

بسمه تعالی



دانشکده مهندسی کامپیوتر

بینایی کامپیوتر

نام استاد: دکتر محمدی

تمرین چهارم

نام دانشجو: آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

آبان ۱۴۰۱

فہرست

| | |
|---|-----------------------|
| ۳ | پاسخ سوال اول |
| ۳ | منابع |
| ۴ | پاسخ سوال دوم |
| ۴ | الف) |
| ۴ | ب) |
| ۵ | پاسخ سوال سوم |
| ۵ | پ) |
| ۵ | ت) |
| ۶ | منابع |
| ۷ | پاسخ سوال چہارم |
| ۷ | الف) |
| ۷ | ب) |
| ۷ | پ) |
| ۸ | منابع) |

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

در اینجا تصویر 2×2 است، یعنی $M=N=2$.

$$F(u,v) = f(0,0) \times 1 + f(1,0) e^{-j\pi u} + f(0,1) e^{-j\pi v} + f(1,1) e^{-j\pi(u+v)}$$

$$F(u,v) = 2 + 3e^{-j\pi u} + e^{-j\pi v} + 4e^{-j\pi(u+v)}$$

با جایگذاری u و v به دست می آوریم:

$$F(0,0) = 2 + 3 + 1 + 4 = 10$$

$$F(0,1) = 2 + 3e^{-j\pi} + 1 + 4e^{-j\pi} = 5 + 5e^{-j\pi} = 5 - 5 = 0$$

$$F(1,0) = 2 + 3e^{-j\pi} + 1 + 4e^{-j\pi} = 3 - 7 = -4$$

$$F(1,1) = 2 + 3e^{-j\pi} + e^{-j\pi} + 4e^{-2j\pi} = 2 - 3 - 1 + 4 = 2$$

| | |
|----|----|
| 10 | -4 |
| 0 | 2 |

پس تبدیل فویر به این صورت است:

پاسخ سوال دوم

الف

برای یک تصویر $n \times n$ میتوانیم به این صورت بنویسیم:

$$F(u, v) = \sum_{x=0}^n \sum_{y=0}^n f(x, y) e^{-2\pi j(ux+vy)/N}$$

و چون بازه هر دو سیگما یکی است، میتوان گفت که مقدار رابطه بالا با رابطه زیر در زوج مرتب های مختلف برای u و v یکی است:

$$F(v, u) = \sum_{x=0}^n \sum_{y=0}^n f(x, y) e^{-2\pi j(vx+uy)/N}$$

در نتیجه تبدیل فوریه همواره نسبت به قطر اصلی قرینه است. پس در ابعاد $n + (n^2 - n)/2$ است.

ب

مطابق با رابطه تبدیل فوریه:

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

میبینیم که اگر $u=v=0$ باشد که $F(0,0)$ را به دست آوریم، حاصل توان e برابر صفر می شود. در نتیجه فقط روی $f(x, y)$ ، سیگما میگیریم. پس خانه ی $0,0$ تبدیل فوریه درواقع مجموع روشنایی تمام نقاط تصویر اصلی است. و همواره عددی حقیقی است و بخش موهومی ندارد.

پاسخ سوال سوم

(پ)

میانه فیلتر قوی تری نسبت به میانگین است چون یک پیکسل outlier در یک ناحیه تأثیر قابل توجهی بر مقدار میانه نخواهد داشت. و این برای برخی از انواع نویزهای خاص مانند نویزهای "گوسی"، "تصادفی" و "نمک و فلفل" بسیار مناسب است. چون نویز فرکانس زیادی دارد و حذف خواهد شد.

از آنجایی که مقدار میانه باید در واقع مقدار یکی از پیکسل های همسایگی باشد، فیلتر میانه مقادیر پیکسل غیر واقعی جدیدی را زمانی که فیلتر در یک لبه قرار می گیرد ایجاد نمی کند. به همین دلیل فیلتر میانی در حفظ لبه های تیز بسیار بهتر از فیلتر میانگین است.

یکی از مشکلات عمده فیلتر میانه این است که محاسبه آن نسبتاً گران و پیچیده است. برای یافتن میانه لازم است همه مقادیر موجود در همسایگی را به ترتیب عددی مرتب کنیم و این کار نسبتاً کند است، حتی با الگوریتم های مرتب سازی سریع مانند quick sort. با این حال، الگوریتم اصلی را می توان تا حدودی برای سرعت افزایش داد. یک روش رایج این است که توجه داشته باشید که وقتی پنجره همسایگی روی تصویر لغزنده می شود، بسیاری از پیکسل های پنجره از یک مرحله به مرحله دیگر یکسان هستند و ترتیب نسبی آنها با یکدیگر بدیهی است که تغییری نکرده است.

(ت)

می توانیم فیلترهای مختلفی را پیش از مشتق گیری اعمال کنیم تا نویزها با حفظ جزئیات، حذف شوند. برای مثال استفاده از فیلتر گوسی برای نویز موجود در این تصویر فکر میکنم مناسب باشد. اما روش خوب و مرسوم استفاده از motphological operator است.

عملیات مورفولوژیکی مجموعه وسیعی از عملیات پردازش تصویر است که تصاویر دیجیتال را بر اساس شکل آنها پردازش می کند. در یک عملیات مورفولوژیکی، هر پیکسل تصویر با مقدار پیکسل دیگر در همسایگی خود مطابقت دارد. با انتخاب شکل و اندازه پیکسل همسایگی، می توانید یک عملیات مورفولوژیکی ایجاد کنید که به اشکال خاص در تصویر ورودی حساس است.

- <https://www.sciencedirect.com/topics/computer-science/median-filter>
- <https://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm#:~:text=By%20calculating%20the%20median%20value,affect%20the%20median%20value%20significantly.>
- <https://www.geeksforgeeks.org/morphological-operations-in-matlab/>

پاسخ سوال چهارم

(الف)

این الگوریتم ۳ مرحله دارد. ابتدا تبدیل فوریه دو بعدی از عکس دو بعدی میگیریم. سپس نواحی نویزی آن ناحیه را حذف میکنیم. به این صورت که ضریب سری فوریه درصد ثابتی از ستون ها و ردیف های میانی را برابر ۰ می گذاریم. برای پیدا کردن این درصد بهینه، با دقت ۰.۰۰۱ مقایسه ای انجام داده ام و بهینه ترین حالت که ۰.۱۳۷ بود را انتخاب کردم که psnr بیشتری دارد. در بخش بعدی psnr را توضیح خواهم داد. در نهایت، تبدیل فوریه معکوس میگیریم و بخش های real حاصل را به عنوان عکس بدون نویز معرفی می کنیم.

(ب)

با مشاهده تصاویر آخرین سلول نوتبوک مشخص است که این نویز به خوبی رفع شده است. PSNR بیشتر برای تخمین راندمان کمپرسورها، فیلترها و غیره استفاده می شود. نسبت سیگنال به نویز پیک (PSNR) نسبت بین حداکثر قدرت ممکن یک تصویر و قدرت نویز مخرب است که بر کیفیت نمایش آن تأثیر می گذارد. برای تخمین PSNR یک تصویر، باید آن تصویر را با یک تصویر تمیز ایده آل با حداکثر توان ممکن مقایسه کرد. هر چه مقدار PSNR بزرگتر باشد، روش فشرده سازی یا فیلتر مربوطه کارآمدتر است.

(پ)

از نوع جمع شونده است، چون متناوب می باشد. نویز جمع شونده به این صورت است:

$$\text{Noisy_image}[t] = \text{original_image}[t] + \text{noise}[t]$$

و نویز ضرب شونده هم به این صورت است:

$$\text{Noisy_image}[t] = \text{original_image}[t] * \text{noise}[t]$$

✓ حذف نویز ضربی دشوارتر از نویز جمع شونده است. مدل های بازیابی تصویر زیادی برای حذف نویز جمع شونده داریم.

✓ تاثیر نویز جمع شونده روی تصویر اصلی کمتر است. چون در نویز ضرب شونده کاملاً پیکسل چند برابر می شود.

✓ نویز لکه ای مثال خوبی برای نویز ضرب شونده و نویز گاوسی مثال خوبی برای نویز جمع شونده است.

✓ نویز ضربی عمدتاً در تصاویر رادار و نویز جمعی در در تصاویر پزشکی دیده می شوند.

منابع

- [Python | Peak Signal-to-Noise Ratio \(PSNR\) - GeeksforGeeks](#)
- www.gofastresearch.com