

Wavelet

علیرضا لرستانی

چکیده	اطلاعات گزارش
تبدیل موجک یا Wavelet یکی دیگر از ابزارهای کاربردی پردازش تصاویر است که برخی از محدودیت‌هایی که روش‌هایی که تا کنون یاد گرفتیم داشتند را مرتفع می‌سازد. در این تمرین با این تبدیل و استفاده‌های مختلف آن آشنا می‌شویم و متدهای گوناگونی که اساس آن‌ها همین تبدیل موجک است را با هم بررسی می‌کنیم.	تاریخ: 1399/10/21
	واژگان کلیدی: Wavelet Transform WT Pyramid هرم Quantize FT تبدیل فوریه STFT تبدیل فوریه زمان کوتاه Denoising Gaussian Pyramid Laplacian Pyramid

۱-مقدمه

برابر با مقیاس آن است. تبدیل موجک تجزیه یک تابع بر مبنای توابع موجک می‌باشد. موجک‌ها (که به عنوان موجک‌های دختر شناخته می‌شوند) نمونه‌های انتقال یافته و مقیاس شده یک تابع (موجک مادر) با طول متناهی و نوسانی شدیداً میرا هستند.

نوشتار حاضر، به بررسی تبدیل موجک و پیاده‌سازی (با زبان برنامه نویسی متلب) تعدادی عملیات پردازش تصویر بر اساس این ایده می‌پردازد.

در بخش‌های گذشته با ایده تبدیل فوریه آشنا شدیم و با استفاده از آن تکنیک‌های پردازشی گوناگونی را بررسی کردیم. با همه‌ی این اوصاف تبدیل فوریه یا FT محدودیت‌هایی داشت که در ادامه به بررسی آن‌ها می‌پردازیم. ایده اولیه در شکل‌گیری تبدیل موجک یا WT مرتفع‌سازی این محدودیت‌هاست. به طور کلی موجک دسته‌ای از توابع ریاضی هستند که برای تجزیه سیگنال پیوسته به مؤلفه‌های فرکانسی آن بکار می‌رود که رزولوشن هر مؤلفه

۲-شرح تکنیکال

به عبارت دیگر:

تبدیل:

- تبدیل فوریه یا Fourier Transform اطلاعات

موجود در تصویر (what) را بیان می کند، اما برای پرسش نسبت به محل وقوع (where) پاسخی را ارائه نمی کند.

- بیان تصویر در حوزه مکان یا spatial به شما مکان وقوع (where) را می دهد، ولی نمیدانید که آنجا چه اتفاقی افتاده (what).

در نتیجه ما به بیانی برای تصویر احتیاج داریم که به ما بگوید چه چیزی در تصویر، و کجا اتفاق افتاده است. در این جا ایده تبدیل فوریه زمان کوتاه مطرح شد.

مراحل تبدیل فوریه زمان کوتاه:

- 1- یک تابع پنجره با طول محدود انتخاب کنید.
- 2- پنجره را در زمان $t = 0$ بر روی سیگنال قرار دهید.
- 3- سیگنال را به کمک این پنجره برش بزنید.
- 4- روی سیگنال برش خورده FT را محاسبه کرده و نتیجه را ذخیره کنید.
- 5- پنجره را به مقدار کمی به سمت راست بلغزانید
- 6- به مرحله 3 بروید تا اینکه به انتهای سیگنال برسید.

STFT اطلاعات زمانی را به کمک محاسبه FT های مختلف برای بازه های زمانی متوالی و سپس در کنار هم قرار دادن آنها فراهم میکند. که همان بیان مدنظر ماست. اما این روش هم با سوالی بنیادی مواجه است:

"اندازه پنجره چقدر باشد؟"

- اگر پنجره خیلی بزرگ باشد: تبدیل به همان FT ساده میشود که اطلاعات فرکانسی خوب اما اطلاعات زمانی مناسبی در اختیار قرار نمیدهد.

یک عمل ریاضی است که با دریافت یک تابع و یا رشته آنرا به تابع یا رشته جدید تبدیل می کند.

چرا به تبدیل ها نیاز داریم؟

- تبدیل یک تابع میتواند اطلاعاتی اضافی از یک تابع یا اطلاعات مخفی شده در تابع را آشکار کند
- تبدیل یک معادله ممکن است آسان تر از اصل آن حل شود
- تبدیل یافته یک تابع/رشته ممکن است نیاز به فضای کمتری برای ذخیره داشته باشد و بنابراین امکان فشرده سازی دیتا را فراهم میکند
- یک عمل ممکن است به تبدیل یافته یک تابع ساده تر اعمال شود تا به خود آن (مثلا کانولوشن)

تبدیل فوریه:

یک تابع پیوسته و یا دارای گسستگی پریودیک می تواند توسط مجموعی از توابع سینوسی بیان شود.

$$F(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

$$f(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} F(\omega) e^{i\omega t} d\omega$$

از مهم ترین کاربردهای تبدیل فوریه، یافتن فرکانسی خاص در سیگنال هاست.

اما نکته ای در این عملیات وجود دارد این است که تبدیل فوریه تمامی اجزای موجود در دل سیگنال را شناسایی می کند، اما هیچ اطلاعاتی در خصوص مکان این اجزا ارائه نمی کند.

- اگر پنجره خیلی کوچک باشد: اطلاعات زمانی خوب اما اطلاعات فرکانسی مناسبی ارائه نمیکند.

اصل عدم قطعیت هایزنبرگ:

اصل عدم قطعیت هایزنبرگ به میگوید که حاصل ضرب رزولوشن فرکانسی و زمانی همواره از مقدار ثابتی کمتر خواهد بود.

در واقع ما نمیتوانیم ما به راه حلی برسیم که حداکثر بازدهی در هر دو را داشته باشد.

$$\Delta t \cdot \Delta f \geq \frac{1}{4\pi}$$

و اما اکنون زمان خودنمایی تبدیل موجک فرا رسیده است.

تبدیل موجک:

ایده اصلی تبدیل موجک مانند فوریه زمان کوتاه است که تغییر زیر در آن گرفته شده است:

با استفاده از یک پنجره با طول متغیر می توان بر مشکل از پیش تعیین کردن رزولوشن غلبه کرد.
پنجره های با طول متغیر برای فرکانس های مختلف استفاده شوند:

- برای فرکانس های بالا از پنجره های باریک تر استفاده میکنیم
- برای فرکانس های پایین از پنجره های عریض استفاده میکنیم.

لازم به ذکر است که اصل عدم قطعیت هایزنبرگ همچنان صادق است.

تابعی که برای پنجره ای کردن سیگنال استفاده میشود موجک نامیده میشود.

$$cwt(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} x(t) \psi\left(\frac{t-\tau}{s}\right) dt$$

τ = translation, s = scale, $\Psi(t)$ = mother wavelet, $\left(\frac{t-\tau}{s}\right)$ = scale factor

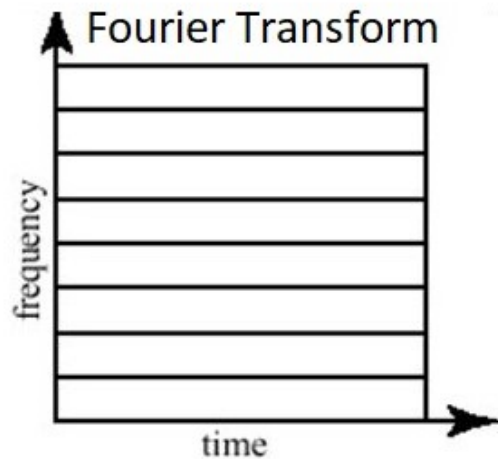
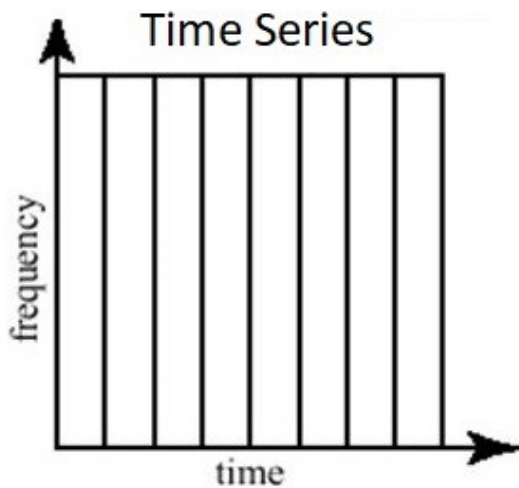
تبدیل فوریه در برابر تبدیل موجک:

همانطور که پیش تر گفته شد موجک نارسایی فوریه در مکان وقوع را پوشش میدهد.

و برای دستیابی به این هدف، ایده تقسیم سیگنال به

بخش های کوچک تر با طول متغیر را مطرح میکند.

تصاویر زیر نحوه نگاه دو تبدیل فوریه زمان کوتاه و موجک به سیگنال را با یکدیگر مقایسه میکنند:



میدهد با این تفاوت که در آن طول پنجره دیگر ثابت نیست و با توجه به اندازه فرکانس تغییر میکند.

هرم ها:

هرم یا نمایش هرمی نوعی نمایش سیگنال چند مقیاسی است که توسط جوامع بینایی ماشین، پردازش تصویر و جوامع پردازش سیگنال ساخته شده است، که در آن یک سیگنال یا یک تصویر تحت مجموعه ای از عملیات ها مانند smoothing و... قرار میگیرد و در نهایت مجموعه از تصاویر مشتق از تصویر اصلی با مقیاس های گوناگون را داریم.



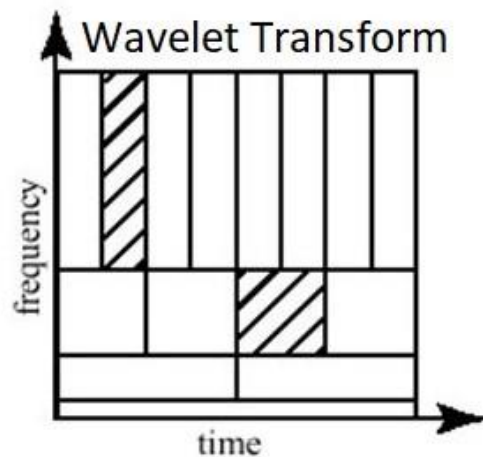
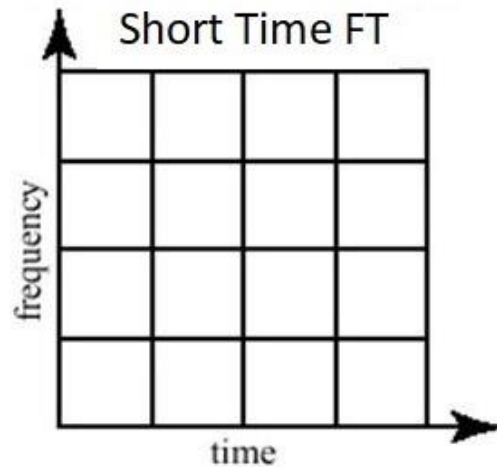
High resolution → Low resolution

دو نوع اصلی از هرم وجود دارد:

- lowpass
- Bandpass

هرم گوسین:

در پایین ترین سطح این هرم تصویر اصلی قرار دارد و هر چه جلو تر میرویم در هر مرحله یک فیلتر گوسین به تصویر اعمال میشود و بدین ترتیب فرکانس های بالای تصویر دور ریخته میشود و سایز تصویر نیز نصف میشود.



همانطور که در تصویر بالا مشاهده میکنید، در بخش اول (بالا سمت چپ) فقط رزولوشن زمانی را پوشش میدهد و همانطور که مشاهده میشود هیچ اطلاعاتی درباره ی فرکانس ندارد.

تصویر بعدی بیانگر تبدیل فوریه است که در آن فقط رزولوشن فرکانسی در نظر گرفته شده است اما هیچ اطلاعاتی درباره ی زمان در اختیار ما نمیگذارد.

تصویر بعدی فوریه زمان کوتاه را نشان میدهد که در آن هم رزولوشن زمانی و هم رزولوشن مکانی در نظر گرفته شده اند. اما عرض پنجره ها همگی یکسان اند.

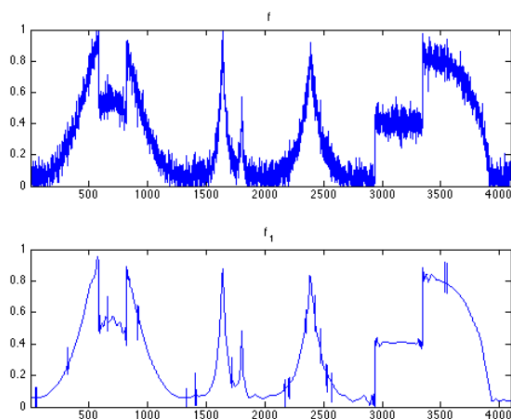
و در نهایت تبدیل موجک را مشاهده میکنید که مانند فوریه زمان کوتاه هر دو رزولوشن زمانی و مکانی را پوشش

هرم لاپلاسین :

پیش از این توضیح دادیم که هرم گوسی با استفاده از هموارکننده ی گوسین و سپس subsampling آن تصویر هموار شده ایجاد می شود. درمورد هرم لاپلاسی میتوان اینگونه گفت که این هرم در سطح اول از طریق تفاضل بین تصویر اولیه و تصویر گوسی سطح دوم که به اندازه ی سایز تصویر اول scale-up شده است بدست می آید، و سطح های بعدی نیز به همین ترتیب ایجاد می شود. به عبارت دیگر میتوان گفت که تصاویر در هرم لاپلاسیان همان دور ریز تصاویر در هر سطح از هرم گوسین است.



حذف نویز:



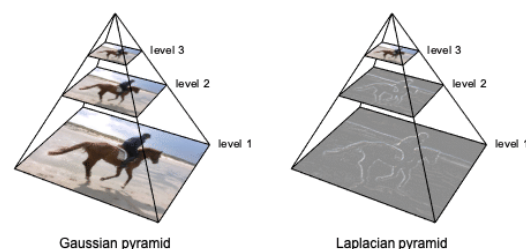
اگر $y(t)$ یک سیگنال ثبت شده تجربی همراه با توصیف اساسی باشد، $g(t)$ یک مدل برای فرایند جمع شدن نویز که $g(t)$ را به $y(t)$ تبدیل می کند، توسط معادله زیر توصیف می شود:

$$y_i = g(t_i) + \sigma \epsilon_i, \quad i = 1, \dots, n$$

در رویکرد معمول حذف نویز، نویز را به عنوان سیگنال با فرکانس بالا به سیگنال اصلی اضافه می کنند. برای ردیابی این فرکانس بالا می توان از تبدیل فوریه استفاده کرد و در نهایت با فیلتر کافی آن را از بین برد. این استراتژی حذف نویز از نظر مفهومی روشن و کارآمد است زیرا فقط به محاسبه DFT سیگنال داده شده بستگی دارد. با این حال، مواردی وجود دارد که باید مورد توجه قرار گیرند.

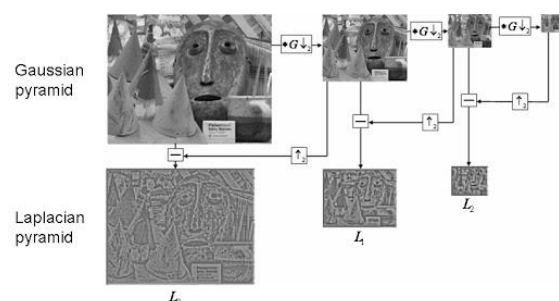
برجسته ترین این موارد زمانی بروز می کند که سیگنال اصلی دارای اطلاعات مهم مربوط به همان فرکانس نویز باشد. وقتی نمایش سیگنال از دامنه فرکانس بدست می آید، فیلتر کردن این فرکانس باعث از دست رفتن محسوس اطلاعات سیگنال هدف می شود.

در مواردی که شرح داده شده است، روش موجهها به دلیل این واقعیت که سیگنال با استفاده از نمایش زمان فرکانس "dual" مورد مطالعه قرار می گیرد، مناسب تر است، که اجازه می دهد فرکانسهای نویز را از فرکانسهای سیگنال با ارزش جدا کند. تحت این روش، نویز به عنوان یک سیگنال ثابت با فرکانس بالا در کل محدوده زمانی نشان داده می شود و بنابراین شناسایی آن آسان تر از استفاده از تجزیه و تحلیل فوریه است.



طبق توضیحات داده شده میتوان نتیجه گرفت با داشتن هرم لاپلاسین و همچنین بالا ترین سطح هرم گوسین میتوان به سایر سطوح هرم گوسین و همچنین تصویر اصلی رسید.

به تصویر زیر دقت کنید:



باشروع از بالاترین سطح در هر مرحله کافی است تصویر در هرم گوسین را 2 برابر کرده و با تصویر در هرم لاپلاسین جمع بزنیم.

پس از مشخص شدن نمایش نویز ، روند حذف شروع می شود. ثابت شده است که یک استراتژی مناسب برای حذف نویز، برابر با صفر قرار دادن ضرایب مرتبط با فرکانس نویز است. این عبارت یک چشم انداز سراسری را برای حذف نویز را نشان می دهد ، روشهای مختلف denoising در نحوه ردیابی و ضبط ضرایب از نمایش متفاوت است. جزئیات مفهومی چندین مورد از این روشها در بخشهای بعدی ارائه شده است.

روش اصلی و ساده برای حذف نویز از یک سیگنال آلوده به آن شامل اصلاح ضرایب موجک ها به روشی هوشمندانه است به طوری که ضرایب "کوچک" مرتبط با نویز اساساً نادیده گرفته می شوند. از این رو می توان از ضرایب به روز شده برای بازسازی عملکرد اصلی و عاری از تأثیر نویز استفاده کرد. در استراتژی ضمنی است که فقط چند ضریب موج "بزرگ" d_{jk} با سیگنال اصلی مرتبط هستند و شناسایی و حذف ضرایب دیگر به شما امکان بازسازی کامل سیگنال اساسی g را می دهد. روش های مختلفی از این ایده استفاده می کنند و آن را به روش های مختلف پیاده سازی می کنند.

در روش جریمه خطی ، هر ضریب موجک تحت تأثیر انقباض خطی خاص مربوط به سطح تفکیک ضریب است. یک عبارت ریاضی برای این نوع رویکرد با استفاده از جمع شدگی خطی در معادله نشان داده شده است:

$$\tilde{d}_{jk} = \frac{\hat{d}_{jk}}{1 + \lambda 2^{2js}}$$

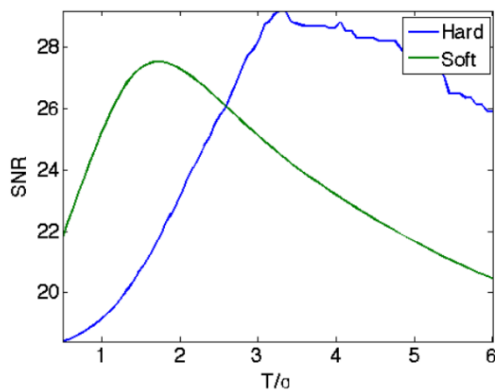
در معادله زیر، پارامتر s شاخص صاف شناخته شده سیگنال اساسی g است ، در حالی که پارامتر λ یک عامل هموار کننده است که تعیین آن برای این نوع تجزیه و تحلیل حیاتی است.

باید گفت که آستانه خطی فقط برای سیگنال فضایی همگن با سطوح مهم نظم کافی است. در صورت عدم رعایت شرایط خانگی و نظم ، آستانه موجک غیرخطی یا جمع شدگی معمولاً مناسب تر است.

donoho1995b و donoho1995 یک استراتژی غیرخطی برای آستانه گذاری پیشنهاد کردند. طبق رویکرد آنها ، آستانه گذاری را می توان با اجرای یک قانون آستانه سخت یا نرم انجام داد. عبارات ریاضی آنها به ترتیب در معادلات زیر نشان داده شده است:

$$\delta_{\lambda}^H(\hat{d}_{jk}) = \begin{cases} 0 & \text{if } |\hat{d}_{jk}| \leq \lambda \\ \hat{d}_{jk} & \text{if } |\hat{d}_{jk}| > \lambda \end{cases}$$

$$\delta_{\lambda}^S(\hat{d}_{jk}) = \begin{cases} 0 & \text{if } |\hat{d}_{jk}| \leq \lambda \\ \hat{d}_{jk} - \lambda & \text{if } \hat{d}_{jk} > \lambda \\ \hat{d}_{jk} + \lambda & \text{if } \hat{d}_{jk} < -\lambda \end{cases}$$



در هر دو روش ، نقش پارامتر λ به عنوان مقدار آستانه به عنوان برآوردگر منجر به تخریب ، کاهش یا افزایش مقدار ضریب موجک ، حیاتی است.

چندین نویسنده درباره خصوصیات و محدودیت های این دو استراتژی بحث کرده اند. آستانه سخت ، به دلیل ناپیوسته بودن ناشی از آن ، حتی به تغییرات کوچک در داده ها نیز می تواند ناپایدار و حساس باشد. از طرف دیگر ، آستانه نرم هنگامی که ضرایب واقعی زیاد هستند می تواند تعصب غیرضروری ایجاد کند. اگرچه روشهای پیچیده تری برای ارائه اشکالات استراتژیهای غیرخطی توصیف شده معرفی شده است ، بحث در این گزارش به روشهای سخت و نرم محدود می شود.

Term-by-Term Thresholding

یک مشکل آشکار در استفاده از روشهای آستانه موجک ، نحوه انتخاب مقدار مناسب برای آستانه ، λ است. در واقع روش های مختلفی برای تعیین مقدار پارامتر مورد نظر

$$\lambda^U = \hat{\sigma} \sqrt{2 \log n}$$

این آستانه به راحتی به خاطر سپرده می شود و اجرای آن در نرم افزار ساده تر است و از مشکل بهینه سازی ضمنی در روش minimax جلوگیری می شود. همچنین ، آستانه سزاسازی با احتمال زیاد اطمینان می دهد که هر نمونه در تبدیل موجک که عملکرد اصلی آن دقیقاً صفر است ، صفر برآورد می شود ، اگرچه سرعت همگرایی (بسته به اندازه نمونه) کند است.

۳-شرح نتایج و نتیجه گیری

در ادامه به بررسی عملیات های یاد گرفته شده بر روی تصویر زیر میپردازیم:



ابتدا تصویر را gray scale میکنیم.

6.1.1

هرم لاپلاسی 9 مرحله ای را از روی آن تشکیل میدهم:



وجود دارد. به معنای کلی ، این استراتژی ها را می توان در دو گروه آستانه های سراسری و آستانه های وابسته به سطح طبقه بندی کرد. آستانه سراسری به معنای انتخاب یک مقدار λ است که برای همه ضرایب موجک اعمال می شود. آستانه های وابسته به سطح بیانگر این است که برای هر سطح وضوح یک λ مقدار آستانه (احتمالاً) متفاوت اعمال می شود. همه گزینه ها نیاز به برآورد سطح نویز σ دارند. انحراف معیار مقادیر داده به طور واضح برآورد خوبی نیست ، مگر اینکه عملکرد اصلی پاسخ g به طور منطقی مسطح باشد. donoho1995 تخمین σ را در حوزه موجک با استفاده از عبارت موجود در معادله زیردر نظر گرفت:

$$\hat{\sigma} = \frac{\text{median}(|\hat{d}_{J-1,k}|)}{0.6745}, \quad k = 0, 1, \dots, 2^{J-1} - 1$$

The minimax threshold

donoho1995 با به حداقل رساندن ریسک در برآورد یک تابع ، مقدار آستانه بهینه λ^M را بدست آورد. آستانه مینیماکس پیش بینی شده به داده های موجود بستگی دارد و سطح نویز آلوده کننده سیگنال را نیز در نظر می گیرد:

$$\lambda^M = \hat{\sigma} \lambda_n^*$$

$$\lambda_n^* = \inf_{\lambda} \sup_d \left\{ \frac{R_{\lambda}(d)}{n^{-1} + R_{\text{oracle}}(d)} \right\}$$

$$R_{\lambda}(d) = E(\delta_{\lambda} \hat{d})^2$$

The universal threshold

donoho1995 این آستانه را به عنوان جایگزینی برای آستانه های مینیماکس ، که برای همه ضرایب موجک اعمال می شود ، پیشنهاد داد. آستانه جهانی در معادله زیر تعریف شده است:



مرحله ی آخر این هرم زمانی ست که تنها یک پیکسل باقی مانده است.

سپس طبق توضیحات داده شده در مرحله شرح تکنیکال به بازسازی تصویر از روی هرم لاپلاسین میپردازیم:



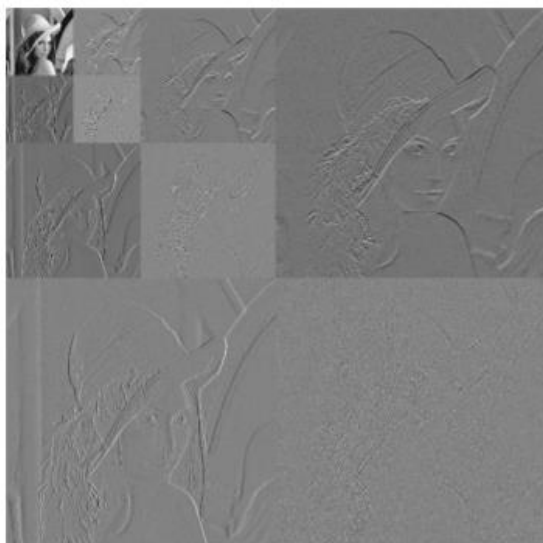
مجددا مقادیر MSE و PSNR را محاسبه میکنیم:

MSE	PSNR
0	inf

همانطور که از مرحله قبل انتظار داشتیم این تصویر نیز عینا مطابق تصویر اصلی است.

6.1.3

هرم موجک 3 مرحله ای را از روی آن تشکیل میدهم:



حال مقادیر mse و psnr را برای تصویر بازسازی شده در مقایسه با تصویر اصلی محاسبه میکنیم:

MSE	PSNR
0	inf

نتایج خبر از تطابق کامل تصویر بازسازی شده با تصویر اصلی میدهند. به عبارت دیگر هرم لاپلاسین شامل تمام داده های مورد نیاز ماست و در هیچ داده ای از بین نمیروند.

6.1.2

هرم 3 مرحله ای را از روی آن تشکیل میدهم:



مجددا مانند بخش قبل تصویر را بازسازی میکنیم:

سپس به محاسبه دو مقدار MSE و PSNR میپردازیم:

MSE	PSNR
1.425	44.324

در این حالت با مقدار کمی از دست رفتن دیتا مواجهیم اما مقدار آن بسیار ناچیز و قابل چشم پوشی است.

6.2

در این بخش ابتدا اندکی نویز به تصویر اضافه خواهیم کرد سپس به رفع آن با تکنیک های اشاره شده میپردازیم:



تصویر نویزی



تصویر denoise شده با تکنیک soft threshold

سپس تصویر اصلی را از روی این تصویر بازسازی میکنیم:



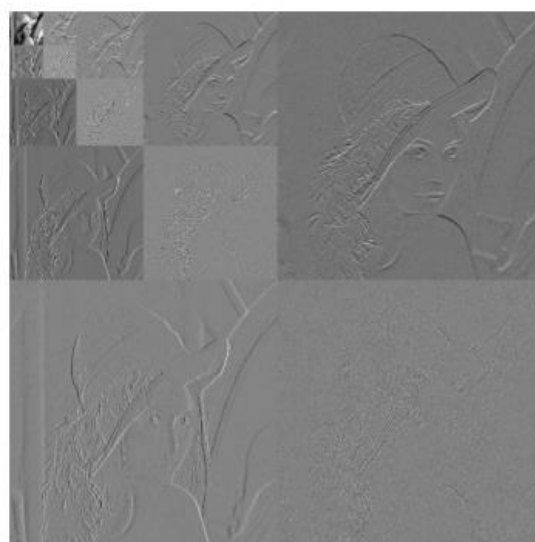
سپس مقدار PSNR و MSE را برای تصویر بازسازی شده محاسبه میکنیم:

MSE	PSNR
0	inf

باز هم مطابقت کامل و باز هم حفظ کلیه داده ها!

6.1.4

هرم موجک چندی سازی شده را برای تصویرمان تشکیل میدهم:



هرم ویولت:

```

3  [B,C] = size(B);
4  out = zeros(B,1,'double');
5  while(1) break; %?
6  [Bn,Cn] = size(Cin,'double');
7  out = (out/2) + (Cn/2) + ones(Bn,Cn,level+1)/ones(B,C,level+1);
8  out(Cn/2+1,Cn/2+1) = 0;
9  out(Cn/2,Cn/2+1) = 0;
10 out(Cn/2,Cn/2) = 0;
11 out(Cn/2,Cn/2) = 0;
12 %
13 [I,J] = size(out);
14 %
15 % out(I,J) = (out(I,J)+1)/2; %? level=1+1;
16 [I,J] = size(out);
17
18 AS = cell(I,J,1);
19 AS = cell(I,J,1);
20
21 W = W/(W+1); %?
22 W = W/(W+1); %?
23
24 out(I,J) = (out(I,J)+1)/2; %? level=1+1;
25 out(I,J) = (out(I,J)+1)/2; %? level=1+1;
26 out(I,J) = (out(I,J)+1)/2; %? level=1+1;
27 out(I,J) = (out(I,J)+1)/2; %? level=1+1;
28
29 out(I,J) = (out(I,J)+1)/2; %? level=1+1;
30 out(I,J) = (out(I,J)+1)/2; %? level=1+1;
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964

```



ریولت چندی ساز شدہ:

```

1  //compute matrix elements
2  fact = 1.0/240.0;
3  compute matrix(a,"1/240.0");
4  //factors
5  fact1 = fact;
6  for (k=0;k<10;k++) fact1 *= fact1;
7  fa = fact1*matrix(a,"1/240.0");
8  ra = fact1*matrix(a,"1/240.0");
9  ca = fact1*matrix(a,"1/240.0");
10 cb = fact1*matrix(a,"1/240.0");
11 output(1/240.0,1/240.0) = summat(fa,baet1);
12 output(1/240.0,1/240.0) = w;
13 output(1/240.0,1/240.0) = w;
14 // = output;
15 [D1,D2] = eig(w);
16 // = eig(D1,D2) [eigen(1,1);2*(1/240.0) 1/240.0];
17 // [e1,e2] = eig(w);
18 // W = eig(D1,D2) [1,1/240.0];
19 // W = eig(D2,D1) [1,1/240.0];
20 // D = eig(D1,D1) [1,1/240.0];
21 // D = eig(D2,D2) [1,1/2*(1/240.0) 1,1/2*(1/240.0)];
22 // = W*D1*W';
23 // = W*D2*W';
24 //show(w);
25 show(w,baet1);
26 end

```

```

3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
```

تصویر denoise شده با تکنیک hard threshold

4- پیوست

هرم لاپلاسي:

```

8  %meanCorr = sumCorr / size(sumCorr, 'rows')
9  %isolving = normCorr('quasiqr', 0.01)
10 % [W, Z] = size(isolvingZ)
11 output = zeros(W, 1, 'double')
12 % [H, V, D] = svd(isolvingZ, 'best')
13 % beta = sqrt(log2(W/3))
14 % sigma = modfun(beta, 0.7, 0.745), *2
15 % [H, V, D] = svd(isolvingZ, 'best', sigma)
16 % D = treshZ(D, beta, sigma)
17 % V = treshZ(V, beta, sigma)
18 output((W/2-1):(W/2) + W/4, 1) = 1
19 % [H, V, D] = svd(isolvingZ, 'best', sigma)
20 output((W/2+1):(W/2+1)+W/4, 1) = 0
21 % = last output, bestw1
22 % = last output, bestw2
23 figure, imshow(output(p, :))
24 end

```

حذف نویز:

```

30 Function: outputs = threshold(img,thres,sigma)
31 [N,C] = size(img);
32 outputs = zeros(N,C);
33 local_sigma = width(img)/2;
34 I = (img.*sigma)/local_sigma;
35 for i=1:N
36     for j=1:C
37         res = I(i,j);
38         if (abs(res)<T)
39             res = 0;
40         else
41             res = sign(res)*abs(res)^T;
42         end
43         output(i,j)=res;
44     end
45 end
46
47 end
48
49
50

```

```

33
34
35 function output = iset(varargin,level)
36 if (level==0)
37     output = varargin{1};
38 else
39     output = varargin{1};
40     for i = 2:numel(varargin)
41         dec_out{i} = output{1:(2^level-1),1:(2^level-1)};
42         out = dec_out{i}(1:r/2,1:r/2);
43         out = dec_out{i}(r/2+1:r,1:r/2);
44         out = dec_out{i}(1:r/2,r/2+1:r);
45         out = dec_out{i}(r/2+1:r,r/2+1:r);
46         output{1:(2^level-1),1:(2^level-1)} =
47             [out;out;out;out];
48     end
49 end

```

[illegible]

```

54:         output = reshape(output, [1, 1, 1, 1])
55:         output = zeros(2*W, 2*W, classNum);
56:         for k = 1:K
57:             for y = 1:W
58:                 for x = 1:W
59:                     output(y, 1+W-x, 1+W-x) = img(x, y);
60:                     output(y, 1+W-x, 1+W-x) = img(x, y);
61:                     output(y, 1+W-x, 1+W-x) = img(x, y);
62:                 end
63:             end
64:         end
65:     end
66: end
67:
68: function output = normImage
69:     % normImage: normalize image
70:     % Inputs:
71:     %   img - matrix image (img(:,:,1))
72:     %   min - min value (img(:,:,1))
73:     %   max - max value (img(:,:,1))
74:     % Outputs:
75:     %   output - (255*(max-min)/img(:,:,1)) * img(:,:,1)
76:
77:     img = img(:,:,1);
78:     min = min(img);
79:     max = max(img);
80:     output = (255*(max-min)/img(:,:,1)) * img(:,:,1);
81: end

```

منابع:

<https://fa.wikipedia.org/wiki/%D9%85%D9%88%D8%AC%DA%A9>

[https://en.wikipedia.org/wiki/Pyramid_\(image_processing\)](https://en.wikipedia.org/wiki/Pyramid_(image_processing))

<https://cnx.org/contents/-yxzQLO-@2/Signal-Denoising-using-Wavelet-based-Methods#uid24>

http://www.numerical-tours.com/matlab/denoisingwav_1_wavelet_1/d