

به نام خدا

گزارش تمرین دوم درس پردازش و تحلیل تصاویر پزشکی

نگین اسماعیل زاده ۹۷۱۰۴۰۳۴

تمرین های نوشتاری :

۱.

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{u}{M}x + \frac{v}{N}y)} \quad , \quad f(x, y) = A e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} A e^{-\frac{(x^2+y^2)}{2\sigma^2}} e^{-j2\pi(\frac{u}{M}x + \frac{v}{N}y)}$$

$$= \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} A e^{-\frac{(x^2+y^2 + j4\pi\sigma^2(\frac{u}{M}x + \frac{v}{N}y))}{2\sigma^2}}$$

$$= A e^{-2\pi^2\sigma^2(\frac{u^2}{M^2} + \frac{v^2}{N^2})} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} e^{-\frac{(x + 2\pi j\sigma^2\frac{u}{M})^2 + (y + 2\pi j\sigma^2\frac{v}{N})^2}{2\sigma^2}}$$

$$\rightarrow A e^{-2\pi^2\sigma^2(\frac{u^2}{M^2} + \frac{v^2}{N^2})} \sum_{x=0}^{M-1} e^{-\frac{(x + 2\pi j\sigma^2\frac{u}{M})^2}{2\sigma^2}} \sum_{y=0}^{N-1} e^{-\frac{(y + 2\pi j\sigma^2\frac{v}{N})^2}{2\sigma^2}}$$

که در رابطه فوق هر کدام از سیگما ها را اگر بر $\sqrt{\pi}$ تقسیم کنیم حاصل برابر مساحت زیر تابع توزیع گاوسی میشود که برابر با ۱ است نتیجتاً داریم :

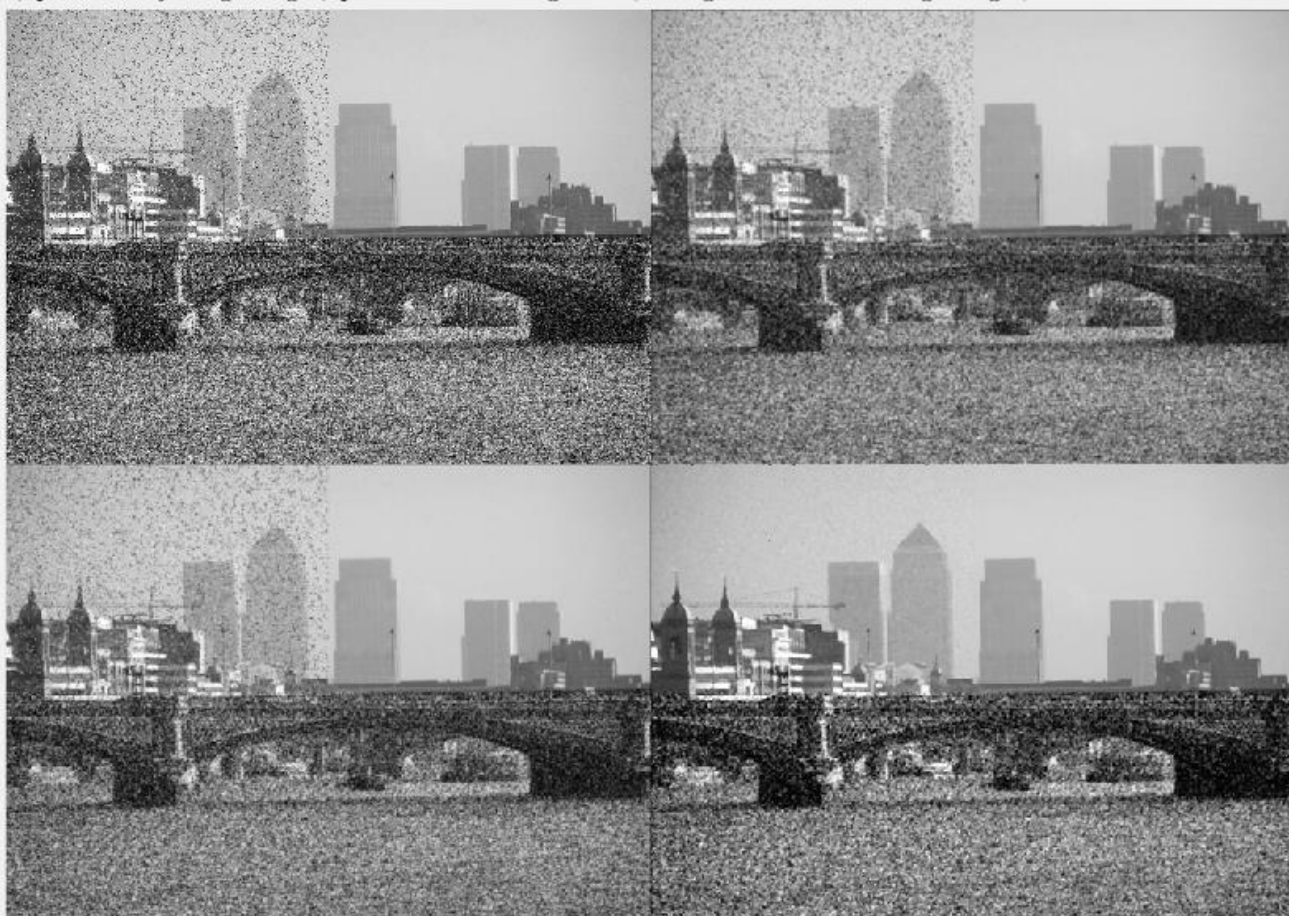
$$F(u, v) = A\pi e^{-2\pi^2\sigma^2\left(\frac{u^2}{M^2} + \frac{v^2}{N^2}\right)}$$

همانطور که میبینیم تبدیل فوریه شبیه به یک فیلتر گوسی شد ، این فیلتر پایین گذر است زیرا اگر فرکانس های افقی و عمودی را به بینهایت میل دهیم حاصل به صفر میل میکند ($\lim_{u,v \rightarrow \infty} \{F(u, v)\} = 0$) و در فرکانس های کم مقادیر F بزرگ است.

تمرین های عملی :

۱. شکل نهایی در ۴ حالت ذکر شده به صورت زیر می باشد :

left,up:main noisy image * right,up:mean filtered image * left,down:gaussian filtered image * right,down:median filtered image



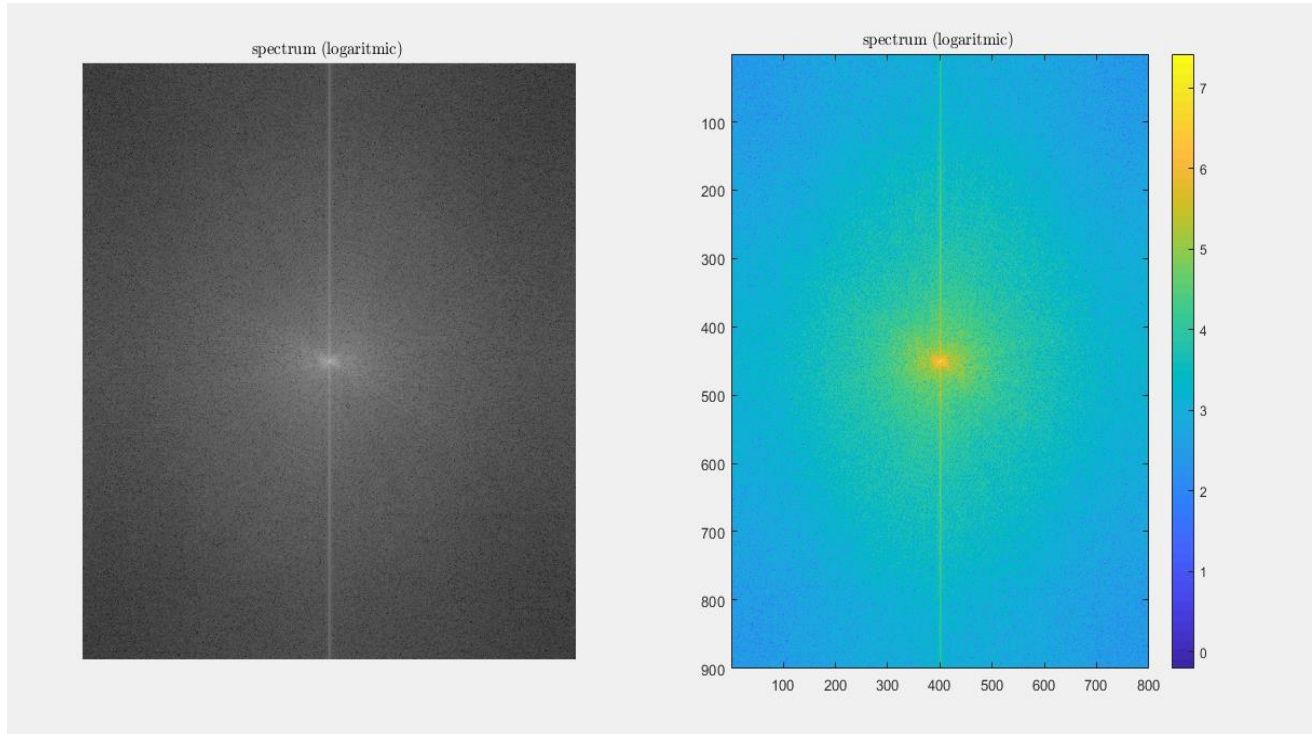
	SNR_region1	11.9952
	SNR_region1_after_gaussianfilter	19.0502
	SNR_region1_after_meanfilter	19.0987
	SNR_region1_after_medianfilter	23.8432
	SNR_region3	3.2549
	SNR_region3_after_gaussianfilter	10.4495
	SNR_region3_after_meanfilter	10.8364
	SNR_region3_after_medianfilter	9.2566
	SNR_region4	4.1325
	SNR_region4_after_gaussianfilter	11.3847
	SNR_region4_after_meanfilter	11.7327
	SNR_region4_after_medianfilter	10.0195

جدول مقایسه کیفیت تصویر بعد از فیلتر	نویز نمک فلفل (ناحیه ۱ تصویر)		نویز نمک فلفل + نویز گاوسی (ناحیه ۳ تصویر)		نویز گاوسی (ناحیه ۴ تصویر)	
	SNR	بصری	SNR	بصری	SNR	بصری
فیلتر گاوسی	خوب	متوسط	خیلی خوب	خوب	خیلی خوب	خوب
فیلتر میانگین گیر	خوب	بد	خیلی خوب	بد	خیلی خوب	خوب
فیلتر میانه گیر	عالی	عالی	خوب	خوب	خوب	بد

در مجموع نتیجه میگیریم مناسب تری فیلتر برای نویز نمک فلفل ، فیلتر میانه گیر است . همچنین برای نویز گاوسی به ترتیب فیلتر گاوسی و فیلتر میانگین گیر مناسب هستند . اما فیلتر میانگین گیر و حتی گاوسی به هیچ وجه نمیتواند برای نویز نمک فلفلی مناسب باشد.

۲.

(آ) لگاریتم اندازه تبدیل فوریه یا لگاریتم طیف تصویر به صورت زیر است.



(ب)

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

$$\text{میانگین سطح روشناسی تصویر} = \frac{F(0,0)}{MN}$$

میانگین سطح روشنایی تصویر برابر است با : $light_level_Image$

سمت راست رابطه از روی طیف محاسبه شده برابر است با : $scaled_F00$

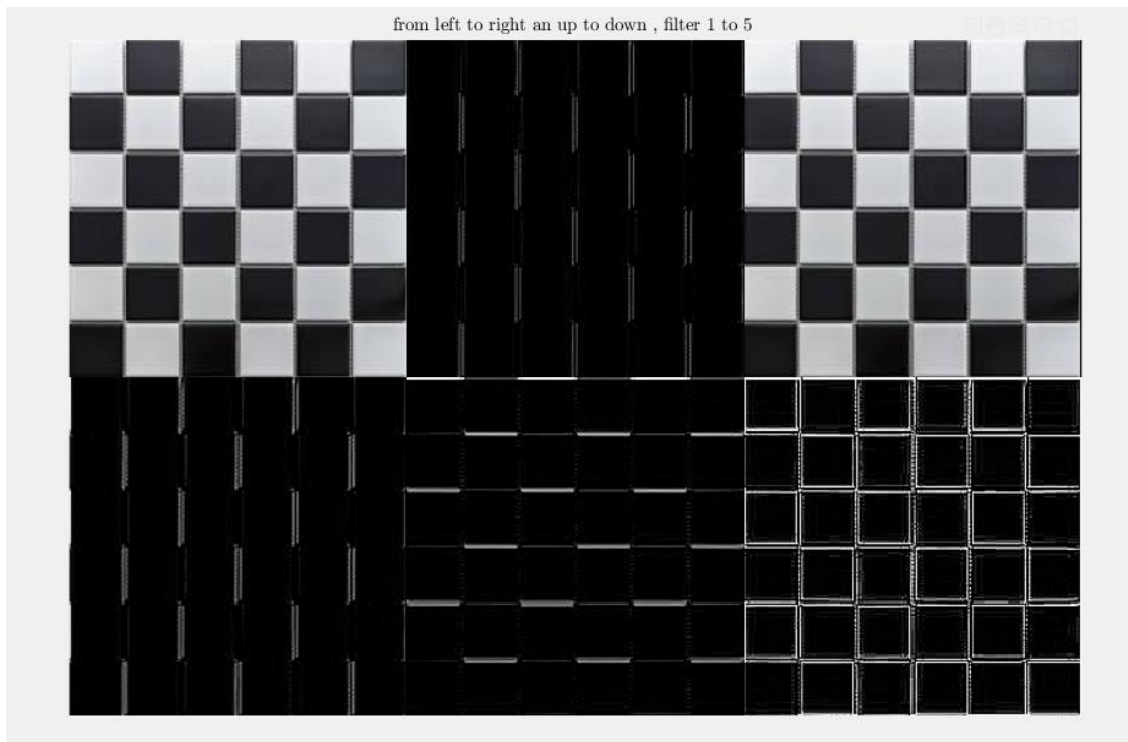
```
light_level_Image =  
  
36.3036  
  
scaled_F00 =  
  
36.3036
```

همانطور که میبینیم از لحاظ عملی هم رابطه ی تساوی بالا اثبات شد :

۳.

$$filter1 = [1, -1] \quad , \quad filter2 = [1, 0] \quad , \quad filter3 = [1, 0, -1] \quad , \quad filter4 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \quad , \quad filter5 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

(۱)



توضیحات :

فیلتر اول در واقع هر اختلاف روشنایی هر خانه را با خانه ی سمت راستش محاسبه میکند . به این ترتیب لبه ها (محل تغییر روشنایی) در محور افقی مشخص خواهند شد (مطابق شکل دوم از بالا و چپ) . فیلتر دوم هر خانه به با صفر برابر خانه ی مجاور سمت راستش جمع میکند که درواقع معادل این است که هر خانه را به خود همان خانه مینگارد ، نتیجتاً تغییری در تصویر حاصل نمیکند (مطابق شکل سوم از بالا و چپ). به همین ترتیب که توضیح داده شد فیلتر سوم لبه های افقی را به صورت زخیم تر نمایان میکند ، در واقع لبه ها جهت دار میشوند لبه ی از چپ به راست و از راست به چپ دو لبه تلقی میشوند (مطابق شکل اول از پایین و چپ). فیلتر چهارم مطابق فیلتر سوم عمل میکند با این تفاوت که روی ستون ها این تاثیر را میگذارد بنابراین لبه های عمودی را زخیم نمایش میدهد (مطابق شکل دوم از پایین و چپ) . فیلتر پنجم هم ارتفاع یافته مجموعه ی فیلتر های قبل است ، که اختلاف روشنایی هر خانه را با ۸ خانه ی مجاورش محاسبه میکند و به این ترتیب همه لبه های افقی و عمودی را پیدا میکند (مطابق شکل پایین راست).

(ب)

الگوریتم *sobel* : روش سوبل لبه ها را با استفاده از تخمین زدن مشتق پیدا می کند به این صورت که لبه ها را در آن نقاطی بر می گرداند که گرادیان تصویر ماکزیمم است . این روش تنها مشتق را در دو جهت افقی و عمودی محاسبه میکند بنابراین دو ماسک مشتق گیر به صورت G_x و G_y داریم بعد از اعمال این دو ماسک به تصویر ، تصویر جدید را به این صورت تعریف میکنیم : $\sqrt{G_y^2\{I\} + G_x^2\{I\}}$ که در این تصویر لبه ها مشخص شده است . در این روش یک پارامتر *threshold* حد تعیین لبه را مشخص میکند .

الگوریتم *canny* : در روش کنی در مرحله ی اول تصویر اولیه با فیلتر گوسی فیلتر میشود تا نویز حذف شود . در مرحله دوم از ماسک سوبل برای یافتن لبه های قوی از روش تشخیص محل ماکزیمم شدن گرادیان استفاده میکنیم.

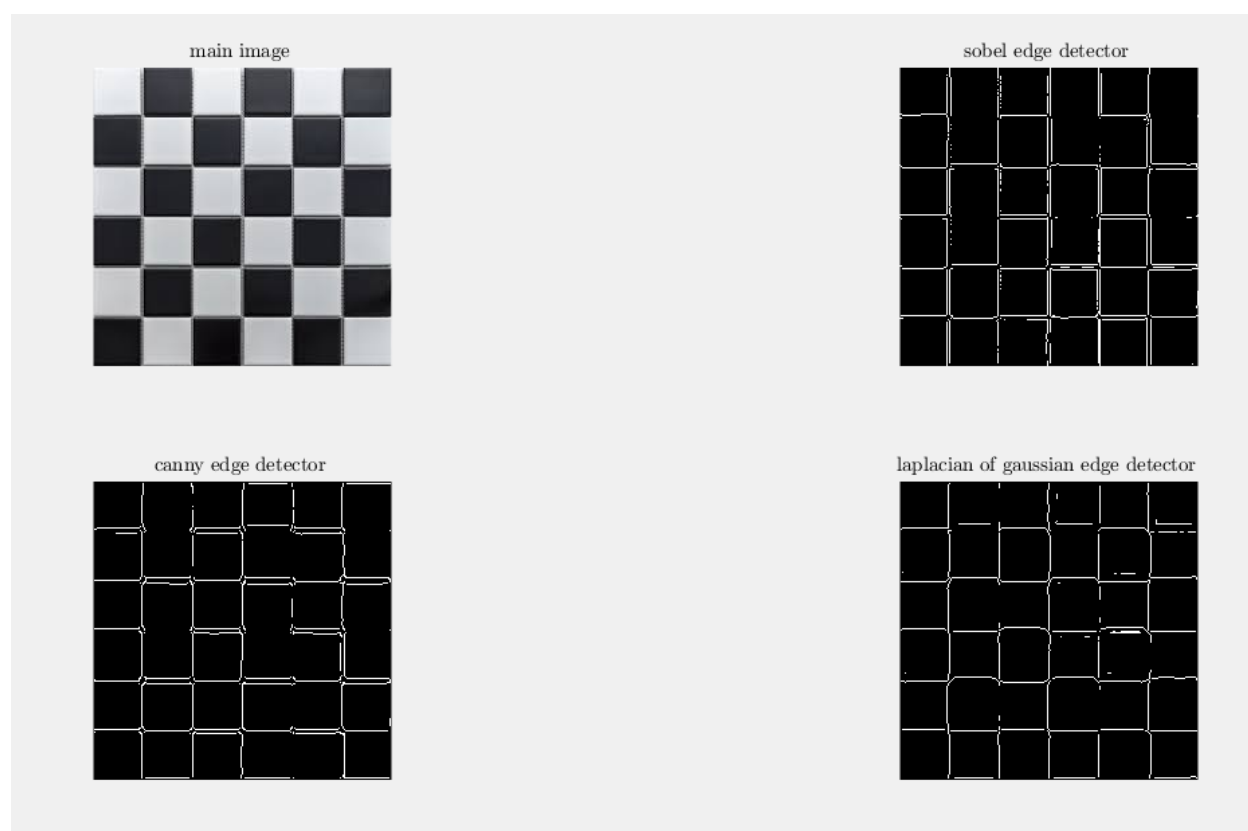
$$|G| = |G_x| + |G_y| , \quad \theta = \tan^{-1} \left\{ \frac{G_y}{G_x} \right\}$$

در مرحله بعد θ های به دست آمده را به یکی از ۴ زاویه ۰ ، ۴۵ ، ۹۰ ، ۱۳۵ می نگاریم . در این مرحله نقاطی که ماکزیمم در ان ها رخن داده تضعیف میوند تا با لبه اشتباه گرفته نشوند و در آخر از دو استانه بالا و پایین استفاده میشود ، هر پیکسل که گرادیان آن از استانه ی بالا بیستر باشد در تصویر نهایی لبه است ، در صورتی

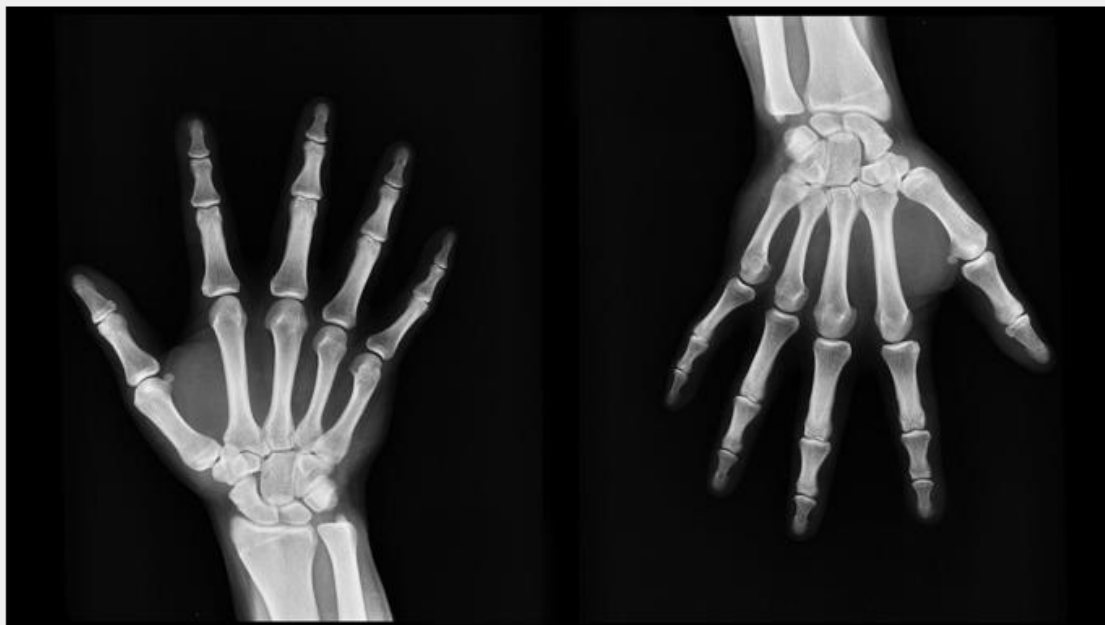
که از استانه پایین مقدار گرادیان پیکسل کمتر باشد لبه نیست و در صورتی که بین این دو باشد به عنوان یک لبه ی ضعیف تلقی میشود و در صورتی در تصویر نهایی به عنوان لبه در نظر گرفته میشود که در مجاورت آن پیکسل لبه ی دیگری حضور داشته باشد . این روش به دو پارامتر $threshold$ برای تعیین استانه بالا و پایین نیاز دارد ، همچنین برای فیلتر گوسی که در مرحله اول اعمال میشود پارامترهای انحراف معیار و اندازه ماسک گوسی قابل تنظیم است .

الگوریتم log : در این الگوریتم نیز مانند روش قبل ابتدا یک فیلتر گوسی جان حذف نویز به تصویر اعمال میشود سپس ماسک لاپاسین را روی آن اعمال میکنیم ، لاپاسین در واقع نمودی از مشتقات مرتبه دوم هست ، بنابراین به تغییرات ناگهانی سطح روشنایی واکنش زیاد نشان میدهد و آن ها را آشکار میکند . علت استفاده از فیلتر ابتدایی نیز به دلیل همین حساسیت این روش به نویز است برای فیلتر گوسی که در مرحله اول اعمال میشود پارامترهای انحراف معیار و اندازه ماسک گوسی قابل تنظیم است .

شکل زیر نتیجه ی اعمال این الگوریتم ها به تصویر است ، چون تصویر لبه های افقی و عمودی دارد و فاقد لبه های منحنی شکل است انتظار داریم همانطور که مشاهده میکنیم الگوریتم سوبل از کنی قدری بهتر عمل کند .



۴. اگر فاز تبدیل فوریه یک تصویر را با زاویه ای جمع کنیم و تبدیل معکوس بگیریم در حوزه ی زمان آن تصویر به اندازه زاویه ی مذکور حول مبدا خود میچرخد ، که در اینجا چون سیگنال تصویر حقیق است ۱۸۰ درجه چرخش در حوزه ی زمان معادل جمع فاز فوریه با π ، قرینه کردن فاز ، یا کانجوگیت کردن فوریه می باشد . حال این کار روی تصویر خواسته شده به صورت زیر می باشد .



اثبات ریاضی :

اگر محور های x و y را θ درجه بچرخانیم ، x و y برای تصویر جدید به صورت زیر محاسبه میشود.

$$\begin{cases} x' = x \cos(\theta) - y \sin(\theta) \\ y' = x \sin(\theta) + y \cos(\theta) \end{cases} \rightarrow \begin{cases} x = x' \cos(\theta) + y' \sin(\theta) \\ y = -x' \sin(\theta) + y' \cos(\theta) \end{cases}$$

حال تبدیل فوریه را در این صفحه یا به عبارتی برای تصویر چرخیده شده محاسبه می کنیم ، داریم :

$$f_r(x, y) = f(x \cos(\theta) + y \sin(\theta), -x \sin(\theta) + y \cos(\theta))$$

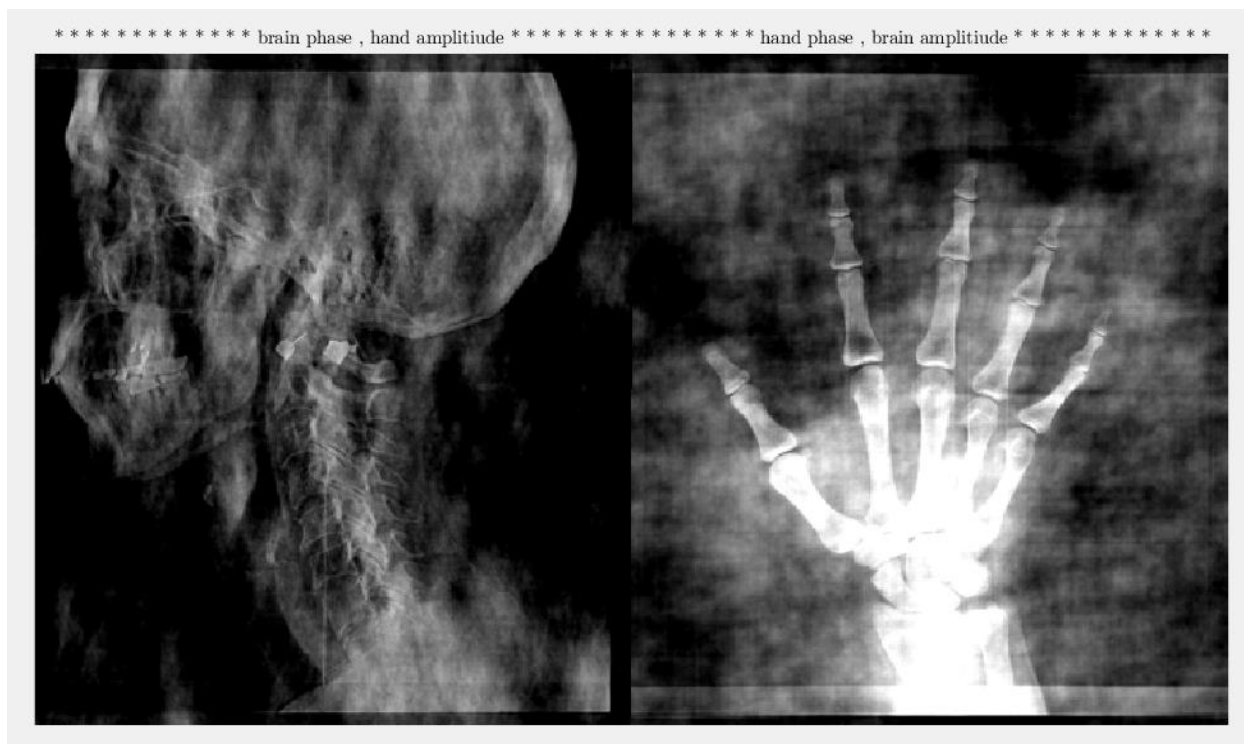
$$F_r(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f_r(x, y) e^{-j2\pi(\frac{u}{M}x + \frac{v}{N}y)}$$

$$\sum_{x'=0}^{M-1} \sum_{y'=0}^{N-1} f_r(x', y') e^{-j2\pi \left(\frac{u}{M} (x' \cos(\theta) - y' \sin(\theta)) + \frac{v}{N} (x' \sin(\theta) + y' \cos(\theta)) \right)}$$

$$\rightarrow F_r(u, v) = F(u \cos(\theta) + v \sin(\theta), -u \sin(\theta) + v \cos(\theta))$$

که به این معنا است که تبدیل فوری تصویر نیز به اندازه θ درجه چرخیده میشود .

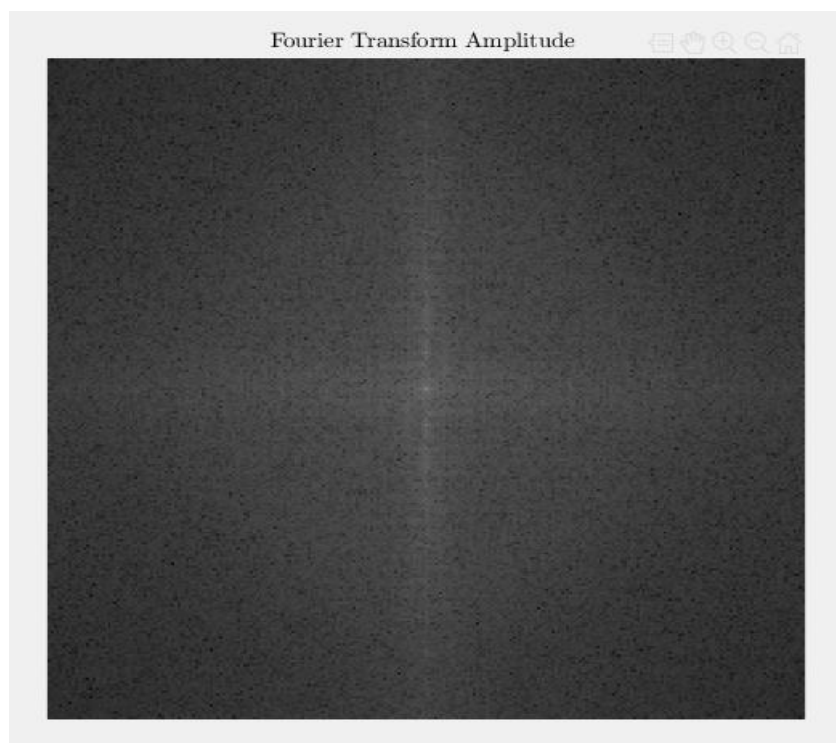
۵.



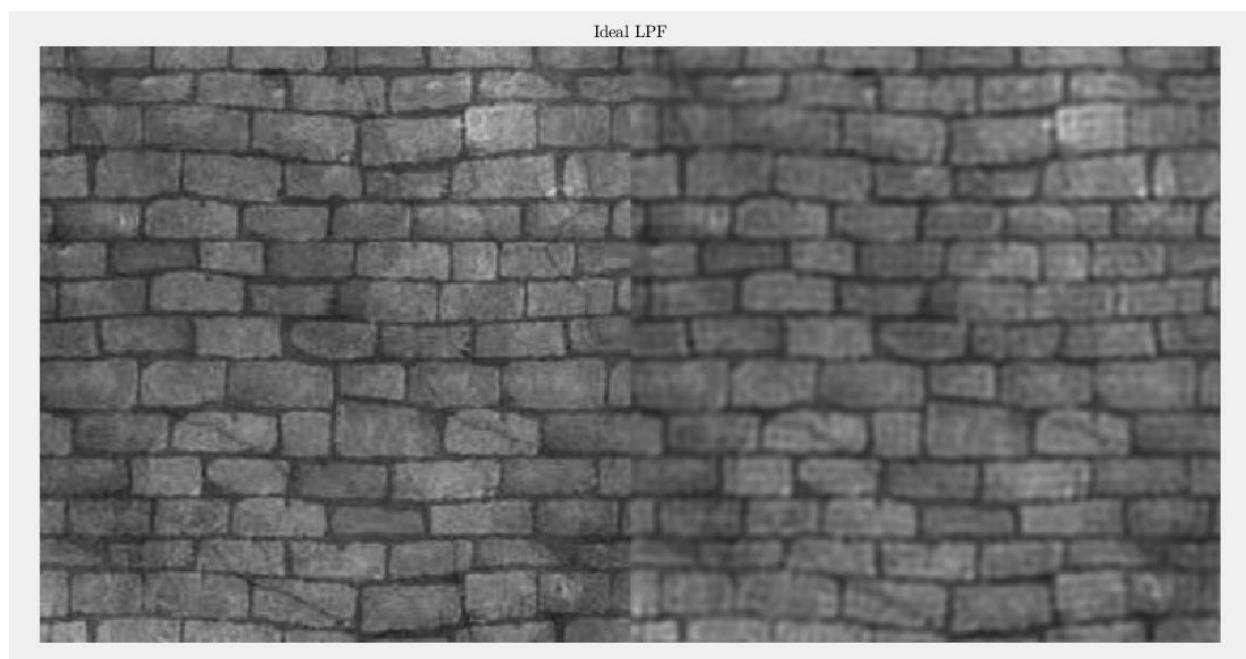
همانطور که در شکل میبینیم تصویری که فاز آن از فاز تصویر اول گرفته شده و اندازه ی آن از اندازه تصویر دوم بسیار شباهت به تصویر اول دارد و تصویری که فاز آن را فاز تصویر دوم و اندازه ی آن از اندازه تصویر اول گرفته شده بسیار شباهت به تصویر دوم دارد . در نتیجه میتوان گفت که اطلاعات اصلی مربوط به تصویر در فاز تبدیل فوری تصویر قرار دارند و در تمامی فرایندها نباید هیچ گاه به فاز تصویر خدشه ایجاد کنیم چون باعث از دست رفتن اطلاعات اصلی مربوط به تصویر میشود .

۶.

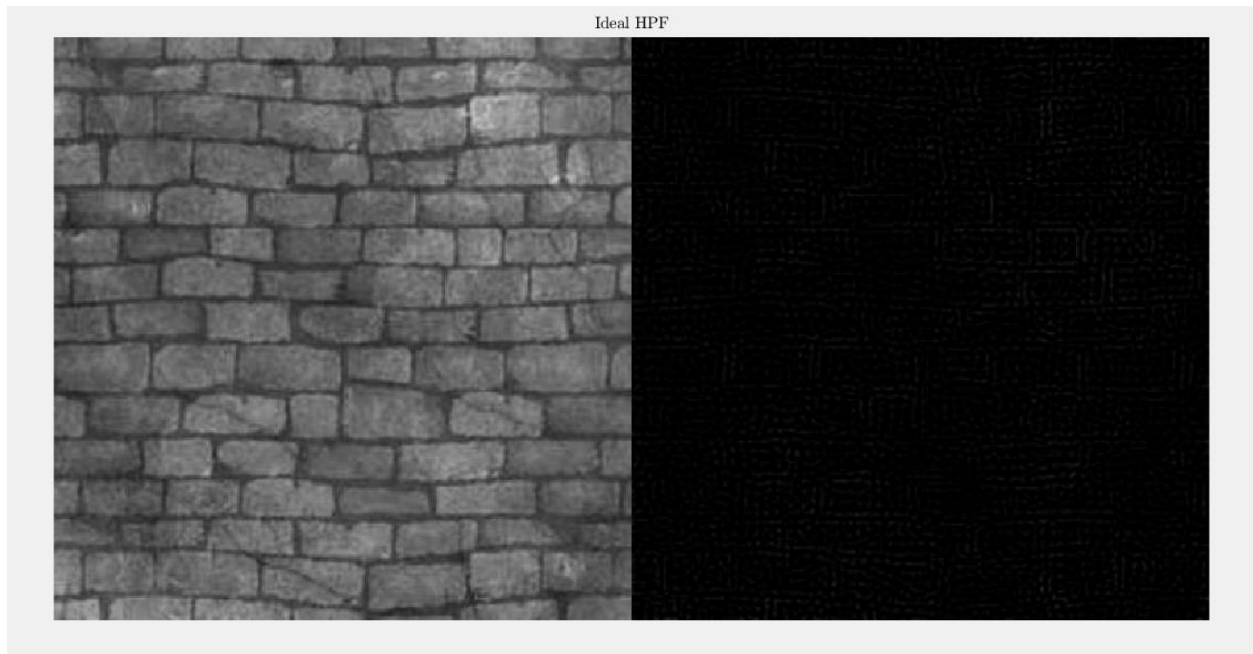
(آ) اندازه تبدیل فوریه تصویر به این صورت است (البته برای وضوح بیشتر ضربی از لگاریتم آن رسم شده است) :



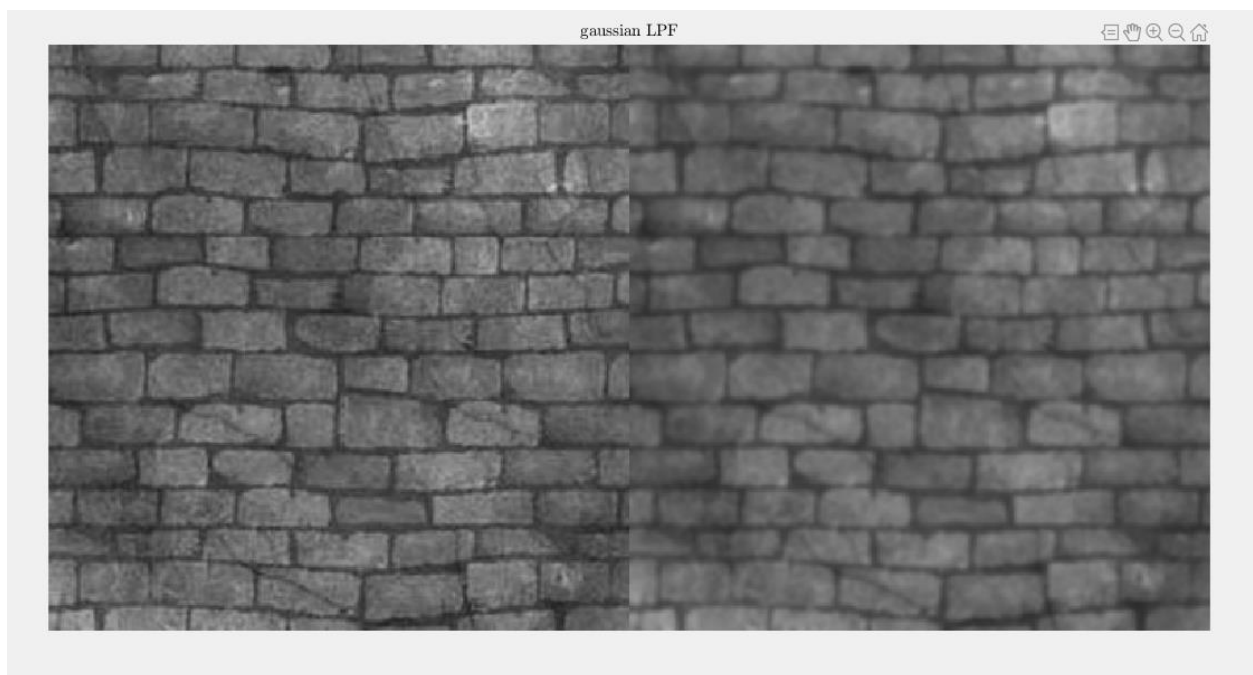
(ب) برای این موضوع فرکانس قطع ۰.۲۸ در نظر گرفته شد ، در نتیجه پس از اعمال فیلتر ایده آل پایین گذر (دیسک) داریم : (تصویر سمت چپ همواره نمایانگر تصویر اصلی قبل از اعمال فیلتر است)



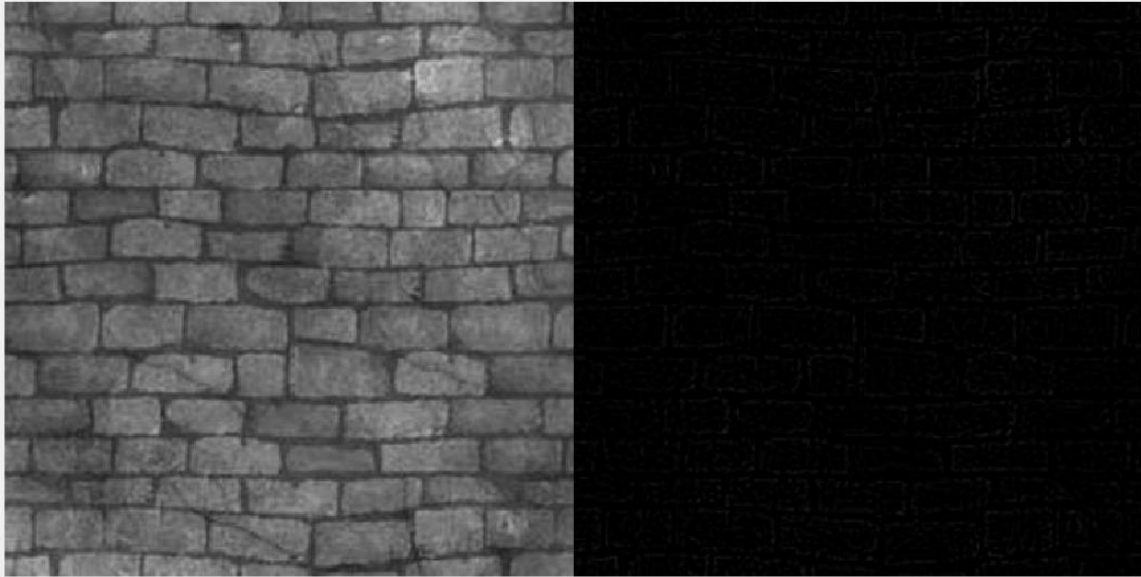
ج) نتیجه اعمال فیلتر ایده آل بالا گذر نیز به صورت زیر می باشد :



د) همین کار را برای فیلتر گوسی انجام میدهیم ، به ترتیب برای حالت پایین گذر و بالا گذر داریم :



gaussian HPF

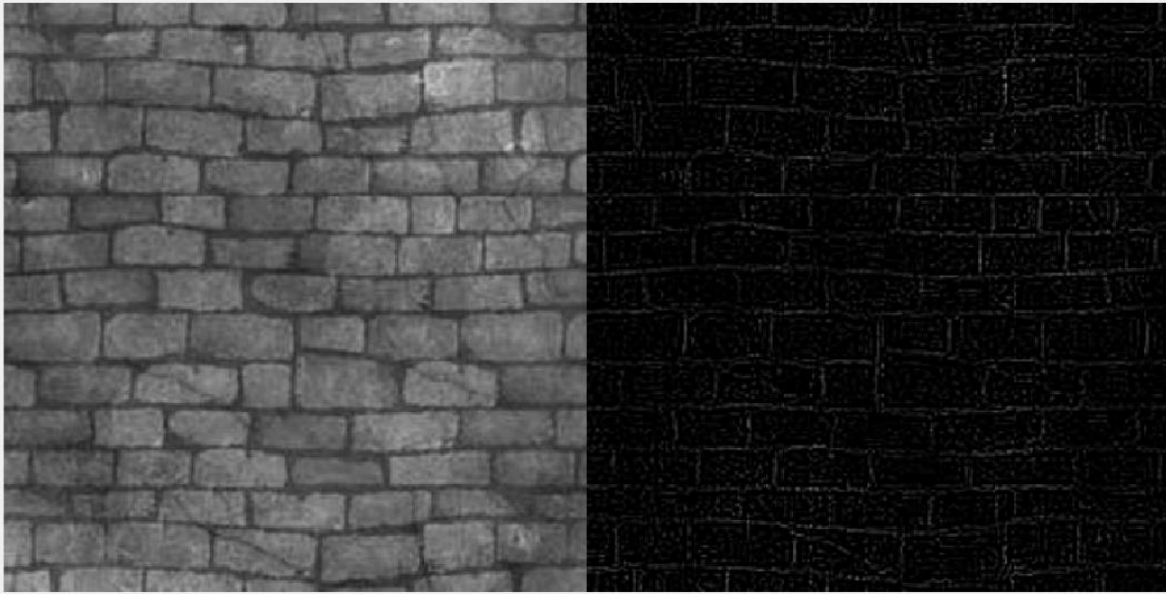


اگر از فیلتر لاپلاسین استفاده کنیم به ترتیب برای حالت پایین گذر و بالا گذر داریم :

laplacian LPF



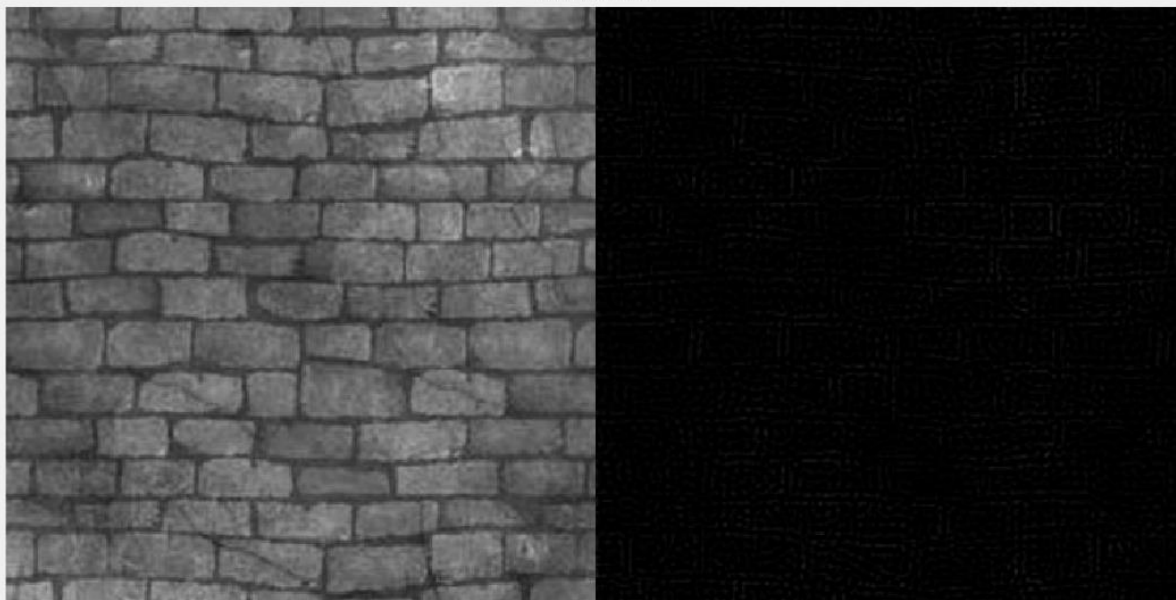
laplacian HPF



اگر از فیلتر باترورف استفاده کنیم به ترتیب برای حالت پایین گذر و بالا گذر داریم : (مرتبه ی فیلتر برابر با ۱۰ فرض شده است)

butterworth LPF





(۵)

همانطور که میبینیم تصویر پس از اعمال فیلتر ایده آل بسیار تار و محو شده است و همچنین با توجه به تصویر فیلتر بالا گذر ایده آل متوجه میشویم نقطه های سفید غیر مطلوب (معادل نقاط سفید در تصویر اصلی) در اثر این فیلتر ایجاد شده (پدیده ی *ringing*). علت این امر این است که تبدیل فوری معکوس فیلتر ایده آل تابع جینک است که محدود نیست و در واقع اگر آن را محدود کنیم (در دنیای گسسته) اطلاعات زیادی از بین میرود که باعث میشود اگر در این حالت تبدیل فوری ی آن را محاسبه کنیم به جای یک دیسک با تقریب از یک دیسک به همراه تعداد زیادی حلقه به همان مرکز مواجه شویم ، در اثر این پدیده این نوع تخریب در تصویر میشود که به هیچ وجه مطلوب نیست چون هم تصویر دچار تخریب شده است هم دچار تار شدگی .