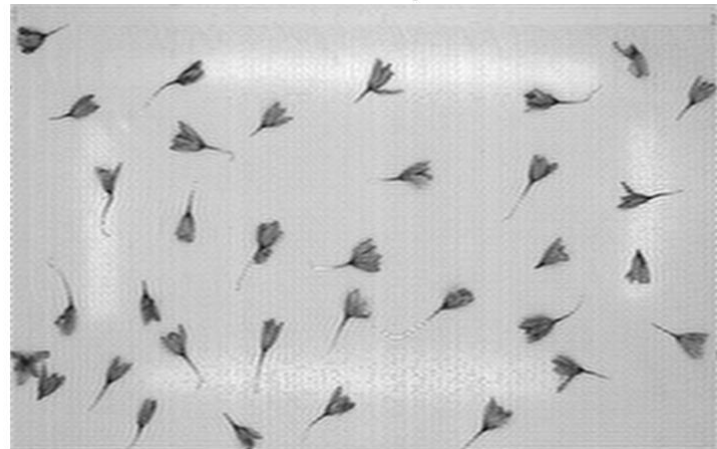
- FFT معکوس:

برای به دست آوردن تصویری که نویز آن حذف شده یک FFT معکوس انجام می‌دهیم. (نکته: اعمال تبدیل فوریه روی یک تصویر و سپس اعمال تبدیل فوریه رویش، به ما تصویر اصلی را خواهد داد.)

Original Image



Denoised Image



(b) لبه یاب Canny را بر روی خروجی مرحله a اجرا کنید. برای لبه یاب از توابع آماده استفاده کنید.
تمام پارامترهای تابع که مقداردهی می‌شوند، با ذکر دلیل توضیح داده شوند. مطلوب است در خروجی این مرحله فقط گلهای زعفران بمانند.

در لبه یاب Canny 2 آستانه داریم، که در اینجا نیز ورودی های تابع ما هستند.

• آستانه 1 (ورودی اول):

این پارامتر مقدار "آستانه پایین" را تعریف می‌کند. این اولین آستانه مورد استفاده در مرحله "آستانه گذاری دو مرحله‌ای" الگوریتم تشخیص لبه Canny است. هر پیکسل دارای اندازه گرادیان شدت روشنایی است، پیکسل ای که اندازه بالاتر از آستانه 1 داشته باشد، به عنوان پیکسل لبه قوی در نظر گرفته می‌شود.

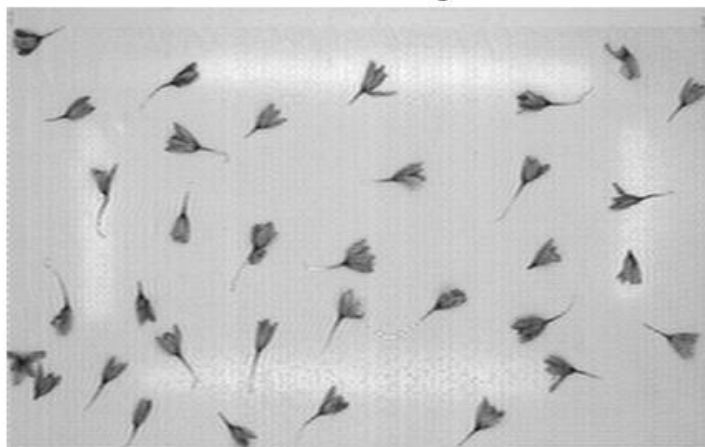
• آستانه 2 (ورودی دوم): 200

این پارامتر مقدار "آستانه بالا" را در مرحله "آستانه گذاری دو مرحله‌ای" مشخص می‌کند. این دومین آستانه مورد استفاده در این مرحله از الگوریتم تشخیص لبه Canny است. هر پیکسل ای که اندازه گرادیان شدت روشنایی آن کمتر از آستانه 2 باشد، حذف می‌شود، مگر اینکه به یک پیکسل لبه قوی متصل باشد.

آشکارساز لبه Canny بدین صورت کار می‌کند که ابتدا فیلتر گاوسی را روی تصویر ورودی برای "هموار کردن تصویر" اعمال می‌کند و نویزها را تا حد امکان حذف می‌کند، سپس از "محاسبه گرادیان" روی تصویر برای یافتن اندازه و جهت لبه استفاده می‌کند. سپس، برای نازک کردن لبه‌ها، "حذف مقادیر غیربیشینه" که به آن "مهار غیر حداکثری" هم می‌گویند را اعمال می‌کند و در نهایت پیوند لبه‌ها را با "آستانه گذاری دو مرحله‌ای" انجام می‌دهد.

مقادیر آستانه (آستانه 1 و آستانه 2) تعیین می‌کنند که کدام لبه ها قوی و کدام ضعیف در نظر گرفته شوند و از این رو بر نتایج تشخیص لبه نهایی تأثیر می‌گذارند. تنظیم این آستانه ها می‌تواند تأثیر قابل توجهی بر لبه های شناسایی شده داشته باشد و امکان تنظیم دقیق فرآیند تشخیص لبه بر اساس ویژگی های خاص تصویر ورودی و خروجی مورد نظر را فراهم کند.

Denoised Image



Canny Edges



(c) از تصویر بدست آمده (مرحله قبل b) گرادیان بگیرید و با استفاده از تابع $\arctan 2$ جهت گرادیان‌های بدست آمده را محاسبه کنید.

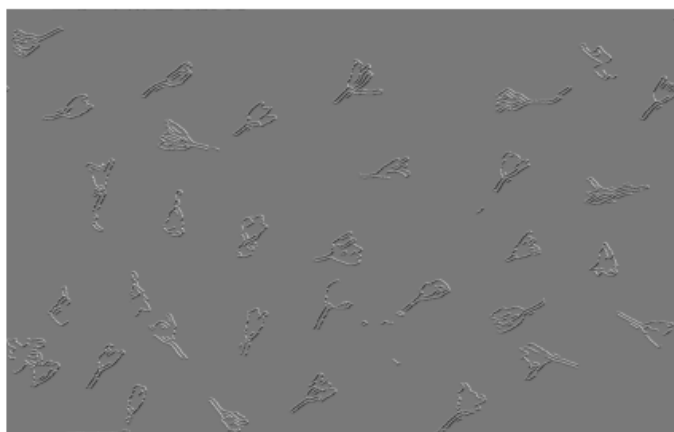
برای محاسبه گرادیان و جهت آن برای تصویر بدست آمده از بخش قبل باید 2 مرحله زیر انجام شود:

- ابتدا برای محاسبه گرادیان تصویر در جهت X و Y ابتدا از یک فیلتر گرادیان، مانند Sobel استفاده می‌کنیم.
- سپس مقدار و جهت شیب‌ها را به ترتیب با استفاده از توابع قدر مطلق و $\arctan 2$ محاسبه می‌کنیم.

Gradient Magnitude



Gradient Direction



(d) امتیازی: با استفاده از جهت گرادیان‌های بدست آمده، راه حلی برای بدست آوردن نقطه برش ساقه از گلبرگ ارائه دهید.

برای به دست آوردن نقطه تقاطع ساقه از گلبرگ در تصویری از گل‌ها با استفاده از جهت گرادیان‌های به دست آمده، می‌توان از تکنیک‌هایی مانند تشخیص لبه و تبدیل Hough استفاده کرد. ولی کل یک رویکرد جامع وجود دارد:

• Edge Detection:

از یک الگوریتم تشخیص لبه مانند Canny edge Detector برای تشخیص لبه‌ها در تصویر استفاده کنیم. این به شناسایی مرزهای گلبرگ و ساقه کمک می‌کند.

• Hough Transform:

از تبدیل Hough برای تشخیص خطوط در تصویر تشخیص لبه استفاده کنیم. با این کار می‌توان خطوط مربوط به گلبرگ و ساقه را پیدا کرد.

- تقاطع خط:

نقطه تلاقی خطوطی که گلبهگ ها و ساقه را نشان می‌دهند را پیدا می‌کنیم. این نقطه تلاقی را می‌توان نقطه تلاقی ساقه با گلبهگ در نظر گرفت.

یادآوری Hough Transform:

ایده اصلی تبدیل Hough بر تغییر فضا و رای‌گیری است، هر خط در فضای (x, y) معادل با یک نقطه در فضای (m, c) است.

$$y = mx + c$$

Hough Transform تکنیکی است برای تشخیص اشکال هندسی ساده مانند خطوط، دایره ها و بیضی ها در یک تصویر استفاده می‌شود. البته از لحاظ محاسباتی گران محسوب می‌شود.

نحوه عملکرد Hough Transform آن نیز به شرح زیر است:

- تشخیص لبه:

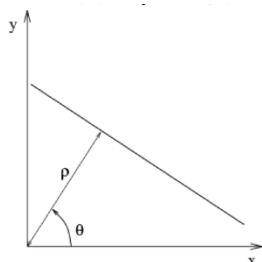
اولین مرحله در Hough Transform معمولاً تشخیص لبه‌ها در تصویر با استفاده از تکنیک‌هایی مانند آشکارساز لبه Canny است. این مرحله پیکسل‌هایی را در تصویر مشخص می‌کند که در آنها تغییر قابل توجهی در شدت وجود دارد که اغلب با مرزهای اشیاء مطابقت دارد.

- فضای پارامتر:

در Hough Transform، هر نقطه در تصویر لبه‌ای (نتیجه مرحله قبل) با یک خط پتانسیل در تصویر اصلی مطابقت دارد. برای هر نقطه (x, y) در تصویر لبه‌ای، تمام خطوط ممکنه را که از آن نقطه عبور می‌کنند در نظر می‌گیریم.

- فضای Hough:

پارامترهای یک خط در یک سیستم مختصات دکارتی دوبعدی معمولاً شیب m و نقطه قطع c (برای خطی که با معادله $y = mx + c$ نشان داده می‌شود) است. با این حال، شیب m برای خطوط عمودی نامحدود می‌شود. برای جلوگیری از این امر، می‌توانیم خطوط را در یک فضای پارامتر متفاوت، به نام فضای Hough نشان دهیم. در فضای هاف، هر خط در فضای دکارتی مربوط به یک نقطه (ρ, θ) در فضای هاف است، جایی که ρ نشان دهنده فاصله مبدأ تا خط در امتداد یک بردار نرمال، و θ نشان دهنده زاویه بین بردار نرمال و محور x .



- **Acumulator Array:**

برای یافتن خطوط در تصویر، در فضای Hough یک آرایه ایجاد می‌کنیم. هر عضو در این آرایه نشان دهنده یک خط احتمالی در تصویر اصلی است. سپس در تمام نقاط لبه تکرار می‌کنیم و برای هر نقطه (x, y) ، عنصرهای آرایه را که مربوط به خطوط عبوری از (x, y) هستند افزایش می‌دهیم.

- **Peak Detection:**

پس از پردازش تمام نقاط لبه، آرایه مرحله قبل را برای یافتن پیک ها بررسی می‌کنیم. این قله‌ها خطوطی را نشان می‌دهند که بیشترین رای را دریافت کرده‌اند، که نشان‌دهنده شواهد قوی مبنی بر مطابقت آنها با خطوط واقعی در تصویر است.

- **استخراج خط:**

در نهایت خطوط مربوط به پیک های موجود در آرایه را استخراج می‌کنیم. این استخراج شامل تبدیل پارامترهای (ρ, θ) به مختصات دکارتی برای به دست آوردن خطوط در تصویر اصلی است.

سودوکد آن نیز به شرح زیر است:

- Initialize accumulator H to all zeros
- For each edge point (x, y) in the image
For $\theta = 0$ to 180
$$\rho = x \cos \theta + y \sin \theta$$
$$H(\rho, \theta) = H(\rho, \theta) + 1$$
- Find the value(s) of (ρ, θ) where $H(\rho, \theta)$ is a large local maximum

سوال 4:

این سوال تئوری است.

(a) سه مثال از روش ها یا ابزارهای مورد استفاده در **Computer Vision** را ارائه دهید که در آنها تحلیل فوریه نقش مهمی را ایفا می کند، یا برای حل یک مسئله، یا برای کارآمدتر کردن محاسبات به کار می رود. برای هر یک از مثال های خود، علت و فایده آن را توضیح دهید.

تحلیل فوریه نقش مهمی در روش ها و ابزارهای مختلف مورد استفاده در بینایی ایفا می کند. در اینجا سه مثال همراه با توضیحات آنها آورده شده است:

فیلتر کردن تصویر:

روش/ابزار: فیلتر مبتنی بر تبدیل فوریه.

توضیح: تبدیل فوریه معمولاً در پردازش تصویر برای فیلتر حوزه فرکانس استفاده می شود. با تبدیل یک تصویر از حوزه فضایی به حوزه فرکانس با استفاده از تبدیل فوریه، می توانیم اجزای فرکانس آن را تجزیه و تحلیل کنیم. این به ما اجازه می دهد تا عملیاتی مانند فیلتر کردن فرکانس پایین گذر، بالا گذر یا باند گذر را برای حذف نویز، بهبود ویژگی ها یا انجام سایر دستکاری های تصویر انجام دهیم. فیلتر کردن در حوزه فرکانس اغلب کارآمدتر از انجام عملیات مشابه به طور مستقیم در حوزه فضایی است، به خصوص برای تصاویر بزرگ، که این به دلیل تبدیل شدن عملیات کانولوشن به یک ضرب ساده است.

مزایا: تکنیک های فیلتر مبتنی بر فوریه روشی قدرتمند برای بهبود یا اصلاح تصاویر با دستکاری انتخابی اجزای فرکانس آنها ارائه می دهند. این کار کارهایی مانند کاهش نویز، بهبود لبه و استخراج ویژگی را امکان پذیر می کند که منجر به بهبود کیفیت تصویر و نتایج تجزیه و تحلیل می شود.

فشرده سازی تصویر:

روش/ابزار: تبدیل کسینوس گسسته (DCT).

توضیح: DCT، یک نوع تبدیل فوریه، به طور گسترده در الگوریتم های فشرده سازی تصویر مانند JPEG استفاده می شود. DCT بلوک های داده های تصویر را از حوزه فضایی به حوزه فرکانس تبدیل می کند، جایی که بیشتر اطلاعات تصویر با چند ضریب فرکانس پایین نمایش داده می شود. این ضرایب را می توان به طور موثرتری نسبت به مقادیر پیکسل اصلی کوانتیزه و کدگذاری کرد و در نتیجه فشرده سازی با حداقل افت کیفیت تصویر را به همراه داشت.

مزایا: تکنیک های فشرده سازی تصویر مبتنی بر DCT کاهش قابل توجهی در اندازه فایل ارائه می دهند و در عین حال کیفیت تصویر قابل قبولی را حفظ می کنند. روش های مبتنی بر DCT با متمرکز کردن فشرده سازی بر روی اجزای فرکانس بالا، به نسبت های فشرده سازی بالا با افت ادراکی نسبتاً کمی دست می یابند که آن ها را برای ذخیره و انتقال تصاویر بر روی کانال های با پهنای باند محدود مناسب می سازد.

تشخیص اشیا:

روش/ابزار: تبدیل رادون و تبدیل هاف.

توضیح: تکنیک‌های تحلیل فوریه مانند تبدیل رادون و تبدیل هاف برای الگوریتم‌های تشخیص اشیا، به‌ویژه برای تشخیص خطوط و اشکال درون تصاویر، اساسی هستند. این تبدیل‌ها می‌توانند الگوهای مربوط به خطوط، دایره‌ها و سایر اشکال هندسی را با تجزیه و تحلیل تجمع نقاط لبه در جهت‌ها و موقعیت‌های مختلف تشخیص دهند. به عنوان مثال، تبدیل رادون، خطوط مستقیم را با اندازه‌گیری مقدار ساختار خطی در جهات مختلف تشخیص می‌دهد، در حالی که تبدیل Hough خطوط و سایر اشکال را با نگاشت آنها به یک فضای پارامتر تشخیص می‌دهد.

مزایا: استفاده از تکنیک‌های مبتنی بر فوریه برای تشخیص اشیا، استحکام را در برابر تغییرات مقیاس، چرخش و روشنایی فراهم می‌کند و آنها را برای طیف گسترده‌ای از کاربردها مانند تشخیص خط در رانندگی خودکار، تشخیص بارکد و تجزیه و تحلیل تصویر پزشکی مناسب می‌سازد. علاوه بر این، این روش‌ها می‌توانند به‌طور مؤثری با تصاویر پر سر و صدا یا به هم ریخته برخورد کنند و دقت و قابلیت اطمینان سیستم‌های تشخیص اشیا را بهبود ببخشند.

(b) اگر $F(u, v)$ تبدیل فوریه تصویر $f(x, y)$ باشد، حاصل $F(0, 0)$ را به دست آورید. (روابط ریاضی محاسبه را بنویسید.)

یادآوری فرمول‌های تبدیل فوریه:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

حال مقدار ورودی را جاگذاری می‌کنیم:

$$F(0, 0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^0$$

می‌دانیم که $e^0 = 1$:

$$F(0, 0) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

بنابراین، مولفه $F(0, 0)$ برابر است با مجموع تمام مقادیر پیکسل در تصویر اصلی $f(x, y)$ ، که با نتیجه به دست آمده با استفاده از تبدیل فوریه پیوسته مطابقت دارد.

سوال 5:

این سوال عملی است و در اینجا گزارش آن آورده شده است.

در این بخش توابع زیر پیاده سازی شده اند.

نام تابع	کاربرد	ورودی	خروجی
conv	محاسبه وزن همسایه های هر پیکسل	ماتریس تصویر، ماتریس کرنل	2D آرایه
gaussian_kernel	فیلتر گوسی	اندازه، سیگما	-
partial_x	محاسبه مشتق جزئی X	ماتریس تصویر	مشتق X
partial_y	محاسبه مشتق جزئی Y	ماتریس تصویر	مشتق Y
gradient	محاسبه گرادیان	ماتریس تصویر	اندازه گرادیان، جهت گرادیان
non_maximum_suppression	حذف غیر حداکثری	اندازه گرادیان، جهت گرادیان	تصویر با لبه های حداکثری
double_thresholding	آستانه 2 مرحله ای	ماتریس تصویر، آستانه بالا، آستانه پایین	لبه های قوی، لبه های ضعیف
get_neighbors	تشخیص همسایه های معتبر	لوکیشن پیکسل (X,Y)، اندازه تصویر (h,w)	پیکسل های همسایه
link_edges	تشخیص لبه های ضعیف متصل به لبه قوی ها	لبه های قوی، لبه های ضعیف	لبه ها
canny	اعمال الگوریتم canny	ماتریس تصویر، اندازه کرنل، سیگما، آستانه بالا، آستانه پایین	لبه های نهایی

سوال 6:

این سوال تئوری است.

(a) می‌خواهیم از الگوریتم **RANSAC** برای یافتن پارامترهای یک دایره در تصویر استفاده کنیم. در صورتی که بدانیم تنها 40 درصد از لبه‌های تصویر مربوط به دایره مورد نظر است و بخواهیم با احتمال بالای 0.99 به پارامترهای صحیح دست پیدا کنیم، به چند تکرار نیاز است؟

ابتدا به توضیح الگوریتم RANSAC می‌پردازیم.

RANSAC (Random sample consensus) یا همان "توافق نمونه تصادفی" بر اساس انتخاب تعداد محدودی از نقاط است که به صورت تصادفی نمونه برداری شده‌اند و آموزش می‌بیند سپس با آن‌ها یک مدل را می‌سازیم و با نقاط دیگر آن را صحت‌سنجی می‌کنیم و در نهایت بهترین مدل انتخاب می‌شود. به عبارت دیگر برای تخمین پارامترهای یک مدل ریاضی از مجموعه‌ای از نقاط داده مشاهده شده که ممکن است حاوی نقاط پرت باشد، استفاده می‌شود. برای بدست آوردن تعداد تکرارهای مورد نیاز در این الگوریتم، در حالت ایده‌آل، باید تمام ترکیب‌های "دو نقطه‌ای" را بررسی کنیم.

$$\frac{N(N-1)}{2}$$

اما در حالت غیر ایده‌آل، فرمول آن به شرح زیر است:

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

که در فرمول بالا، w نسبت تعداد نقاط داخلی (inlier) به تمام نقاط است. و p احتمال یافتن یک مجموعه از نقاط بدون outlier است. و k نیز تعداد تکرار است.

اگر فرض کنیم برای تخمین یک خط تنها به دو نقطه نیاز داریم، احتمال آنکه یک مجموعه کاملاً از نقاط inlier تشکیل شود برابر w^2 است. اما در حالت کلی S برابر تعداد نقاط مورد نیاز است.

توضیح گام به گام الگوریتم RANSAC به شرح زیر است:

- مقداردهی اولیه:

یک زیرمجموعه انتخابی تصادفی از نقاط داده مشاهده شده (حداقل مورد نیاز برای تخمین پارامترهای مدل) را انتخاب می‌کنیم و از آنها برای ساخت یک مدل استفاده می‌کنیم.

- ساخت مدل:
یک مدل را با زیر مجموعه انتخاب شده از نقاط داده، می‌سازیم. مدل می‌تواند هر نمایش ریاضی ای باشد که با مسئله ای که می‌خواهیم حل کنیم، مطابقت داشته باشد. به عنوان مثال، اگر خطوط را در یک تصویر تشخیص می‌دهید، مدل ممکن است یک معادله خطی باشد.
- انتخاب درونی:
تعیین می‌کنیم کدام نقاط داده در مجموعه اصلی با مدل سازگار است. این معمولاً با محاسبه خطا یا فاصله بین هر نقطه داده و مدل انجام می‌شود. نقاط داده ای که خطاهای آنها زیر یک آستانه معین است، "غیرطبیعی" در نظر گرفته می‌شوند.
- ارزیابی مدل:
کیفیت مدل را با شمارش تعداد ورودی‌های آن ارزیابی می‌کنیم. اگر تعداد ورودی‌ها از یک آستانه از پیش تعریف‌شده فراتر رفت یا اگر مدل به اندازه کافی با داده‌ها مطابقت داشت، آن را به عنوان یک راه‌حل بالقوه در نظر می‌گیریم.
- شرایط خاتمه:
مراحل 1-4 را برای تعداد از پیش تعیین شده تکرار یا تا زمانی که معیار مناسب مدل مورد نظر برآورده شود، تکرار می‌کنیم.
- اصلاح مدل (اختیاری):
پس از تکرار برای تعداد معینی از تکرارها، ممکن است مدل را با استفاده از تمام موارد درونی (یعنی نقاط داده ای که به خوبی با مدل مطابقت دارند) اصلاح کنیم. این مرحله می‌تواند به بهبود دقت مدل نهایی کمک کند.
- خروجی:
پارامترهای مدل را برمی‌گردانیم که مطابق با مجموعه ای از ورودی‌هایی هستند که به بهترین وجه با مدل مطابقت دارند.

ایده کلیدی پشت RANSAC این است که حتی اگر مجموعه داده شامل نسبت قابل توجهی از نقاط پرت باشد (نقاط داده ای که با مدل مطابقت ندارند)، باز هم می‌توان با ساخت مکرر مدل به زیر مجموعه‌های داده، تخمین قوی از پارامترهای مدل به دست آورد. با تکرار چندین بار این فرآیند و انتخاب مدلی که بیشترین مقدار را دارد، RANSAC می‌تواند تخمین قابل اعتمادی از پارامترهای مدل ارائه دهد.

RANSAC به طور گسترده در کاربردهای مختلفی مانند برازش خط، برازش دایره، تخمین هموگرافی و موارد دیگر استفاده می‌شود، جایی که تخمین مدل قوی در حضور نقاط پرت ضروری است.

حال با توجه به فرمول و توضیحات بالا به حل سوال می‌پردازیم.

ورودی های مساله:

$$P = 0.99$$

$$W = 0.4 \text{ (40\%)}$$

$$S = 3 \text{ (دایره نیاز به حداقل 3 نقطه دارد)}$$

$$k = \frac{\log(1 - 0.99)}{\log(1 - 0.4^3)} = \frac{\log(0.01)}{\log(0.936)} = \frac{-4.605}{-0.066} \approx 69.77$$

بنابراین، برای دستیابی به احتمال 0.99 برای یافتن پارامترهای صحیح دایره با استفاده از RANSAC، تقریباً به 70 تکرار نیاز داریم. از آنجایی که نمی‌توانیم کسری از یک تکرار داشته باشیم، آن را به نزدیکترین عدد صحیح گرد کردیم. بنابراین، ما به 70 تکرار نیاز داریم.

سوال 7:

قسمت a این سوال تئوری و بقیه عملی هستند و در اینجا گزارش آن ها آورده شده است.

(a) میدانیم برای تشخیص خط از الگوریتم *Hough* و *LSD* استفاده می شود. این دو روش را از جنبه های مختلف با هم مقایسه کنید. (حداقل سه مورد را بررسی کنید).

مقایسه دو الگوریتم *Hough* و *LSD* (Line Segment Detector) به شرح زیر است.

- رویکرد:

Hough Transform: تبدیل *Hough* با تبدیل نقاط موجود در فضای تصویر به خطوط در فضای پارامتر *Hough* کار می کند. سپس با یافتن پیک ها در فضای پارامتر *Hough* خطوط را در فضای تصویر شناسایی می کند.

LSD: *LSD* با استفاده از یک سری فرضیه های پاره خط مستقیماً بخش های خط را در فضای تصویر تشخیص می دهد. و با جستجوی تغییرات قابل توجه در شدت در طول یک خط، بخش های خط را تشخیص می دهد.

- پیچیدگی محاسباتی:

Hough Transform: تبدیل *Hough* سنتی می تواند از نظر محاسباتی گران باشد (به دلیل بردن تصویر به فضای دیگر و برگرداندن آن در انتها)، به خصوص برای تصاویر بزرگ یا زمانی که به یک فضای پارامتر با وضوح بالا نیاز است. با این حال، بهینه سازی هایی مانند "تبدیل تصادفی هاف" و "تبدیل احتمالی پیشرونده هاف" به کاهش بار محاسباتی کمک می کنند.

LSD: *LSD* عموماً سریعتر از *Hough Transform* است، مخصوصاً برای کاربردهای بلادرنگ. این به طور موثر بخش های خط را در فضای تصویر بدون نیاز به تبدیل به فضای پارامتر تشخیص می دهد.

- استحکام در برابر نویز و انسداد:

Hough Transform: *Hough Transform* به دلیل رویکرد مبتنی بر رأی در فضای پارامتر تا حدی در برابر نویز و انسداد مقاوم است. با این حال، می تواند با صحنه های پیچیده که در آن چندین خط قطع یا همپوشانی دارند، مبارزه کند.

LSD: *LSD* به گونه ای طراحی شده است که در برابر نویز و انسداد جزئی مقاوم باشد. این می تواند بخش های خط را حتی در تصاویر پر سر و صدا یا زمانی که قسمت هایی از خطوط مبهم هستند تشخیص دهد.

- حساسیت پارامتر:

Hough Transform: عملکرد *Hough Transform* می تواند به تنظیم پارامتر حساس باشد، مانند آستانه تشخیص اوج در فضای پارامتر *Hough* و وضوح فضای پارامتر.

LSD: LSD نسبت به روش های سنتی Hough Transform حساسیت کمتری به تنظیمات پارامتر دارد. معمولاً به پارامترهای کمتری برای تنظیم نیاز دارد که استفاده از جعبه را آسان تر می کند.

- دقت تشخیص:

Hough Transform: Hough Transform تمایل دارد نتایج دقیقی را برای خطوط کاملاً مشخص، به ویژه در تصاویر تمیز با کمترین نویز ارائه دهد. با این حال، ممکن است با تشخیص خطوط کوتاه یا تکه تکه مشکل داشته باشد و در صحنه های درهم و برهم، مثبت کاذب ایجاد کند.

LSD (Line Segment Detector): LSD به ویژه در تشخیص بخش های کوتاه موثر است و می تواند خطوط تکه تکه شده را بهتر از Hough Transform کنترل کند. همچنین در صحنه های بهم ریخته کمتر مستعد ابتلا به موارد مثبت کاذب است.

به طور خلاصه، در حالی که Hough Transform و LSD معمولاً برای تشخیص خط استفاده می شوند، اما در رویکرد، پیچیدگی محاسباتی، استحکام، حساسیت پارامتر و دقت تشخیص متفاوت هستند. انتخاب بین دو روش بستگی به الزامات خاص برنامه مانند راندمان محاسباتی، استحکام در برابر نویز و پیچیدگی صحنه مورد تجزیه و تحلیل دارد.