



تمرین شماره 3

درس پردازش و تحلیل تصاویر پزشکی

استاد درس : دکتر فاطمیزاده

نام : محمد سینا حسن نیا

شماره دانشجویی : 96108515

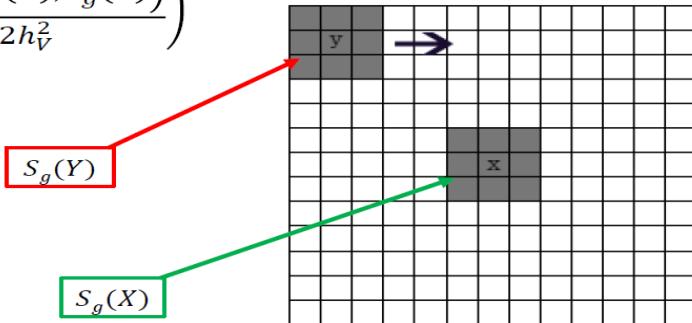
بخش تئوری

سوال 1

الف- روش NLM یکی از روش های Point and Block Estimation است که در آن با در نظر گرفتن پنجره هایی حول پیکسل شناور و مرکزی، به هر پیکسل یک کرنل اختصاص میدهیم که مساله را به فضای ویژگی ببرد (تابع کرنل را در ادامه تعریف میکنیم) و در نهایت بر اساس معیارهایی نوعی شباهت بین دو پنجره مذکور در نظر میگیریم و یک تابع فاصله تعریف میکنیم و برای هر نقطه تخمین بدون نویز را حساب میکنیم:

کرنل:

$$K(g(X), g(Y)) = \exp\left(-\frac{dist(S_g(X), S_g(Y))}{2h_V^2}\right)$$



تخمین:

$$\hat{f}(x, y) = \frac{\sum_{Y \in \Omega} g(Y) K(g(X), g(Y))}{\sum_{Y \in \Omega} K(g(X), g(Y))}$$

از طرفی در روش INLM عملکرد مشابهی داریم و در آن تخمین تصویر حذف نویز شده از ترکیب خطی تصویر(f)، تصویر حذف نویز شده از همان روش NLM ، ($N(f)$) و اعمال NLM دوباره بر روی حذف نویز شده ، ($N(N(f))$) طبق رابطه زیر به دست می آید:

$$u = \frac{1}{4}f + \frac{1}{2}N(f) + \frac{1}{4}N(N(f))$$

همانطور که در نتایج گذاشته شده قابل مشاهده است PSNR تصویر رفع نویز شده در روش جدید از روش NLM بهتر می باشد.

ب- روش Trilateral مشابه Bilateral می باشد به این صورت که از دو ترم گوسی، یکی برای اعمال اثر خاصیت های تشابه مکانی (فاصله) و دیگری برای تاثیر روشنایی استفاده می کند و به خوبی تصاویر رفع نویز می شوند ولی برای رفع مشکل محوشدگی لبه ها، علاوه بر دو ترم قبلی از شباهت های ساختاری هم استفاده میکند تا با یک فیلتر باریک 3×3 لبه ها را حفظ کند. درواقع چون از 3 ویژگی برای تعیین وزن ها استفاده می کند به آن Trilateral میگویند. به طور کلی تر این روش برای نقاط همگن صرفا از یک فیلتر پایین گذر عبور میکنند تا هموار شوند ولی نواحی ناهمگن مثل نزدیک لبه ها با سه ویژگی تشابه ساختاری و روشنایی و مکانی هموار می شوند. در نهایت اگر این دو روش را روی تصاویر با سایز یکسان مقایسه کنیم، این روش عملکرد بهتری نسبت به Bilateral دارد. در مورد فرمول بندی نیز اگر بخواهیم صحبت کنیم روش bilateral filtering می تواند به صورت زیر خلاصه شود (در فرم گستته):

$$\vec{I}^*(\vec{x}) = \frac{1}{k(\vec{x})} \sum_{\vec{\xi} \in \mathcal{N}_{\vec{x}}} \vec{I}(\vec{\xi}) \cdot c(\vec{\xi}, \vec{x}) \cdot s(\vec{I}(\vec{\xi}), \vec{I}(\vec{x}))$$

که در آن :

parameters	description
\vec{x}	spatial coordinates
$\vec{\xi}$	spatial coordinates
\vec{I}	noisy image
\vec{I}^*	filtered image
$\mathcal{N}_{\vec{x}}$	spatial window around the pixel at \vec{x}
$k(\vec{x})$	normalization constant

در Trilateral filtering فرمول بندی روش به صورت زیر است:

$$\vec{I}^{(t+1)}(\vec{x}) = \frac{1}{k(\vec{x})} \sum_{\vec{\xi} \in \mathcal{N}_{\vec{x}}} \vec{I}^{(t)}(\vec{\xi}) \cdot w(\vec{\xi}, \vec{x}, t)$$

که:

$$w(\vec{\xi}, \vec{x}, t) = (1 - a(\vec{x})) \cdot c(\vec{\xi}, \vec{x}) + \\ a(\vec{x}) \cdot c(\vec{\xi}, \vec{x}) \cdot s(\vec{I}^{(t)}(\vec{\xi}), \vec{I}^{(t)}(\vec{x})) \cdot \sum_{i=1}^{D-1} d_i(\vec{\xi}, \vec{x})$$

که در آن :

parameters	description
$a(\vec{x})$	regularized local signal amplitude of the pixel at \vec{x}
D	the dimensionality of the image
d_i Measures the similarity of the rank i local structural orientation between the pixels at $\vec{\xi}$ and \vec{x}	$d_i(\vec{\xi}, \vec{x}) = \exp\left(-\frac{\delta^2(\vec{\xi} - \vec{x}, \hat{e}_i)}{2\sigma^2}\right)$
$\delta(\vec{u}, \vec{v})$	$\cdot \left \frac{\vec{u} \cdot \vec{v}}{\ \vec{u}\ \ \vec{v}\ } \right $

در اینجا باید گفت که در واقع a به نوعی مشخص کننده **homogeneity** می باشد به طوری که هنگامی که a به سمت صفر می رود در واقع محیط **homogeneous** است. در رابطه دلتا نیز در واقع همجهتی دو بردار \vec{u}, \vec{v} بررسی می شود که برای مثال هنگامی که زاویه 90 است $a.b=0$ و در نتیجه مقدار 1 را داریم و در مواقعی که زاویه صفر است مقدار صفر را خواهیم داشت. به عبارت

دیگر، فیلتر Trilateral با وزن دادن به مقدار شدت هر پیکسل در محدوده باریک پنجره فضایی یک تصویر smooth تولید می کند

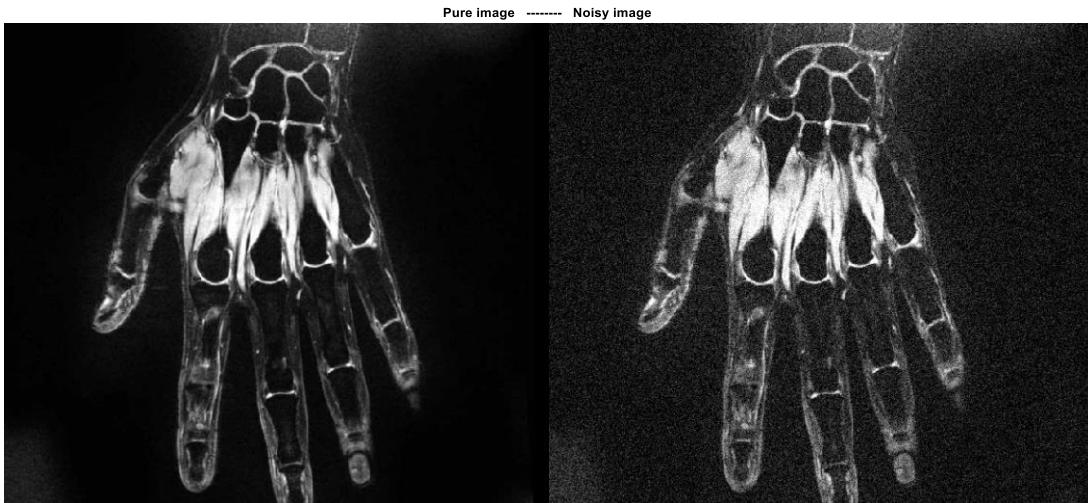
سوال 2

از آن جایی که در رابطه داده شده صرفا نویز فلفل نمکی داریم باید اگر پیکسل 0 یا 1 نویزی نشده باشد، u همان مقدار 7 را نگه دارد و یا گرادیان را کمینه کند. بنابراین نرم باید صفر انتخاب شود؛ چراکه فقط زمانی که $u, 7$ دقیقاً برابر باشد صفر میشود. اگر نویز وجود نداشته باشد $u, 7$ خود را در اکثر مواقع که ترم دوم با توجه به پارامتر لامبدا به صرفه باشد یکسان می کند و اگر هم نویز باشد صفر کردن ترم دوم به شدت گرادیان و هزینه را بالا میبرد و به صرفه نیست که اتفاق بیافتد.

بخش شبیه‌سازی

سوال 1

الف) ابتدا نویز گوسی با میانگین و واریانس ۰.۰۵ و ۰.۰۱ را به تصویر اضافه می‌کنیم. با استفاده از تابع montage داریم:



برای SNR داریم:

SNR is equal to 6.6627

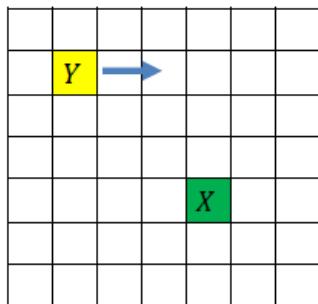
ب) این روش که به صورت کلی از دسته point estimation method است فرمولاسیون کلی به صورت زیر دارد:

$$K(X, Y; g(X), g(Y)) = K(X, Y) = \exp\left(-\frac{\|X - Y\|_2^2}{2h_X^2}\right) = G_{h_X}(\|X - Y\|)$$

که بنابراین خواهیم داشت:

$$\hat{f}(X) = \frac{\sum_{Y \in \Omega} g(Y) \exp\left(-\frac{\|X - Y\|_2^2}{2h_X^2}\right)}{\sum_{Y \in \Omega} \exp\left(-\frac{\|X - Y\|_2^2}{2h_X^2}\right)} = \frac{\sum_{Y \in \Omega} g(Y) G_{h_X}(\|X - Y\|)}{\sum_{Y \in \Omega} G_{h_X}(\|X - Y\|)}$$

عبارت بالا فرم کانولوشنی دارد. در واقع همانطور که در شکل زیر مشخص است ما ∇ را حرکت می دهیم حالا هر چه ∇ دور تر باشد مقدار کمتری خواهیم داشت در نتیجه به نوعی اینجا متوسط گیری محلی انجام می شود. در اینجا ما شاهد linear filtering + Normalization هستیم. ایراد این روش آن است که در واقع به دلیل همان موردی که بالا گفتیم که متوسط گیری محلی انجام می دهد شاهد edge blurring خواهیم بود.



برای حذف نویز با استفاده از این روش داریم :



برای SNR داریم :

SNR is equal to 10.4648

ج) این روش که به صورت کلی از دسته point estimation method است فرمولاسیون کلی به صورت زیر دارد:

$$K(X, Y; g(X), g(Y)) = \exp\left(-\frac{\|X - Y\|_2^2}{2h_X^2}\right) \exp\left(-\frac{\|g(X) - g(Y)\|_2^2}{2h_g^2}\right)$$

$$K(X, Y; g(X), g(Y)) = G_{h_X}(\|X - Y\|) G_{h_g}(\|g(X) - g(Y)\|)$$

که بنابراین خواهیم داشت :

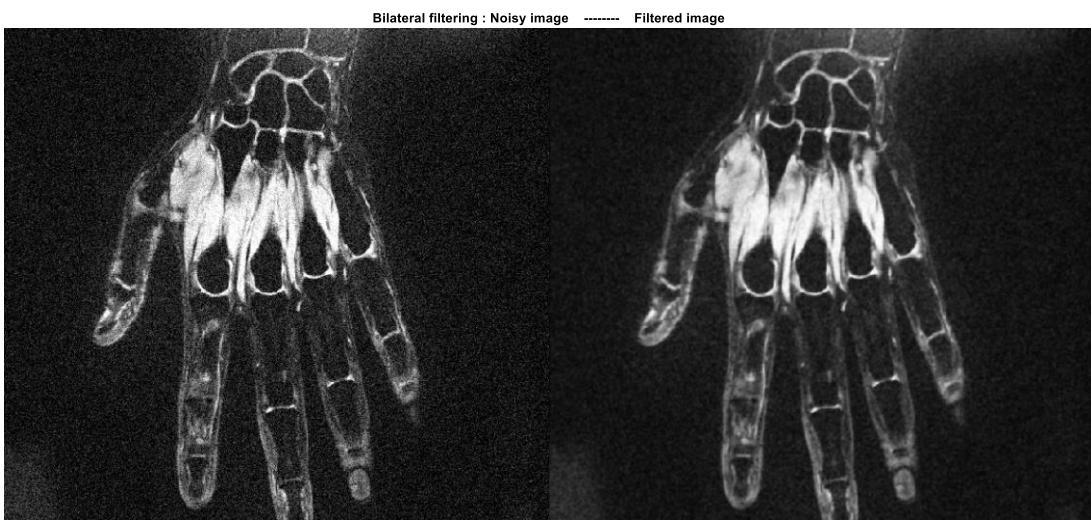
$$\hat{f}(X) = \frac{\sum_{Y \in \Omega} g(Y) G_{h_X}(\|X - Y\|) G_{h_g}(\|g(X) - g(Y)\|)}{\sum_{Y \in \Omega} G_{h_X}(\|X - Y\|) G_{h_g}(\|g(X) - g(Y)\|)}$$

همانطور که میبینیم در این روش 2 ترم گوسی داریم. یکی از آن ها برای اعمال اثر فاصله و دیگری برای اعمال اثر روشنایی است. برای مثال اگر $g(y)$ نسبت به پیکسل مرکزی اختلاف زیاد بود کرنل ما کاهش پیدا می کند و بالعکس. در واقع با این کار نه اجازه می دهیم پیکسل در فاصله دور اثر گذار باشد و نه پیکسل با اختلاف زیاد در سطح روشنایی. در اینجا ما مشاهد non-linear filtering آن است که نسبت که بخش computation cost + computation cost میباشد هستیم. علت اینجا ما شاهد هستیم. علت آن است که نسبت که بخش قبلی که مخرج یکبار محاسبه می شد ولی در این قسمت هر جا که برویم باید دوباره مخرج را محاسبه نماییم.

دو پارامتر برای کنترل پنجره های وزنهای ضرب شده به صورت فیلترهای bilateral h_x, h_g میباشد همانطور که از فرم آن واضح است هرچه مقدار آن بیشتر باشد باعث میشود وزن بیشتری گرفته و فیلتر به اصطلاح از ضربه بودن دورشود. در حالت اهمیت h_x تاثیر گذاری در فیلتر مکانی دارد از انجا که مقدار $|X - Y|_2$ همواره ثابت است با بیشتر کردن مقدار آن باعث میشود که که اختلاف مکانی تاثیری کمتری داشته باشد و همه به یک صورت برخورد کند. در حالت نهایی به نوعی فیلتر میانگین گیر میشود پس در این مولفه قدرت میانگین گیری را مشخص میکنیم هرچه کمتر باشد میانگین گیری ضعیف تری انجام میشود در حالتی که واریانس نویز خیلی بالا باشد زیاد کرد مقدار آن میتواند قدرت میانگین گیری را بیشتر کند به عبارت دیگری فرکانس قطع را پایین تر

میاورد. عبارت h_g برای بررسی و مقایسه اختلاف شدت میباشد و در بخش فیلتر ساختاری اعمال میشود، که تاثیر آن راچقدر تنظیم کند هرچه مقدار آن کمتر باشد حساسیت آن به لبه ها بیشتر میشود و در خروجی مقدار کمتری میدهد تا کرنل لبه را حفظ کند پس بر اساس اهمیت لبه براساس اختلاف شدت باید مقدار آن را کمتر تنظیم کرد پس براساس اینکه تصویر چقدر نویزی و شدت آن چقدر است که در تشخیص لبه بودن چه تاثیری دارد میتوان مقدار آن را تنظیم کرد، به طور مثال با زیاد کردن آن درواقع داریم مجبور میکنیم به ازای اختلافات شدید تر کرنل مقدار کمی بگیرد که در نهایت باعث شود لبه را حذف نکند در حالی که اگر مقدار آن را کم کنیم لبه های ضعیف تر نیز در تصویر حفظ میشوند با تنظیم آن به مقدار مناسب میتوان تمایز بین قدرت نویز و لبه ها حاصل را تعیین کرد. بنابراین به صورت خلاصه h_g را edge و h_x را smooth کنترل کنیم لبه بودن را کنترل می کند. این که برای چه تصاویری نیز چه مقادیری مناسب است در پاراگراف بالا به صورت کامل مورد بحث قرار گرفته است. با توجه به آن چه در بالا گفته شد مقدار $h_g = 1.2$ و $h_x = 0.8$ نویز زدایی مناسبی برای ما انجام می دهد.

برای حذف نویز با استفاده از این روش داریم :

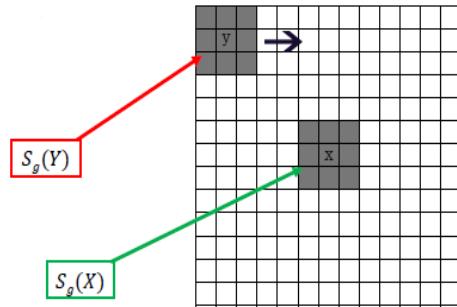


برای SNR داریم :

SNR is equal to 10.5467

۵) این روش که به صورت کلی از دسته point estimation method است فرمولاسیون کلی به صورت زیر دارد:

$$K(g(X), g(Y)) = \exp\left(-\frac{\text{dist}(S_g(X), S_g(Y))}{2h_v^2}\right)$$



که در آن با در نظر گرفتن پنجره هایی حول پیکسل شناور و مرکزی، به هر پیکسل یک کرنل اختصاص میدهیم که مساله را به فضای ویژگی ببرد و در نهایت بر اساس معیارهایی نوعی شباهت بین دو پنجره مذکور در نظر میگیریم و یک تابع فاصله تعریف میکنیم و برای هر نقطه تخمین بدون نویز را حساب میکنیم. برای محاسبه نیز از تابع زیر استفاده می کنیم که در آن برای distance از فاصله اقلیدسی استفاده شده است و برای ماسک از ماسک گوسی استفاده شده است.

```

for i = pi-nsh:pi+nsh
    for j = pj-nsh:pj+nsh
        Iij = Im(i,j);
        B = Im(i-psh:i+psh, j-psh:j+psh);
        B = A-B;
        B = B.^2;
        B = B.*mask;
        wt = sum(B, 'all');
        wt = exp(-wt/sigma2);
        normalising_factor = normalising_factor + wt;
        weighted_sum = weighted_sum + wt*Iij;
    end
end

```

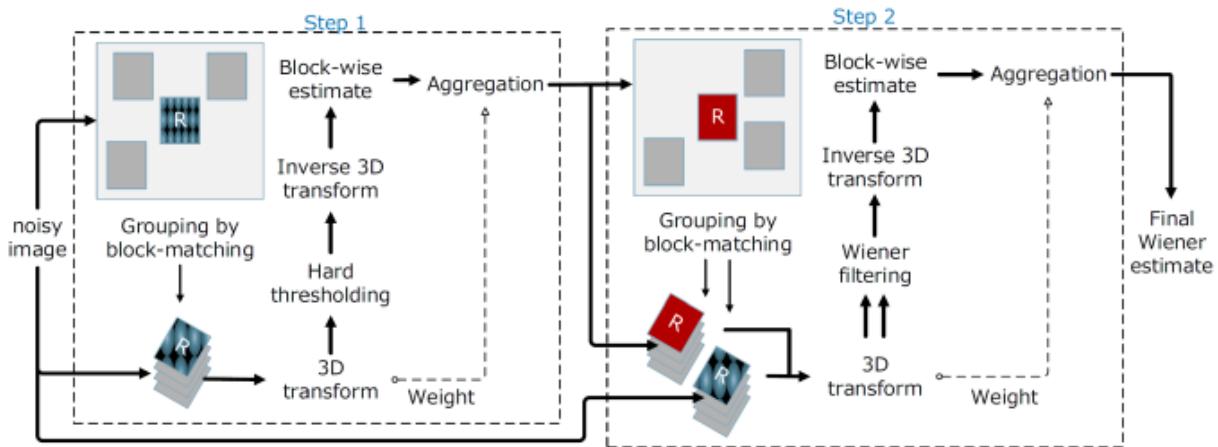
برای حذف نویز با استفاده از این روش داریم :



برای SNR داریم :

SNR is equal to 10.822

۵- امتیازی) تصویر پایین به بهترین نحو عملکرد این روش را بررسی می کند.

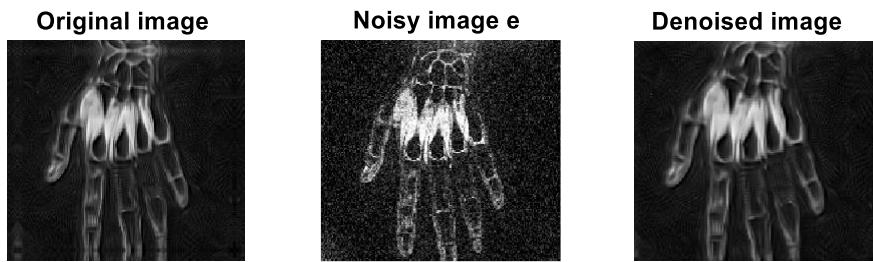


در گام اول به این صورت عمل می کند که برای هر بلاک در تصویر نویزی بلاک هایی را که مشابه بلاک هستند پیدا و آن ها را با هم استک می کند (به صورت 3d) و یک تبدیل 3 بعدی روی آن می زند و hard thr انجام می دهد و در انتها estimate های بلاک ها را به مکان اصلی آن ها برمی گرداند. در مرحله بعدی با استفاده از میانگین گیری وزن دار basic estimate را به دست می آورد. حال در گام بعدی که گام نهایی است با استفاده از basic estimate به دست آمده و

callaborative wiener filtering تصویر دینویز شده را به دست می آورد. گام اول و گام دوم هر کدام به صورت مجزا در کد پیاده شده اند.

برای حذف نویز با استفاده از این روش داریم :

visualization of original,noisy and denoised image



برای SNR داریم :

SNR is equal to 10.5428

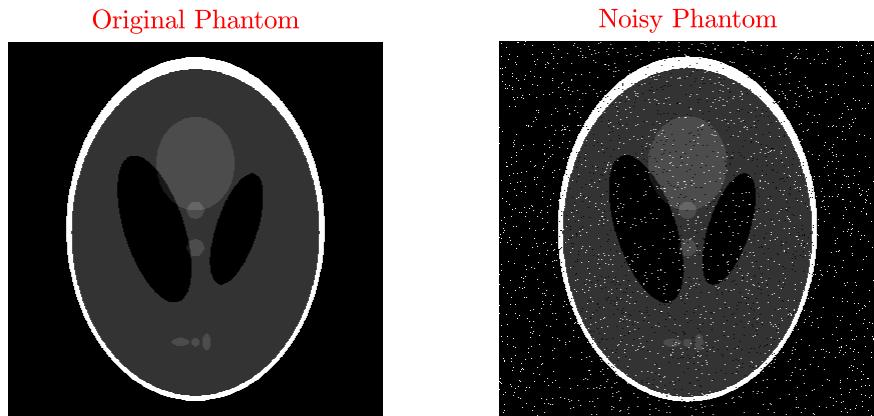
این روش برای تصویری با این تعداد پیکسل complexity بسیار بالایی دارد!! و زمان زیادی برای ران شدن کد طول می کشد.

و) مقادیر SNR را در همان بخش های مورد نظر به دست آورده ایم. همانطور که دیده می شود مقادیر SNR در بخش دازج بیشتر و در بخش جازب بیشتر می باشد که با توجه به موارد گفته شده طبیعیست و به نوعی روش ها از قسمت ب تا د المان های بیشتری را برای دینویز کردن در نظر می گیرند که به صورت کامل مورد بحث قرار گرفت. همچنین عملکرد روش BM3d نیز مطلوب است و عملکرد نسبتا خوبی دارد.

سوال 2

الف) به کمک دستور **phantom** یک فانتوم Modified Shepp Logan به ابعاد 700×700 می سازیم و سپس به آن نویز **salt & peper** با چگالی 0.03 اضافه می کنیم. برای نمایش تصویر نویزی و اصلی داریم :

Visualizzazione of original and noisy Image



ب) تابع داده شده به صورت زیر می باشد:

```
function diff = anisodiff(im, niter, kappa, lambda, option)
```

در تابع بالا **im** تصویر ورودی، **niter**، تعداد **iteration** ها، ضرایب کاپا به نوعی بیانگر کنترast محلی است و در تعیین ضرایب c_N و c_S و c_E و c_W کاربرد دارد. پارامتر بعدی پارامتر لاندا است که در واقع مаксیمم آن برای پایداری 0.25 می باشد. **option** که 2 نوع است در واقع مشخص کننده روش مورد استفاده برای محاسبه c_N و c_E و c_S و c_W می باشد. این موضوع

در پایین به صورت دقیق‌تر توضیح داده می‌شود. برای تعیین c_N و c_E و c_W از روابط زیر استفاده می‌کنیم:

$$\begin{aligned} c_{N_{i,j}}^t &= g(|\nabla_N I_{i,j}^t|) & c_{E_{i,j}}^t &= g(|\nabla_E I_{i,j}^t|) \\ c_{S_{i,j}}^t &= g(|\nabla_S I_{i,j}^t|) & c_{W_{i,j}}^t &= g(|\nabla_W I_{i,j}^t|) \end{aligned}$$

که در آن تابع g می‌تواند یکی از ۲ حالت زیر باشد:

$$g(\nabla I) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2} \quad g(\nabla I) = e^{-(\|\nabla I\|/K)^2}$$

این که از کدام رابطه ضرایب c_N و c_S و c_E و c_W محاسبه شوند درواقع بستگی به این دارد که از کدام یک از g های بالا استفاده شود. در تابع داده شده با انتخاب $\text{option} = 1$ از تابع ۱ و با انتخاب $\text{option} = 2$ از تابع ۲ استفاده شده است.

```

elseif option == 2
    cN = 1./(1 + (deltaN/kappa).^2);
    cS = 1./(1 + (deltaS/kappa).^2);
    cE = 1./(1 + (deltaE/kappa).^2);
    cW = 1./(1 + (deltaW/kappa).^2);
end

```

```

if option == 1
    cN = exp(-(deltaN/kappa).^2);
    cS = exp(-(deltaS/kappa).^2);
    cE = exp(-(deltaE/kappa).^2);
    cW = exp(-(deltaW/kappa).^2);

```

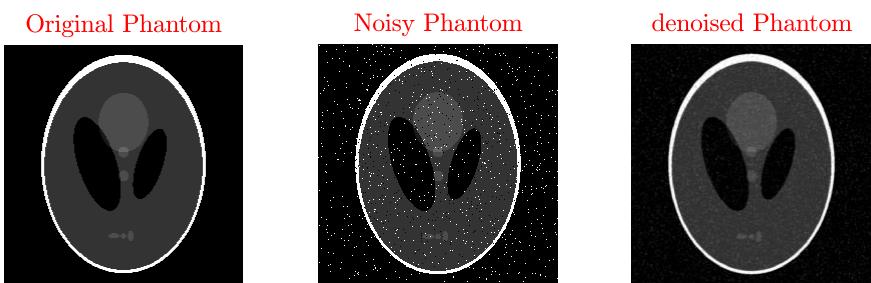
ضریب کا پانیز در به دست آوردن ضرایب بالا نیز تاثیر گذار می باشد.

در مرحله بعد پاسخ عددی Diffusion P.D.E. را بر اساس فرمول ارائه شده در مقاله به دست می‌آوریم:

$$\begin{aligned}
I_{i,j}^{t+1} &= I_{i,j}^t + \lambda [c_N \cdot \nabla_N I + c_S \cdot \nabla_S I \\
&\quad + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I]_{i,j}^t \\
&= I_{i,j}^t (1 - \lambda(c_N + c_S + c_E + c_W)_{i,j}^t) \\
&\quad + \lambda(c_N \cdot I_N + c_S \cdot I_S + c_E \cdot I_E + c_W \cdot I_W)_{i,j}^t \\
&\leq I_{M_{i,j}^t} (1 - \lambda(c_N + c_S + c_E + c_W)_{i,j}^t) \\
&\quad + \lambda I_{M_{i,j}^t} (c_N + c_S + c_E + c_W)_{i,j}^t \\
&= I_{M_{i,j}^t}
\end{aligned} \tag{12}$$

نکته تکمیلی که باید اشاره کرد این است که زمانی که گرادیان در راستا خیلی زیاد است ضریب C صفر می شود و در نتیجه اردر bluring را محدود می کند. حال که به صورت کلی با نحوه عملکرد اینتابع آشنا شدیم و ریاضیات آن را مرور کردیم باید براساس دانسته هایمان مقادیر پارامتر ها را تعیین کنیم. هر چه الگوریتم بیشتر اجرا شود در واقع مقدار blurring تصویر بیشتر می شود بنابراین 5 ایتریشن می تواند گزینه خوبی باشد. (از 5 به بعد تاثیر چندانی در نویز زدایی نداشت) کاپا در فرمول g اثر گذار است و g در تعیین ضرایب c_N و c_E و c_S و c_W بنابراین مطابق نکته تکمیلی انتخاب کاپا مناسب از blur شدن edge ها جلوگیری می کند بنابراین کاپا را برابر 50 قرار می دهیم و می تواند گزینه خوبی باشد. لاندا نیز برای پایداری می تواند حداقل 0.25 باشد. (به نوعی تعیین کننده سرعت آپدیت در زمان است) بنابراین آن را برابر 0.11 قرار می دهیم. همچنین استفاده از option1 با option2 تفاوت چندانی نداشت. اما به دلیل فرم نمایی از option1 استفاده کردیم. بنابراین داریم :

Visualizatoin of original and noisy Image



ج) ابتدا برای معیار EPI توضیح می دهیم.

بیشتر روش های مقایسه تصویر مانند MSE, SNR, SSIM کلیت محتوای تصویر را با یکدیگر مقایسه میکنند و به مشابهت محتوای لبه ها توجه نمیکند برای همین نیز روش هایی مانند EPI ارائه شده است تا به مقایسه این خاصیت بپردازد. فرمول کلی این روش به صورت زیر است:

$$EPI = \frac{\Gamma(\Delta s - \bar{\Delta}s, \widehat{\Delta}s - \bar{\Delta}s)}{\sqrt{\Gamma(\Delta s - \bar{\Delta}s, \Delta s - \bar{\Delta}s) \cdot \Gamma(\widehat{\Delta}s - \bar{\widehat{\Delta}s}, \widehat{\Delta}s - \bar{\Delta}s)}}$$

$$\Gamma(s_1, s_2) = \sum_{i,j \in ROI} s_1(i,j) \cdot s_2(i,j)$$

که در آن $\bar{\Delta}s$ و $\bar{\widehat{\Delta}s}$ نسخه بالاگذر شده منطقه مورد نظر ما (region of interest) میباشد که در حالات خاصی میتواند برای این امر باشد که ما در حال پردازش بخش خاصی از تصویر باشیم ولی در مورد ما کل تصویر میباشد. $\bar{\Delta}s$ برای تصویر اصلی و $\bar{\widehat{\Delta}s}$ برای حالت تبدیل یافته تصویر است. برای تولید آن از فیلتر لاپلاسین که با کرنل ۳ در ۳ تولید شده است استفاده میکنند. مقادیر $\bar{\Delta}s$, $\bar{\widehat{\Delta}s}$ مقادیر میانگین در این دو حالت است پس به صورت کلی عمل را بخواهیم توصیف کنیم فرم بالاگذر سیگنال ها (به بیان دیگر محتوای فرکانس بالا که شامل لبه ها است) تولید شده و میانگین صفر میشود تا شدت روشنایی تاثیر گذار نباشد) هرچند یک فیلتر بالاگذر مناسب باید این کار را انجام دهد!) سپس با ضرب نقطه ای این دو یک امتیاز بدست میاورد اگر کورولیشن این دو بالا باشد علامت و مقدار یکسانی دارند پس زمانی که به اندازه تک تکشان نرمالیزه میشوند خروجی نزدیک به یکی دارند ولی اگر فرم رفتار یکسان نباشد و کورولیشن بین این دو پایین باشد ضرب دو ترم در هم به اندازه کافی بزرگ نمیشود و وقتی نرمالیزه میشود (به اندازه این ضرب برای هر کدام به صورت جدا که در مخرج انجام میشود) انگاه عبارت به صفر و صفر نزدیک تر میشود. مشکل اصلی

این روش استفاده از لاپلاسین برای تولید فیلتر بالاگذر است که به شدت به نویز حساس میباشد و ممکن است خطاهایی برای مقایسه ایجاد کند.(خصوصا اگر تصویر پر از نویز باشد!)

برای محاسبه EPI و SNR داریم :

```
EPI error = 0.24712
SNR = 13.4624
```

همانطور که مشخص است معیار EPI برابر 0.24 می باشد می دانیم هر چه قدر لبه های تصویر اصلی و دینویز شده کورلیشن بیشتری داشته باشند این معیار بالاتر است بنابراین چندان در لبه موفق نبودیم. همچنین SNR علی رغم افزایشی که داشته است نسبت به روش سوال 3 چندان بالا نیست ولی به صورت کلی افزایش یافته است و مناسب است.

سوال 3

الف) با داشتن تصویر ورودی u_0 تصویری که تا حد امکان به تصویر u_0 نزدیک باشد از رابطه زیر می تواند محاسبه شود:

$$\min_u \int_{\Omega} |u - u_0|^2 dx$$

این مسئله یک مسئله موسوم **ill-posed** است بنابراین یک راه برای حل آن این است که یک ترم TV regularization به آن اضافه شود. یک مدل برای انجام این کار مدل

است. بنابراین مسئله به صورت زیر است :

$$J(u) = \min_u \left\{ \int_{\Omega} |u - u_0|^2 dx + \lambda \int_{\Omega} |\nabla u| dx \right\}$$

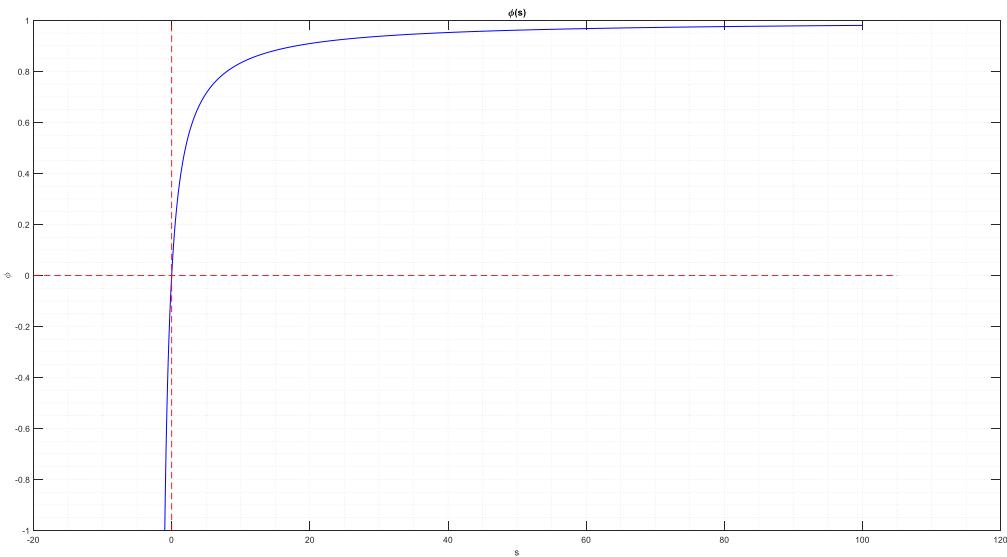
ترم اول در واقع فاصله بین تصویر u_0 , u را مینیمم می کند و ترم دوم یک ترم **smooth** کننده است. اگرچه مدل بالا کارآمد است اما مدل TV regularization نمی تواند بعضی لبه های

جزئیات را به دلیل اثر isotropic آن تشخیص دهد. برای حل این مشکل غیر محدب زیر پیشنهاد شد:

$$\int_{\Omega} \varphi(|\nabla u|) dx$$

Parameters	Description
$\varphi(s)$	$\alpha s / (1 + \alpha s)$
α	positive parameter

تابع $\varphi(s)$ به صورت زیر می باشد:

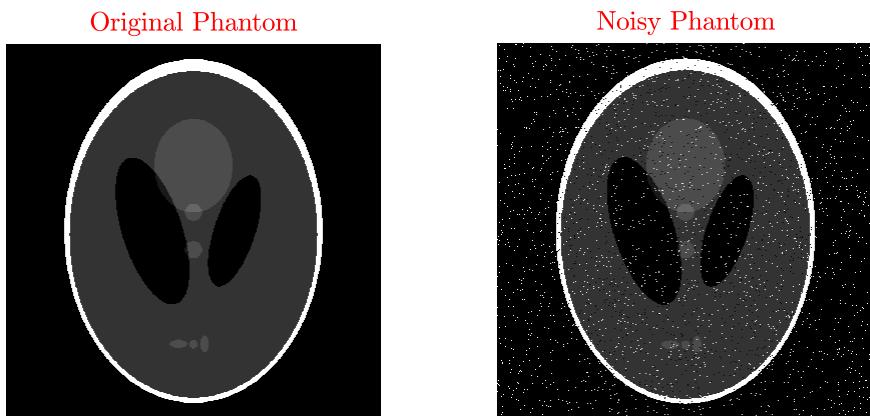


بنابراین همانطور که در شکل مشخص شده است این تابع حول صفر رفتار خطی دارد اما هنگامی که گرادیان زیاد می شود **saturate** می شود در این فرم معادلات زمانی که لبه داشته باشیم با توجه به آن که در گرادیان بالا **saturate** می شود برعکس حالت قبلی که هنگامی که گرادیان زیاد می شود (لبه) تابع نیز زیاد می شد لبه ها را بهتر حفظ می کند. **regularizer** غیر محدب در حفظ لبه ها کارآمد است، اما به دلیل ویژگی غیر محدب آن، نویز با شدت بالا به جای حذف شدن، احتمالاً در یک ناحیه همگن از تصویر حفظ می شود، زیرا به عنوان جزئیات واقعی در نظر

گرفته می شود. اما *TV Regularizer* می تواند نویز را به طور موثر حذف کند. به منظور ترکیب هر دو ویژگی آنها، *Regularize TV Regularizer* غیر محدب را در یک مدل یکپارچه ترکیب می کنیم. بنابراین فرم مسئله به صورت زیر می شود:

$$\min_u \left\{ \int_{\Omega} |u - u_0|^2 dx + \varepsilon \int_{\Omega} |\nabla u| dx + (1-\varepsilon) \int_{\Omega} \varphi(|\nabla u|) dx \right\}$$

Visualizzazione of original and noisy Image



ب) فانکشن داده شده به صورت زیر می باشد:

```
function newim = TVL1denoise(im, lambda, niter)
```

تابع مورد نظر ۳ ورودی می گیرد. ورودی اول تصویر ورودی است. پارامتر لاندا یک پارامتر **niter** است. هر چه مقدار آن کمتر تصاویر **smooth** تری تولید خواهد کرد. **regularization** تعداد **iteration** ها است. پارامتر **lt** از روی لاندا محاسبه می شود.

```
lt=lambda*tau;
```

نحوه عملکرد تابع به این صورت است که ابتدا تصویر ورودی نرمالیزه می شود. سپس پارامتر P مشتق های تصویر نرمالایز شده به صورت اختلاف تصویر و شیفت تصویر به دست می آید.

```
p(:, :, 1)=u(:, [2:width, width]) - u;
p(:, :, 2)=u([2:height, height], :) - u;
```

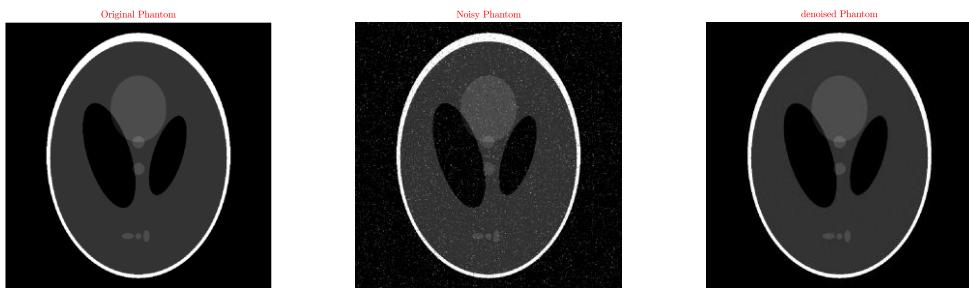
حال گرادیان های مربوطه محاسبه و با استفاده از نرم اول div آپدیت می شوند.

```
ux=u(:, [2:width, width]) - u;
uy=u([2:height, height], :) - u;
p= p + sigma*cat(3, ux, uy);
% project
normep = max(1, sqrt(p(:, :, 1).^2 + p(:, :, 2).^2));
p(:, :, 1)=p(:, :, 1)./normep;
p(:, :, 2)=p(:, :, 2)./normep;

% shrinkage
% compute divergence in div
div=[p([1:height-1], :, 2); zeros(1, width)] - [zeros(1, width); p([1:height-1], :, 2)];
div=[p(:, [1:width-1], 1) zeros(height, 1)] - [zeros(height, 1) p(:, [1:width-1], 1)] + div;
```

تعداد iteration را برابر 100 و مقدار λ را برابر 1 قرار می دهیم. نتیجه به صورت زیر است:

Visualizatoin of original and noisy and denoised Image(using TVL1denoise)



ج) معیار EPI و SNR را محاسبه می کنیم. نتیجه به صورت زیر می باشد. همانطور که دیده می شود خروجی حاصل از این روش عملکرد به شدت بهتری در زمینه SNR و همچنین در زمینه EPI دارد. همانطور که دیده می شود مرز نواحی نیز به صورت بهتر مشخص شده است و معیار EPI نیز همین موضوع را تایید می کند زیرا نزدیک به یک است.

EPI error = 0.93344

SNR = 25.2059