

## بخش تئوری

سوال ۱: الف) در الگوریتم NLM، همان طور که از اسم آن نیز مشخص است، برای تعیین وزن مشارکت پیکسل‌های تصویر در بازسازی پیکسل دلخواه  $(i, j)$ ، بجای فاصله مکانی پیکسل‌ها از یکدیگر و یا اختلاف شدت روشنایی آن‌ها، از فاصله دیگری بر مبنای میزان شباهت پنجره‌های اطراف آن‌ها استفاده می‌شود. یعنی داریم:

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

بر اساس نحوه تعریف فاصله مذکور، روش‌های متعددی به وجود آمده‌اند. در ساده‌ترین روش NLM، میانگین بدون وزن یا وزن دار گوسی، مربع اختلاف شدت روشنایی پیکسل‌های متناظر در پنجره‌ها، به عنوان معیار فاصله در نظر گرفته شده است. شایان ذکر است در روش‌های NLM، هر چه اندازه پنجره بزرگ‌تر باشد، دقت سنجش شباهت بین دو پیکسل بیشتر است، اما میزان محاسبات نیز بیشتر خواهد بود.

روش پیشنهادی مقاله مذکور به شرح زیر است:

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

در واقع در این روش، بجای اینکه تنها از خروجی حاصل از یک مرتبه اعمال فیلتر NLM استفاده شود، میانگین وزن داری از تصویر ورودی، خروجی حاصل از یک مرتبه اعمال فیلتر NLM و خروجی حاصل از دو مرتبه اعمال فیلتر NLM، به عنوان تصویر خروجی مد نظر قرار می‌گیرد.

ب) روش Trilateral Filtering، علاوه بر شباهت‌های مکانی و شدت روشنایی که در روش Bilateral Filtering مورد استفاده قرار می‌گیرند، شباهت ساختاری محلی را نیز مد نظر قرار می‌دهد، تا هم‌زمان با حفظ لبه، تصاویر را در یک پنجره مکانی کوچک، هموار کند. در واقع، در این روش اطلاعات ساختاری محلی برای تعیین میزان ناهم‌ریختی تصاویر استفاده می‌شود. به طوری که از یک طرف، در نواحی هم‌ریخت، تنها یک فیلتر پایین‌گذر گوسی بر روی تصاویر اعمال می‌شود و از طرف دیگر در نواحی ناهم‌ریخت، هموارسازی با در نظر گرفتن شباهت‌های مکانی، شدت روشنایی و جهت‌گیری ساختاری محلی صورت می‌گیرد.

فرمول‌بندی این روش به صورت زیر است:

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

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

در عبارت بالا  $a(\vec{x}) \in [0,1]$  اندازه تنظیم شده سیگنال محلی پیکسل واقع در مکان  $\vec{x}$  است. برای نواحی هم‌ریخت، این مقدار به صفر میل می‌کند و تنها عبارت  $c(\vec{\xi}, \vec{x})$  تاثیرگذار خواهد بود. اما در نواحی ناهم‌ریخت، این مقدار زیاد می‌شود و شباهت‌های شدت روشنایی و جهت‌گیری ساختاری محلی نیز اهمیت می‌یابند.

همچنین  $d_i(\vec{\xi}, \vec{x})$  شباهت جهت‌گیری ساختاری محلی مرتبه  $i$  بین پیکسل‌های واقع در مکان‌های  $\vec{\xi}$  و  $\vec{x}$  را می‌سنجد.

$$d_i(\vec{\xi}, \vec{x}) = \exp\left(-\frac{\delta^2(\vec{\xi} - \vec{x}, \hat{e}_i)}{2\sigma^2}\right)$$

$$\text{where} \quad \delta(\vec{u}, \vec{v}) = 1 - \left| \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|} \right|$$

در عبارت بالا  $\hat{e}_i$  راستای جهت‌گیری ساختاری محلی مرتبه  $i$  است.

سوال ۲: به ازای مقدار صفر برای  $p$ ، رابطه مورد نظر جهت حذف نویز نمک-فلفل مناسب خواهد بود؛ زیرا در این حالت داریم:

$$u = \operatorname{argmin}_u \left( \int_{\Omega} |\nabla u| \, dx + \lambda \int_{\Omega} \|u - v\|_0 \, dx \right)$$

از آن جایی که مقدار  $\|u - v\|_0$  به ازای هر پیکسل برابر صفر یا یک است، جواب مسئله فوق در هر تکرار دو حالت ممکن خواهد داشت.

اگر مقدار  $|\nabla u|$  کوچک باشد،  $u^*$  مساوی  $v$  می‌شود و اگر مقدار  $|\nabla u|$  بزرگ باشد،  $u^*$  برابر میانگین چهار پیکسل اطراف خود خواهد بود. در نتیجه در نقاطی که نویز وجود ندارد، تغییری در مقدار پیکسل ایجاد نمی‌شود؛ اما در نقاطی که در اثر نویز نمک-فلفل، صفر یا یک شده‌اند، مقدار پیکسل با میانگین چهار پیکسل اطراف، جایگزین می‌شود.

سوال ۱: الف) خروجی به صورت زیر بدست می‌آید:

without noise and noisy grayscale images



ب) در Classical Regression Filtering، برای تعیین وزن مشارکت پیکسل‌های تصویر در بازسازی پیکسل دلخواه  $(i,j)$ ، تنها فاصله پیکسل‌ها از یکدیگر مد نظر قرار می‌گیرد؛ در نتیجه هر چه یک پیکسل به پیکسل مورد نظر نزدیک‌تر باشد، وزن بیشتری در بازسازی آن پیکسل خواهد داشت. در این روش فرمول حذف نویز عبارت است از:

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

$h_x$  شدت افت وزن گوسی با افزایش فاصله را تعیین می‌کند. به عبارت بهتر، هر چه این مقدار بزرگ‌تر باشد، پیکسل‌های بیشتری در بازسازی پیکسل مورد نظر، تاثیرگذار خواهند بود.

خروجی به صورت زیر بدست می‌آید:

Classical Regression denoised image



ج) در Bilateral Filtering، برای تعیین وزن مشارکت پیکسل‌های تصویر در بازسازی پیکسل دلخواه  $(i,j)$ ، علاوه بر فاصله پیکسل‌ها از یکدیگر، اختلاف شدت روشنایی آن‌ها نیز مد نظر قرار می‌گیرد؛ در نتیجه در بازسازی پیکسل‌های لبه، تنها از پیکسل‌های طرفی از لبه استفاده می‌شود که پیکسل مورد نظر در آن قرار دارد. در اثر این امر، ضمن حذف نویز تا حد خوبی از تار شدن لبه‌ها جلوگیری می‌شود. در این روش فرمول حذف نویز عبارت است از:

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

همچون قسمت قبل،  $h_x$  شدت افت وزن گوسی با افزایش فاصله را تعیین می‌کند. به عبارت بهتر، هر چه این مقدار بزرگتر باشد، پیکسل‌های بیشتری در بازسازی پیکسل مورد نظر، تاثیرگذار خواهند بود. به طور مشابه، کاربرد  $h_g$  کنترل شدت افت وزن گوسی با افزایش اختلاف روشنایی پیکسل‌ها است.

این مقادیر باید بر اساس نوع تصویر و آستانه‌های مد نظر کاربر تعیین شوند؛ با توجه به اینکه تابع گوسی به ازای ورودی  $3\sigma$ ، تقریباً صفر می‌شود، برای داشتن یک فیلتر گوسی  $9 \times 9$  یا  $11 \times 11$ ، مقدار ۱ برای پارامتر  $h_x$  مناسب خواهد بود. همچنین اگر آستانه اختلاف روشنایی در حدود  $0.2$  باشد، مقدار پارامتر  $h_g$  تقریباً  $0.4$  بدست می‌آید (شدت روشنایی به بازه صفر تا یک تصویر شده است).

برای تصاویر یکنواخت، که سرعت تغییر شدت روشنایی آن‌ها کم است، با انتخاب مقادیر بزرگ برای  $h_x$ ، می‌توان تعداد پیکسل‌های تاثیرگذار در بازسازی پیکسل مورد نظر و در نتیجه کیفیت نویززدایی را افزایش داد؛ اما برای تصاویر غیریکنواخت، انتخاب مقادیر کوچک مناسب‌تر است.

در مورد  $h_g$ ، در صورتی که لبه‌های تصویر تیز باشند (اختلاف شدت روشنایی در دو طرف لبه زیاد باشد)، انتخاب مقادیر بزرگ باعث در بر گرفتن پیکسل‌های بیشتر برای بازسازی و داشتن خروجی بهتر می‌شود؛ اما اگر چنین نباشد، باید مقادیر کوچک‌تری انتخاب شوند تا از تار شدن لبه‌های تصویر جلوگیری شود.

خروجی به صورت زیر بدست می‌آید:

Bilateral denoised image



د) در الگوریتم NLM، همان طور که از اسم آن نیز مشخص است، برای تعیین وزن مشارکت پیکسل‌های تصویر در بازسازی پیکسل دلخواه  $(i,j)$ ، بجای فاصله مکانی پیکسل‌ها از یکدیگر و یا اختلاف شدت روشنایی آن‌ها، از فاصله دیگری بر مبنای میزان شباهت پنجره‌های اطراف آن‌ها استفاده می‌شود. در این روش فرمول حذف نویز عبارت است از:

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

بر اساس نحوه تعریف فاصله مذکور، روش‌های متعددی به وجود آمده اند. در ساده‌ترین روش NLM، میانگین بدون وزن یا وزن دار گوسی، مربع اختلاف شدت روشنایی پیکسل‌های متناظر در پنجره‌ها، به عنوان معیار فاصله در نظر گرفته شده است. شایان ذکر است در روش‌های NLM، هر چه اندازه پنجره بزرگ‌تر باشد، دقت سنجش شباهت بین دو پیکسل بیشتر است، اما میزان محاسبات نیز بیشتر خواهد بود.

خروجی به صورت زیر بدست می‌آید:



ه) الگوریتم BM3D از دو بخش تشکیل شده است.

در بخش اول مشابه با الگوریتم NLM، با تعریف معیاری برای سنجش شباهت، برای پنجره اطراف هر پیکسل، تعدادی پنجره مشابه در تصویر انتخاب می‌شود. در ادامه این پنجره‌ها به صورت یک پشته وارد یک تبدیل خطی مانند تبدیل Wavelet می‌شوند و سپس ضرایب کوچک، به عنوان مولفه‌های نویز، بر اساس یک آستانه سخت یا نرم حذف می‌شوند. در آخر با اعمال تبدیل خطی معکوس، حدس اولیه‌ای از تصویر بدون نویز بدست می‌آید.

در مرحله دوم عمل مشابهی برای یافتن پنجره‌های شبیه به پنجره اطراف هر پیکسل، در تصویر خروجی قسمت قبل، انجام می‌شود؛ سپس پنجره‌های متناظر در تصویر نویزی انتخاب می‌شوند و با استفاده از پنجره‌های بدست آمده از تصویر خروجی قسمت قبل برای تخمین  $P_{FF}$ ، فیلتر wiener مناسبی بر روی آن‌ها اعمال می‌گردد.

شایان ذکر است در هر مرحله از سیستم وزن‌دهی خاصی برای تجمیع خروجی‌های حاصل از پیکسل‌های مختلف و رسیدن به یک خروجی واحد استفاده می‌شود.

خروجی به صورت زیر بدست می آید:



و) خروجی به صورت زیر بدست می آید:

Noisy Image SNR = 8.4646

Classical Regression SNR = 9.0122

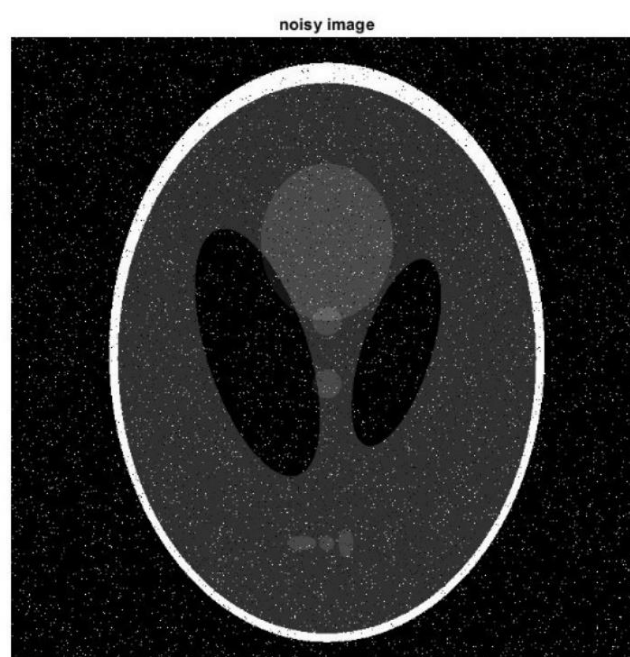
Bilateral SNR = 9.1819

NLM SNR = 10.5766

BM3D SNR = 10.5898

همان طور که انتظار می رفت، خروجی هر قسمت نسبت به قسمت قبل بهبود یافته است.

سوال ۲: الف) خروجی به صورت زیر بدست می آید:



ب) تابع anisotropic، پنج ورودی دارد. ورودی اول تصویر نویزی، ورودی دوم تعداد تکرار الگوریتم، ورودی سوم ضریب  $K$ ، ورودی چهارم ضریب  $\lambda$  و ورودی پنجم روش محاسبه ضرایب اثرگذاری مشتق جهت‌های مختلف در تعیین مقدار جدید است. در این تابع به تعداد ورودی تعیین کننده تعداد تکرار الگوریتم، مراحل زیر انجام می‌شوند:

ابتدا مشتق‌های بالا، پایین، چپ و راست محاسبه می‌شوند. سپس بر اساس ورودی تعیین کننده روش محاسبه ضرایب اثرگذاری مشتق جهت‌های مختلف، ورودی  $K$  و همچنین مقدار مشتق در هر جهت، ضریب اثرگذاری آن مشتق در تعیین مقدار جدید بدست می‌آید.

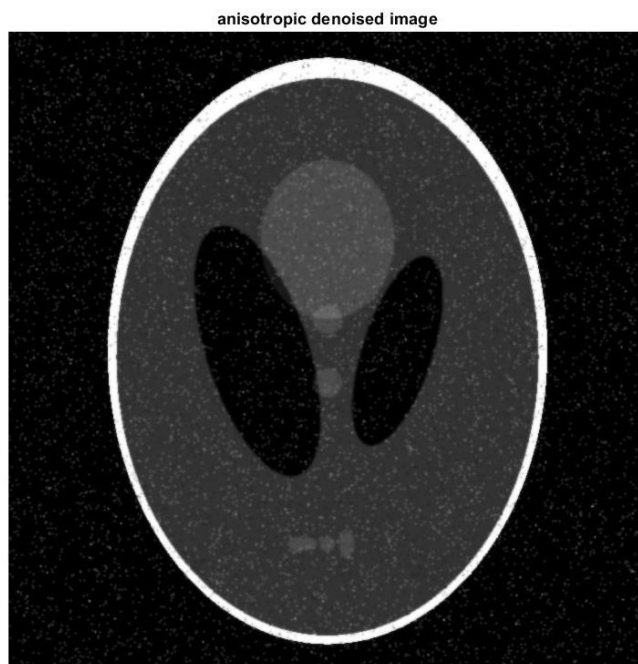
$$\text{way 1: coefficient} = \exp\left(-\left(\frac{\text{derivative}}{\kappa}\right)^2\right), \text{way 2: coefficient} = \frac{1}{\left(1 + \frac{\text{derivative}}{\kappa}\right)^2}$$

در آخر بر اساس ورودی  $\lambda$  مقدار جدید از روی مشتق‌های جهت‌های مختلف و ضریب اثرگذاری هر یک تولید می‌شود.

$$u^{i+1} = u + \lambda(cN. (\nabla_u)_N + cS. (\nabla_u)_S + cE. (\nabla_u)_E + cW. (\nabla_u)_W)$$

به روش آزمون و خطا مقادیر مناسب و خروجی به صورت زیر بدست می‌آیند:

$$\text{number of iterations} = 10, \kappa = 0.5, \lambda = 0.1, \text{method number} = 2$$



معیار Edge Preserving index (EPI) به منظور ارزیابی میزان حفظ لبه‌های تصویر بعد از حذف نویز استفاده می‌شود. برای محاسبه این معیار، ابتدا با استفاده از فیلتر لاپلاسین، تخمینی از میزان لبه در هر پیکسل تصویر نویزی و تصویر خروجی بدست می‌آید و سپس ضریب همبستگی این مقادیر محاسبه می‌شود. به وضوح این مقدار همواره عددی بین  $-1$  تا  $1$  است و هر چه به  $1$  نزدیک‌تر باشد، الگوریتم مورد نظر در حفظ لبه‌های تصویر عملکرد بهتری داشته است.

خروجی به صورت زیر بدست می‌آید:

$$EPI = 0.6735$$

$$SNR = 15.0004$$

سوال ۳: در روش‌های مبتنی بر فیلتر غیرخطی، با تعریف مسائل بهینه‌سازی مناسب، سعی می‌شود ضمن حذف نویز از تار شدن لبه‌های تصویر جلوگیری شود. برای مثال در روش ROF داریم:

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

در این حالت اگر مقدار  $|\nabla u|$  کوچک باشد، میانگین وزن داری از مقدار پیکسل‌های اطراف پیکسل مورد نظر و اگر مقدار  $|\nabla u|$  بزرگ باشد، مقداری نزدیک به مقدار قبلی، به عنوان تخمین جدید مقدار پیکسل مورد نظر بدست می‌آید. در نتیجه نکته اصلی در روش‌های مذکور، رفتار غیرخطی بر اساس مقدار  $|\nabla u|$  یا عبارتی مشابه است.

یکی از مشکلات مربوط به روش ROF، این است که این روش تمایزی میان لبه‌های ساختاری و تغییرات کوچک ناشی از نویز قائل نمی‌شود. پیشنهاد مقاله مذکور، اضافه کردن یک ترم غیرخطی برای حل این مشکل است. فرم مسئله بهینه‌سازی جدید به صورت زیر خواهد بود:

$$J(u) = \operatorname{argmin}_u \left( \int_{\Omega} |u - u_0|^2 dx + \varepsilon \int_{\Omega} |\nabla u| dx + (1 - \varepsilon) \int_{\Omega} \varphi(|\nabla u|) dx \right)$$

$$\text{where } \varphi(s) = \frac{\alpha s}{1 + \alpha s}$$

در اثر اضافه کردن این ترم، به ازای مقادیر کوچک  $|\nabla u|$  تغییر چندانی ایجاد نمی‌شود؛ اما با بزرگ شدن  $|\nabla u|$ ، تخمین جدید مقدار پیکسل مورد نظر، با سرعت بیشتری به سمت مقدار قبلی آن میل می‌کند. زیرا در این حالت تابع  $\varphi(|\nabla u|)$  تقریباً اشباع می‌شود. در نتیجه این روش ضمن حذف نویز، عملکرد بهتری در حفظ لبه‌های تصویر دارد.

ب) تابع TV1denoise، سه ورودی دارد. ورودی اول تصویر نویزی، ورودی دوم ضریب  $\lambda$  و ورودی سوم تعداد تکرار الگوریتم است.

در این تابع به تعداد ورودی تعیین کننده تعداد تکرار الگوریتم، مراحل زیر انجام می‌شوند:

ابتدا مشتق در راستاهای  $X$  و  $Y$  محاسبه می‌شوند، سپس بر اساس پارامتر  $\sigma$  و مقادیر مشتق در راستاهای  $X$  و  $Y$  این مرحله و مقادیر مشتق اصلاح شده در راستاهای  $X$  و  $Y$  مرحله قبل، مقادیر مشتق اصلاح شده در راستاهای  $X$  و  $Y$  این مرحله بدست می‌آیند.

adjusted derivative in level  $i = \text{adjusted derivative in level } (i - 1) + \sigma$ . (derivative in level  $i$ )

در ادامه با استفاده از این مقادیر، مقدار دیورژانس در هر پیکسل محاسبه می‌شود و سپس با مقایسه مقدار دیورژانس در هر پیکسل با ضریب  $\lambda$ ، تخمین جدید مقدار پیکسل مورد نظر، بر اساس تابعی سه ضابطه‌ای بدست می‌آید.

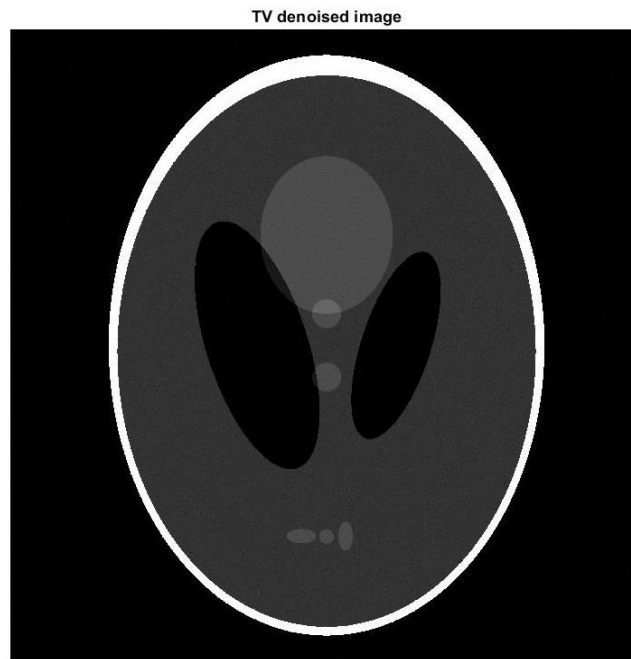
در آخر از اختلاف تصویر بدست آمده و تصویر مرحله قبل، برای محاسبه تصویر نهایی این مرحله، به صورت زیر، استفاده می‌شود.

final  $u$  in level  $i = \text{calculated } u \text{ in level } i + \theta$ . (calculated  $u$  in level  $i - \text{final } u \text{ in level } (i - 1)$ )

در این قسمت برای استفاده از تابع TV1denoise، بر اساس صورت تمرین، تعداد تکرار برابر 100 و مقدار  $\lambda$  مساوی 1 در نظر گرفته شده است.



خروجی به صورت زیر بدست می آید:



(ج) خروجی به صورت زیر بدست می آید:

$$\text{EPI} = 0.9371$$

$$\text{SNR} = 25.3993$$

به وضوح عملکرد روش Total Variation بر اساس هر دو معیار بهتر است.