

Frequency domain

فهمیم جعفری

اطلاعات گزارش	چکیده
تاریخ:	فیلتر کردن در حوزه فرکانس شامل اصلاح تبدیل فوریه تصویر ، و سپس محاسبه تبدیل معکوس برای به دست آوردن نتیجه ی پردازش شده است. بنابراین با توجه به تصویر دیجیتالی $f(x,y)$ به اندازه $M*N$ ، معادله اصلی فیلتر کردن که به آن علاقه مندیم به صورت زیر است: $G(x,y)=Finverse[H(u,v)F(u,v)]$ ، توابع F, H, G ، ارایه هایی به اندازه $M*N$ هستند که مثل تصویر ورودی است. ضرب $F(u,v)H(u,v)$ با استفاده از ضرب ارایه شکل گرفت. تابع فیلتر، تبدیل ورودی را اصلاح میکند تا خروجی پردازش شده ی $g(x,y)$ بدست آید
واژگان کلیدی:	
Fourier Transform	
Phase	
Magnitude	
Zero padding	
Dft leakage	
Mmse	
Freauency domain	
Smooth filter	
Edge detection	
Shift	
Imagin	
Real	

1-مقدمه

2-شرح تکنیکال

Phase: $\phi = \tan^{-1} \frac{I(\omega)}{R(\omega)}$

• Fourier Transform

تبدیل فوریه به طور کلی مختط میباشد و میتوان به صورت قطبی نوشت:

$$F(u, v) = |F(u, v)|e^{j\phi(u, v)}$$

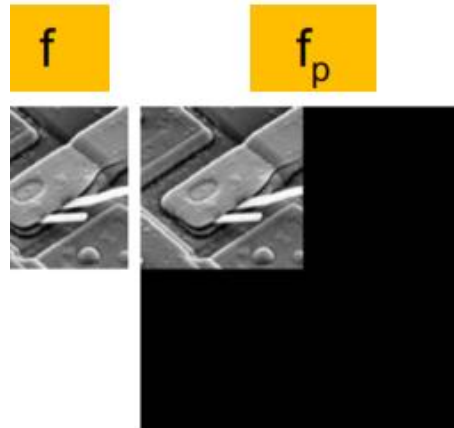
که طیف ان به صورت زیر میباشد

$$F(u, v) = [R^2(u, v) + I^2(u, v)]^{\frac{1}{2}}$$

و فاز ان به صورت

که ما فقط با قسمت طیف ان کار داریم چرا که دستکاری فاز موجب بهم ریختگی تصویر میشود فیلتر و تصویر در حوزه فرکانس باید اندازه یکسانی داشته باشند تا انها در هم ضرب کرد پس نیاز است اندازه فیلتر را که معمولا کوچکتر از تصویر میباشد بزرگ کرد برای اینکار اضافه کردن صفر به فیلتر در حوزه مکان گزینه مناسبی میباشد چرا که تغییری در فوریه ان نمیکند اما نکته ای دیگر وجود دارد و ان این است که اضافه

کردن صفر موجب بهبود دقت در تبدیل میشود
چرا که نمونه هایی که در تبدیل برداشته میشود
بیشتر میشود پس میتوان از این استفاده کرد تا
دقت تبدیل فوریه تصویر را نیز با افزودن صفر به
آخر آن افزود مانند شکل زیر



برای اینکه تبدیل فوریه را به صورت خروجی
نمایش دهیم باید آن را در حوزه فرکانس شیف
داد چرا که مبدا در نمایش تصویر گوشه سمت
چپ بالا میباشد برای اینکار تصویر را قبل از
محاسبه dft در $(-1)^{x+y/2}$

ضرب میکنیم که x, y ابعاد تصویر میباشد چرا
که شیف در حوزه فرکانس باعث ضرب در-
 $1^{(x+y)}$ میشود

همینطور چون فرکانس در مبدا دارای مقدار
زیادی میباشد و نمایش حوزه فرکانس در
نمایشگر فقط یک نقطه دیده میشود چون در
نمایشگر مقادیر بین رنج 0 تا 255 میباشد پس
ما از فرکانس لگاریتم میگیریم تا تغییرات مقدار
فرکانس در نمایشگر نسبت به فرکانس قابل
محسوس باشد

پس مراحل اعمال فیلتر در حوزه فرکانس به
صورت زیر میباشد

- بدست آوردن P و Q با توجه به اندازه $M \times N$ تصویر f عموماً $P=2M$ و $Q=2N$ انتخاب می شود
- ساختن تصویر f_p با عمل افزودن صفر به تصویر f
- ضرب کردن f_p در $(-1)^{x+y}$ برای انتقال مبدا به مرکز
- بدست آوردن DFT حاصل
- اعمال فیلتر حقیقی با مبدا مرکز $(Q/2, P/2)$ با ضرب آرایه ای
- بدست آوردن IDFT حاصل
- انتخاب بخش حقیقی نتیجه (چون با تغییر ضرایب ممکن است بخش موهومی هم به تصویر اضافه شده باشد که قابل صرف نظر کردن است)
- ضرب کردن نتیجه در $(-1)^{x+y}$ برای بدست آوردن g_p
- برداشتن بخشی به اندازه $M \times N$ از گوشه ی بالای سمت چپ g_p

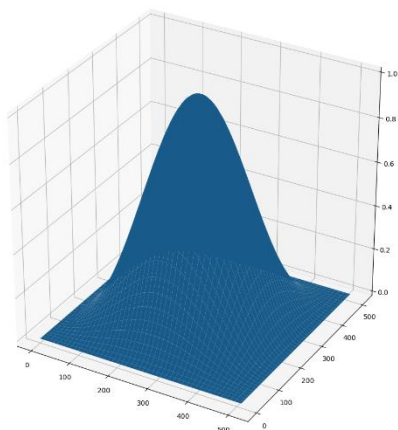
فیلتر هایی که بکار خواهیم برد فیلتر های زیر
میباشد

$$a) \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad b) \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad c) \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

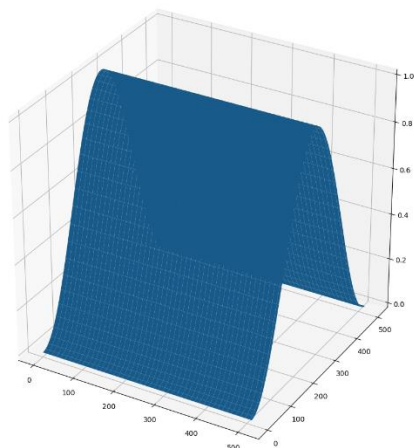
بعضی از فیلتر ها را میشود جدا کرد به طوری که
دو فیلتر بدست بیاید و اگر هر کدام را جداگانه
بر روی تصویر به طور متوالی اعمال کنیم نتیجه
یکسانی با فیلتر اصلی دارد و هنگامی که فیلتر
ها جدا میشود محاسبات در حوزه مکان کمتر
است چرا که نیاز به کانوالو یک بعدی است از بین
فیلتر های بالا فقط فیلتر a را میشود جدا کرد
چرا که طبق علم جبر خطی در صورتی یک
ماتریس را به دو وکتور شکست که ردیف های آن
ضربیی از هم باشند که در ماتریس a این ویژگی
دیده میشود و میتوان آن را به دو وکتور زیر که
یکی در راستای x و دیگری در راستای y فیلتر
میکند

1	2	1
---	---	---

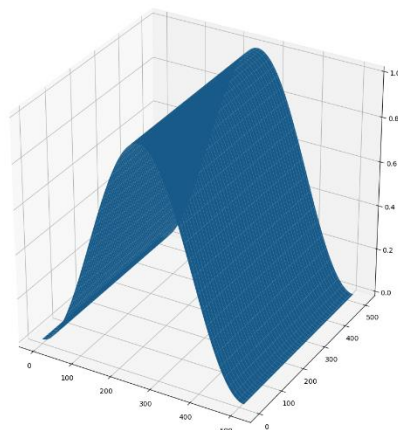
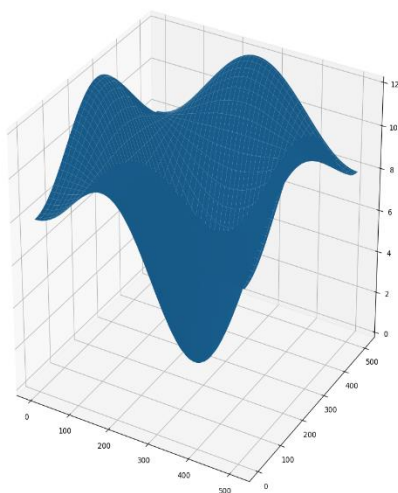
1
2



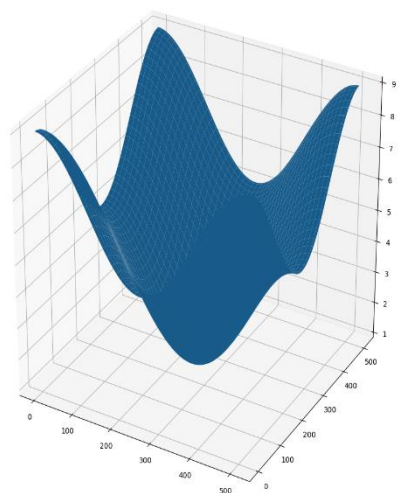
که هنگامی که به حوزه فرکانس برده میشوند
بترتیب به شکل زیر در می آیند



و دو فیلتر دیگر در حوزه فرکانس به ترتیب b,c
به شکل زیر میباشد



و فیلتر اصلی که در حوزه مکان ضرب ماتریس
این دو میباشد در حوزه فرکانس ضرب ارایه اگر
انجام بگیرد به شکل زیر در می آید که دقیقاً
شبیه فوریه خود فیلتر اصلی است.



• مقایسه فیلتر کردن در حوزه مکان و

فرکانس (قسمت 4.2.1)

برای اینکه عمل فیلتر کردن در حوزه مکان را در حوزه فرکانس انجام دهیم باید تبدیل فوری فیلتر را ضرب ارایه ای در تبدیل فوری تصویر کرد برای اینکار مراحل زیر را انجام میدهیم

1. اضافه کردن صفر به فیلتر به اندازه ای که اندازه فیلتر برابر تصویر بشود زیرا برای اینکه ضرب ارایه در حوزه فرکانس انجام دهیم باید اندازه دو عملوند برابر باشد

2. ضرب فیلتر و تصویر در $(-1)^{(x/2+y/2)}$

که x, y ابعاد تصویر میباشد این ضرب برای این میباشد که چون ارایه ها اندیس منفی ندارند و ما مجبور به شیفت مبدا فرکانس میشویم این ضرب موجب شیفت مبدا در حوزه فرکانس میشود

3. محاسبه تبدیل فوری فیلتر

4. محاسبه تبدیل فوری تصویر

5. ضرب بزرگی طیف تبدیل فوری تصویر در تبدیل فوری فیلتر چون کانوالو در حوزه مکان موجب ضرب در حوزه فرکانس میشود

6. محاسبه معکوس تبدیل فوری با قسمت بزرگی طیف نتیجه مرحله قبل و قسمت فاز تبدیل فوری تصویر بدون تغییر

7. ضرب نتیجه مرحله قبل در $(-1)^{(x+y)}$ تا دوباره به مبدا اصلی برگردیم

8. از آنجا که نتیجه معکوس تبدیل فوری ممکن است دارای قسمت مختلط باشد ما بزرگی نتیجه را نمایش میدهیم

البته جای این نکته باقی است که چون در پیاده سازی از توابع آماده استفاده شده است و عمل شیفت دادن توسط خود تابع در حوزه فرکانس انجام میگردد پس ترتیب مراحل بالا کمی متفاوت است

قسمت دوم:

فیلتر کردن در حوزه فرکانس تصویر را پریودیک در نظر میگیرد پس در نتیجه اگر صفر به آن

اضافه نشود که در سوال اینطوری در نظر گرفته شده است حاشیه ها تحت تاثیر هم قرار میگیرند و در نتیجه حاشیه ها در فیلتر کردن در حوزه مکان و فرکانس مقادیر یکسانی تولید نمیکنند به عبارتی $Y(m,n)=Z(m,n)$ به ازای m,n حاشیه با هم برابر نمیشوند و این رنج نابرابری به سبب فیلتر بستگی دارد پس به ازای m,n های غیر حاشیه خروجی فیلتر کردن در حوزه مکان و فرکانس با هم برابر میباشد

• DFT certain Zeroout coefficients

یکی از روش های فیلتر کردن در حوزه فرکانس حذف کردن (صفر کردن) رنج خاصی از بزرگی (magnitude) فرکانس هاست. یکی از کاربردهای این فیلتر ها حذف کردن فرکانس های بالا میباشد که منجر به اسموت کردن تصویر میشود که نتیجه ای شبیه فیلترهای جعبه ای میدهد. اگر تصویر را به حوزه فرکانس برده و فرکانس هایی که میخواهیم آنها را نگه داریم در 1 ضرب کنیم و فرکانس هایی که می خواهیم در صفر ضرب کنیم کار این فیلتر ها را انجام داده ایم.

تصویر در حوزه فرکانس اگر آن را شیفت ندهیم فرکانس های بالا در وسط تصویر قرار میگیرند. چرا که قسمت های منفی یا قرینه فرکانس های اصلی به علت شروع اندیس در تصویر از سمت چپ بالا در گوشه ها قرار میگیرند. مانند شکل پایین که با فرمول آن نمایش داده شده است

$$F(k,l) = 0 \text{ for } TN < \{k,l\} < (1 - T)N, T = 1/4, 1/8$$

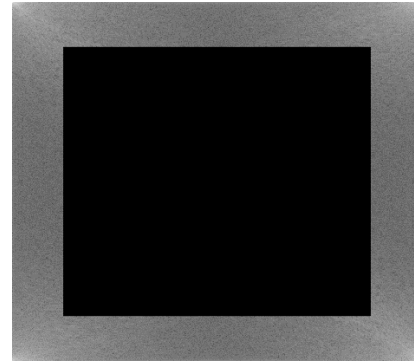


اجازه دهید نگاهی به تصویر زیر که مربوط به فیلتر a در راستای y میباشد بیندازیم



حال بیایید این دو را مقایسه کنیم همانطور که مشاهده میشود در تصویر بالایی (x) خطهای افقی اسموث یا ضعیف نشدند ولی خطهای عمودی همه که بر روی روسری Barbara میباشد ضعیف شدند ولی برعکس در تصویر پایینی خطهای عمودی اسموث نشدند و خط های افقی که در قسمت کتاب ها مشاهده میکنید ضعیف شدند

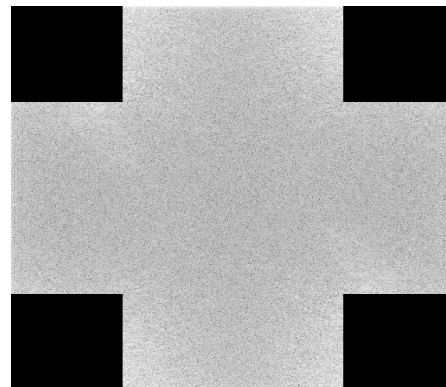
تصویر زیر فیلتر a در هر دو جهت میباشد که لبه ها در هر دو جهت ضعیف شدند



همینطور میتوان توسط این فیلتر فرکانس های پایین را حذف کرد که گوشه ها قرار دارند و فقط فرکانس های بالا که حاوی لبه ها میباشد را نگه داشت. مانند شکل پایین که ترکیب 4 حالت b میباشد در بخش تشریح نتایج هر کدام جداگانه بررسی شده است

b. Let $F(k,l) = 0$ for the following regions:

- i. $0 \leq \{k \text{ and } l\} \leq TN$;
- ii. $0 \leq k \leq TN, \text{ and } (1-T)N \leq l \leq N-1$;
- iii. $(1-T)N \leq k \leq N-1 \text{ and } 0 \leq \{l\} \leq TN$;
- iv. $(1-T)N \leq k \text{ and } l \leq N-1; T = 1/4, 1/8$



3-شرح نتایج

• Fourier transform

خوب از آنجا که فیلتر a جدا پذیر بود ما این فیلتر را در سه حالت امتحان کردیم 1 فیلتر a در راستای x 2 فیلتر a در راستای y 3 فیلتر a در هر دو جهت که هر کدام لبه ها را به ترتیب در راستای عمودی، راستای افقی، هر دو جهت ضعیف تر میکند

تصویر زیر مربوط به فیلتر در راستای X میباشد



که مشاهده میشود فقط لبه ها باقی مانده است
برای اینکه مقایسه بین حوزه فرکانس و حوزه
مکان داشته باشیم یک mmse بین این دو
گرفته شده است که برابر 1.6×10^{-7} میباشد
که نزدیک صفر است . خروجی ها برابرند

تصویر زیر مربوط به اعمال فیلتر c میباشد
همانطور که که از شکل این فیلتر در حوزه
فرکانس مشخص است این فیلتر فرکانس های
بالا را با قطعیت کمتری تقویت میکند و فرکانس
های پایین را قطعیت کمتری حذف میکنند در
نتیجه خروجی تصویری است که لبه ها در آن
تقویت شده .

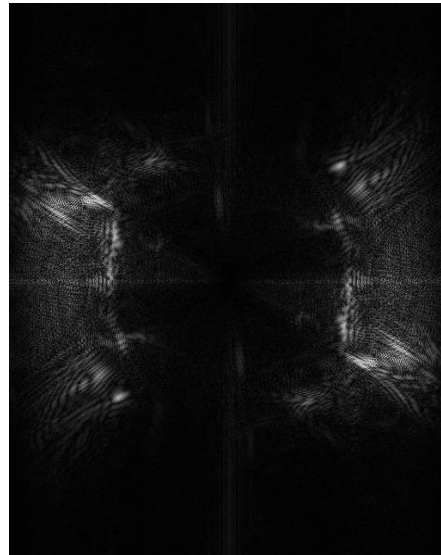


تصویر زیر مربوط به اعمال تبدیل فوریه بر روی
تصویر Lena میباشد

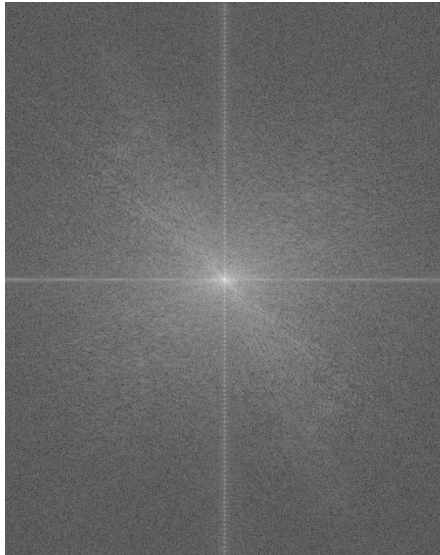


برای اینکه مطمئن شویم نتیجه جدا شده ی
فیلتر با نتیجه فیلتر اصلی یکی است یک mmse
بین خروجی فیلتر اصلی و متوالی اعمال کردن
فیلتر های جدا شده اندازه گرفته شد که برابر با
0.11 میباشد که مشاهده میشود عدد کوچکی
است این عدد میتواند حاصل خطای محاسبات
در dft منظور از خطا پدیده مثل dft leakage
و... میباشد

تصویر زیر مربوط به حوزه فرکانس تصویر
Barbara پس از اعمال فیلتر میباشد

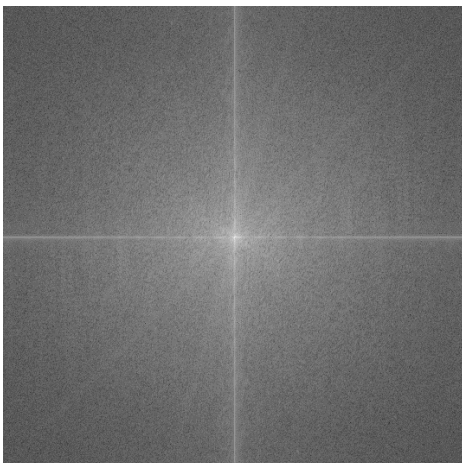


مشاهده میشود که فرکانس های بالا تقویت و
فرکانس های پایین حذف و ضعیف شدند که
فرکانس های بالا مربوط به لبه ها میباشد
و تصویر حاصل آن در حوزه مکان به شکل زیر
میباشد

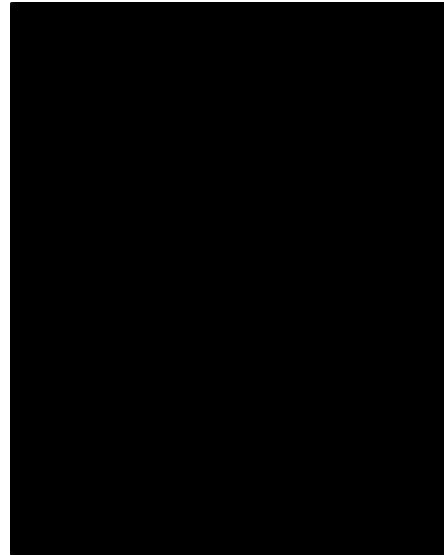


اگر بخواهیم تحلیلی بر روی آن داشته باشیم میتوان گفت تصویر اصلی یک تصویر smooth بوده است و از فرکانس های بالا کم برخوردار بوده است زیرا تبدیل فوریه آن بیشترین مقدار فرکانس ها فرکانس های نزدیک صفر میباشد و در راستای عمودی و افقی میباشد که نشان دهنده ی لبه های عمود بر آنها در حوزه مکان می باشد

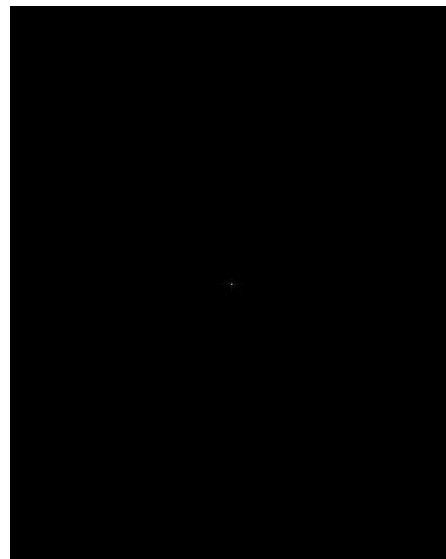
تصویر زیر مربوط به تبدیل فوریه عکس F16 میباشد که لگاریتم گرفته شده و شیفت خورده است



در تصویر بالا بیشترین مقادیر در فرکانس های نزدیک به صفر و در راستای افقی و عمودی و یک خط با شیب مثبت میباشد که این خط ناشی از لبه های با شیب منفی در تصویر F16 میباشد



در تصویر بالا چیزی دیده نمیشود زیرا بیشترین مقدار تبدیل فوریه در مبدا میباشد که مبدا عکس در گوشه سمت راست بالا است و چون پس زمینه به رنگ سفید است به خوبی تمایز داده نمیشود برای رفع این مشکل فوریه را شیفت میدهیم که به شکل زیر در می آید



در تصویر بالا فقط یک نقطه دیده میشود برای چون نسبت بزرگی فرکانس در مبدا تبدیل فوریه بسیار بزرگتر است از بقیه فرکانس ها و چون رنج فرکانس به بین 0 تا 256 تغییر کرده است بقیه فرکانس ها مقدار صفر (سیاه) را گرفته اند برای اینکه این مشکل را حل کنیم و تغییرات فرکانسی را حس کنیم از تبدیل فوریه لگاریتم میگیریم که به شکل زیر در می آید

قبلی این تصویر به دلیل وجود فرکانس های بالا
تصویری شلوغتر نسبت به بقیه میباشد

• مقایسه فیلتر در حوزه مکان و فرکانس

مراحل انجام عملیات قسمت اول سوال در شرح
تکنیکال گفته شده است تصویر زیر فیلتر کردن در
حوزه مکان میباشد که بین 0 تا 256 scale شده
است فیلتر لاپلاسی است



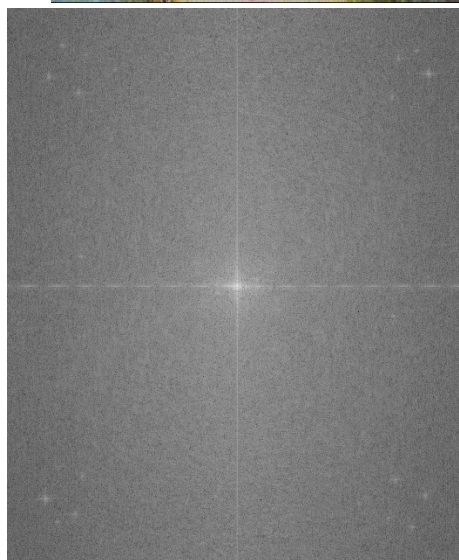
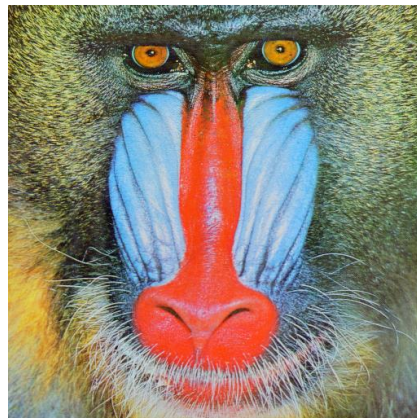
تصویر زیر فیلتر در حوزه فرکانس ان میباشد



همانطور که مشخص میشود دو تصویر یکی هستند
برای اطمینان مساوی بودن نتایج mmse ان دو و
تفریق انها اندازه گرفته شده است که mmse برابر
39 است به علت این که در حوزه فرکانس تصویر
پریودیک در نظر گرفته میشود و ما در قسمت
محاسبه تبدیل فوریه به تصویر صفر اضافه کرده ایم

که مربوط به لبه ی کوه در تصویر است که عمود
بر خط با شیب مثبت در حوزه فرکانس است

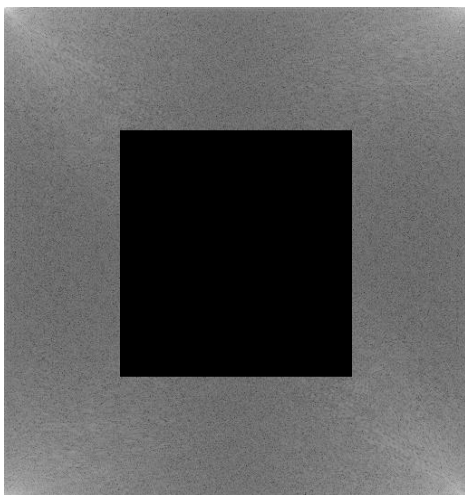
تصویر زیر مربوط به تبدیل فوریه و عکس
Baboon میباشد که لگاریتم گرفته شده و
شیفت خورده است



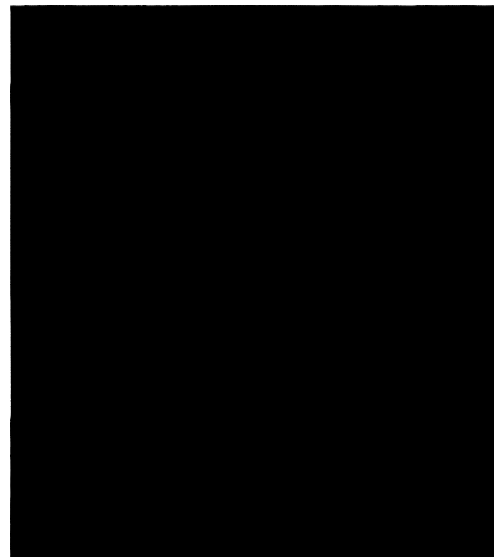
در این تصویر نقطه های در فرکانس های بالا
دیده میشود که به این علت میباشد که در تصویر
اصلی (baboon) لبه هایی یکسان با فاصله
یکسان از هم وجود دارد که با موج های سینوسی
به خوبی انطباق داده شده و correlation بالایی
دارد مانند لبه هایی تقریباً پریودیک که در گونه
های بابون وجود دارد که یک پریودیک بودن
دیگر هم در حوزه فرکانس در قسمت نقطه چین
بودن خط افقی وجود دارد که نشان دهنده ی
نظم خاص در لبه های موجود در تصویر بابون
است اگر بخواهیم مقایسه داشته باشیم با تصاویر



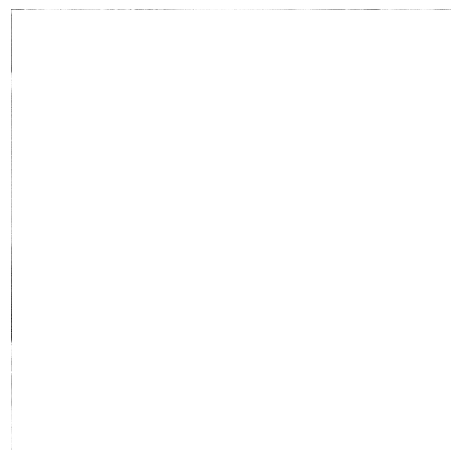
برای اینکه تصویر را اسموث کنیم و از شدت لبه ها بکاهیم باید در حوزه فرکانس ،فرکانس های بالا را حذف کنیم که تصویر زیر حوزه فرکانس و مکان با فیلتر اعمال شده با $T=1/4$ را نشان میدهد



در نتیجه فقط حاشیه های تصویر با هم برابر نیستند و تفریق تصویر به شکل زیر است



چون تصویر پس زمینه سفید است سفیدی حاشیه تصویر قابل تمایز نمیباشد با پس زمینه نیست برای همین تصویر را نگتیو میکنیم که در شکل زیر به خوبی حاشیه دیده میشود

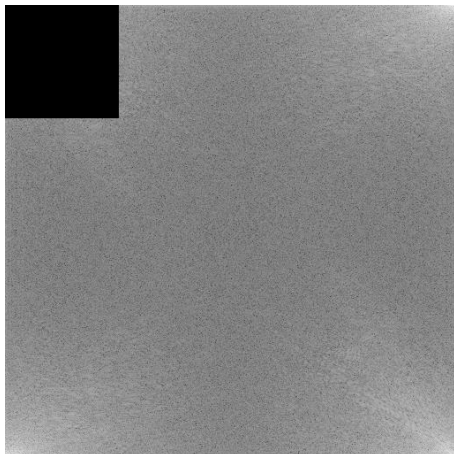


• DFT certain Zeroout coefficients

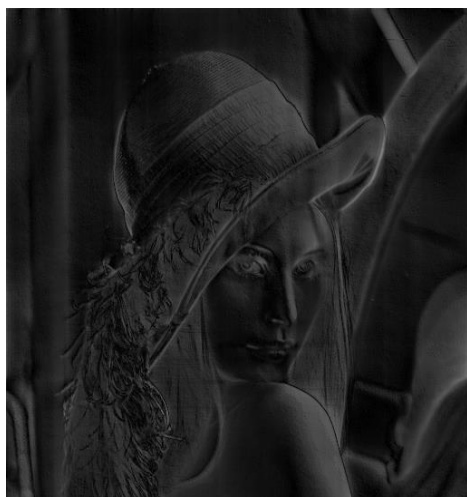
تصویر زیر تصویر اصلی میباشد که فیلتر ها قرار است در حوزه فرکانس بر روی ان ها انجام شود

تصویر زیر مربوط فوریه تصویر لنا میباشد که فرکانس های زیر با $t=1/4$ از آنها حذف شده است

$$0 \leq \{k \text{ and } l\} \leq TN;$$



تصویر زیر مربوط به حوزه مکان تصویر فوریه بالا میباشد

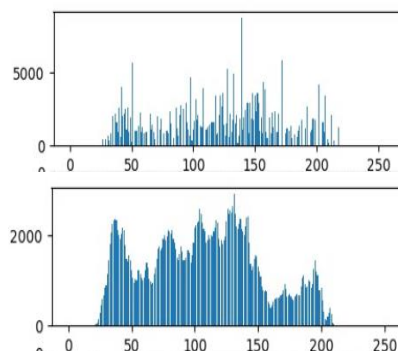


همانطور که مشاهده میشود تصویر تاریک شده است به دلیل آن است که فرکانس dc حذف شده است علاوه بر آن لبه های عمود بر رنج فرکانسی که حذف شده است نیز حذف شده است برای $T=1/8$ به شکل زیر میباشد

در تصویر حوزه فرکانس یک لگاریتم گرفته شده است تا تغییرات در حوزه فرکانس محسوس شود همانطور که مینید فرکانس های بالا را که در وسط تصویر است حذف کردیم که فرمول آن در شرح تکنیکال میباشد. همانطور که دیده میشود تصویر اسموث شده است و لبه ها ضعیف شدند تصویر زیر مربوط به $T=1/8$ میباشد که مشاهده میشود تصویر اسموث تر شده است



چون داریم در حوزه فرکانس حذف میکنیم و در حوزه فرکانس پریودیک میباشد پس لبه ها به صورت پریودیک حذف میشوند و این حذف پریودیک در تصویر به صورت خط های موازی هم و با فاصله یکسان دیده میشود دو تصویر زیر هیستوگرام بدون فیلتر و با فیلتر را نشان میدهد که با دقت در آنها درمیابیم که با زدن فیلتر هیستوگرام شیفت خورده و گسستگی های آن پیوسته تر میشوند



برای اینکه لبه ها از تصویر استخراج کنیم باید فرکانس ها پایین را حذف کنیم که در حوزه فرکانس مربوط به گوشه ها میشود

مشاهده میشود که اینبار لبه های عمود برا حالت قبل حذف شده است ولی تفاوت ان با **bi** این است که لبه ی ها افقی حذف نشده است چون در فرکانس هم نیز ما خط عمودی حذف نکردیم باز هم تصویر اسمو ت تر شده است و دلیل اینکه تصویر روشن تر است نسبت به حالت قبل چون فرکانس **dc** حذف نشده است

اگر با $t=1/8$ امتحان کنیم حالت **bii** را به شکل زیر در می آید



چون فرکانس های بالای کمتری حذف شده است در نتیجه لبه های کمتری نسبت به حالت قبل ضعیف شده است

تصویر زیر مربوط به بخش **biii** با $t=1/4$ میباشد

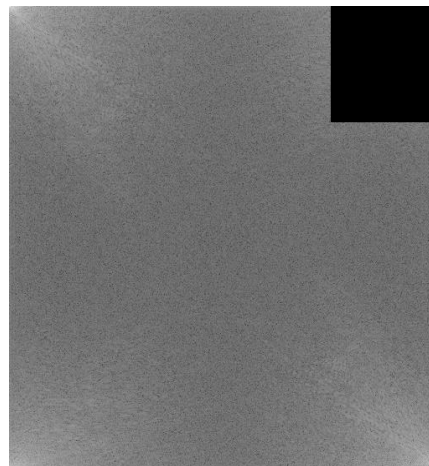


در تصویر بالا شبیه حالت **bii** میباشد با این تفاوت که در این تصویر به جای حذف لبه های عمودی، لبه های افقی حذف شده است که حوزه



اگر دقت کنید مشاهده میشود که چون فرکانس های بالای کمتری حذف شده است لبه ها تقویت شده اند

در تصویر زیر فرکانس های دیگری را حذف کردیم که در شرح تکنیکال قسمت **bii** و $t=1/4$ میباشد





این تصویر مشابه حالت b_i میباشد با این تفاوت که لبه های افقی و عمودی و فرکانس dc از آن حذف نشده است به خاطر همین تصویر روشن تر دیده میشود

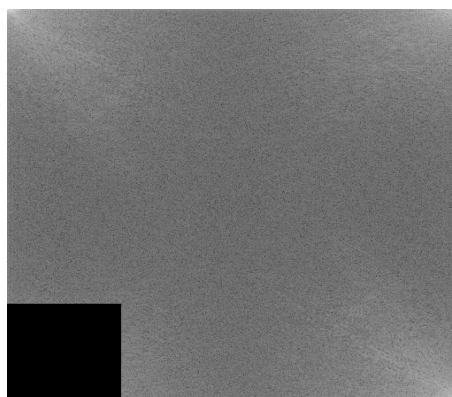
تصویر زیر b_{iv} با $T=1/8$ میباشد



مشاهده میشود که لبه های کمتری حذف شده است و تصویر تیزتر میباشد نسبت به حالت قبل.

تصویر زیر مربوط به گزارش $mmse$ این فیلتر ها میباشد

فرکانس آن هم در زیر نشان داده است که در آن جا ما خط های عمودی ($x=0$) را حذف کردیم و خط افقی ($y=0$) را دست نزدیم



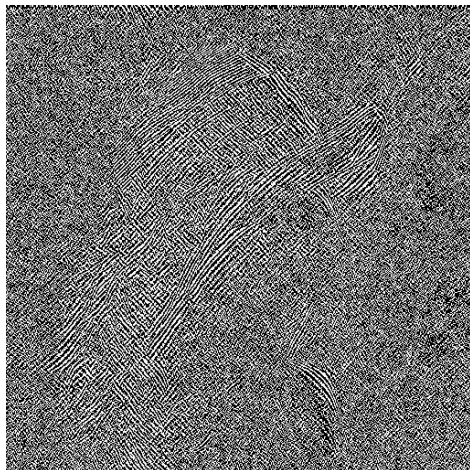
تصویر زیر مربوط b_{iii} با $t=1/8$ میباشد که به شکل زیر در آمده است



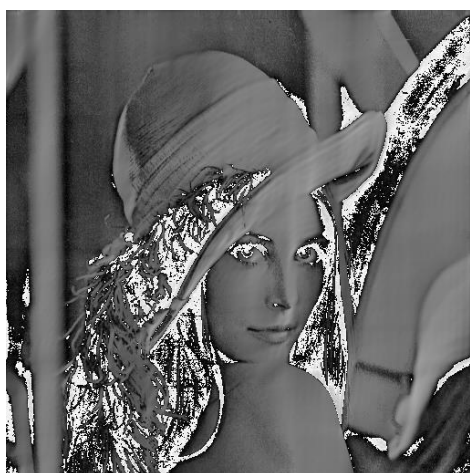
مشاهده میشود که باز هم لبه های کمتری ضعیف شده اند

تصویر زیر مربوط به b_{iv} با $t=1/4$ میباشد

نتیجه را گرفت که بیشتر توان تصویر در فرکانس های پایین قرار دارد
تصاویر زیر حاصل تفریق تصویر اصلی و فیلتر های اعمال شده میباشد که نشان دهنده ی لبه های حذف شده میباشد



فیلتر a



فیلتر bi



lv	1130.3714637455328	1130.3714637455328
lv	259.29876540055074	251.22200615383514
bii	171.9141728930046	165.4137374683444
bii	271.1276384960592	264.04287534706515
bi	7769.014617885755	7754.455895985297
a	433197113591428	21.577982932801
	0.25	0.125

همانطور که از جدول مشخص است در فیلتر هایی که فرکانس هایی پایین از آنها حذف شده است مقدار mmse زیاد است پس میتوان این

```

    FP = np.zeros(F.shape,dtype=float)
    for i in range(0,F.shape[0]):
        for j in range(0,F.shape[1]):
            FP[i,j] =
cmath.phase(F[i,j])
    return FA,FP

#convert amplitude and phase to imagine
and real parts
def toRect(A,P):
    F = np.zeros(A.shape,dtype=complex)
    for i in range(0,F.shape[0]):
        for j in range(0,F.shape[1]):
            a =
cmath.rect(A[i,j],P[i,j].real)
            F[i,j] = a
    return F

img =
cv2.imread('../images/hw4.1.1/resfiltera
x.png')
img =
cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
N = img.shape[0]
M = img.shape[1]
p = N
q = M

fp = img
Fuv = np.fft.fft2(fp)
FuvA,FuvP = toPolar(Fuv)
FuvA = np.fft.fftshift(FuvA)

#define filters
filtera = [[1,2,1],[2,4,2],[1,2,1]]
filterax = [[1,2,1]]
filteray = [[1],[2],[1]]
filterb = [[-1,-1,-1],[-1,8,-1],[-1,-1,-
1]]
filterc = [[0,-1,0],[-1,5,-1],[0,-1,0]]
filtsize = 1
filtsizey = 3
filt = filteray
filtname = 'filteray'
filtgain = 1/4

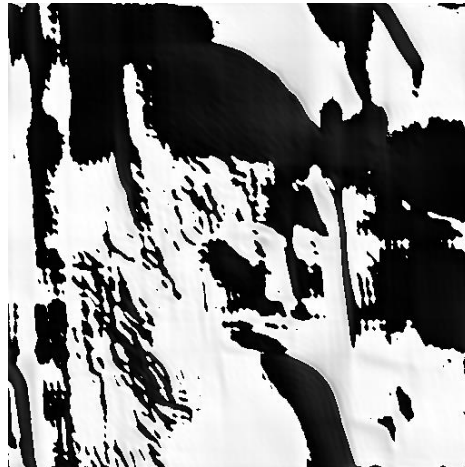
paddvaluex = p -filtsize
paddvaluey = p - filtsizey
filt = filtgain * np.asarray(filt)
filtp =
np.pad(filt,((0,paddvaluey),(0,paddvalue
x)))
H = np.fft.fft2(filtp)
HA,HP = toPolar(H)
HA = np.fft.fftshift(HA)

RESA = FuvA * HA          #do filter
RESA += 1                 #this is added
by 1 because log(0) is infinity
showRES = np.log(RESA)
min = np.min(RESA)
max = np.max(RESA)
showRES = normalizeimg(RESA,min,max)
cv2.imwrite('../images/hw4.1.1/RESA{}and
filterax.png'.format(filtname),showRES)

RESA = np.fft.ifftshift(RESA)
RES = toRect(RESA,FuvP)
resp = np.fft.ifft2(RES)

```

فیلتر bii



فیلتر biii



فیلتر biv

4-کد برنامه

کد قسمت 4.1.1

```

import cv2
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import cmath
from src.test import laplacian_filter

#normalize image
def normalizeimg(img,min,max):
    img = (img-min)/(max-min)*255
    return img

#compute magnitude and phase of input
iamgine array
def toPolar(F):
    FA = abs(F)

```

```

N = img.shape[0]
M = img.shape[1]
p = 2*N
q = 2*M

uu, vv = np.meshgrid(range(p), range(q))
fig = plt.figure(figsize=(7,10))

fp = np.pad(img, (0,M))
Fuv = np.fft.fft2(fp)
FuvA, FuvP = toPolar(Fuv)
min = np.min(FuvA)
max = np.max(FuvA)
cv2.imwrite('../images/hw4.1.2/{ }WosWol.png'.format(picName), normalizeimg(FuvA, min, max))
logres = np.log10(FuvA)
minlog = np.min(logres)
maxlog = np.max(logres)
cv2.imwrite('../images/hw4.1.2/{ }WosWl.png'.format(picName), normalizeimg(logres, minlog, maxlog))

FuvA = np.fft.fftshift(FuvA)
cv2.imwrite('../images/hw4.1.2/{ }WsWol.png'.format(picName), normalizeimg(FuvA, min, max))
logres = np.log10(FuvA)
cv2.imwrite('../images/hw4.1.2/{ }WsWl.png'.format(picName), normalizeimg(logres, minlog, maxlog))

```

```

res =
abs(resp[0:img.shape[0],0:img.shape[1]])
#for normalize image uncommetn
# min = np.min(res)
# max = np.max(res)
# res = normalizeimg(res,min,max)
cv2.imwrite('../images/hw4.1.1/res{ }andfilterax.png'.format(filename), res)

uu, vv = np.meshgrid(range(p), range(q))
fig = plt.figure(figsize=(10,10))
ax =
fig.add_subplot(1,1,1,projection='3d')
ax.plot_surface(uu,vv,HA[uu,vv],alpha=1)
plt.show()
fig.savefig('../images/hw4.1.1/{ }.png'.format(filename))

#compute mmse for ax and ay
# imga =
cv2.imread('../images/hw4.1.1/resfiltera.png')
# imga =
cv2.cvtColor(imga, cv2.COLOR_RGB2GRAY)
# mmse =
np.square(np.subtract(imga, res)).mean()
# print(mmse)

```

کد قسمت 4.1.2

کد قسمت 4.2.1

```

import cv2
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import cmath
from src.test import laplcian_filter

#normalize image
def normalizeimg(img,min,max):
    img = (img-min)/(max-min)*255
    return img

#compute magnitude and phase of input
iamgine array
def toPolar(F):
    FA = abs(F)
    FP = np.zeros(F.shape,dtype=float)
    for i in range(0,F.shape[0]):
        for j in range(0,F.shape[1]):
            FP[i,j] =
cmath.phase(F[i,j])
    return FA,FP

#convert ampitude and phase to imagine
and real parts
def toRect(A,P):
    F = np.zeros(A.shape,dtype=complex)
    for i in range(0,F.shape[0]):
        for j in range(0,F.shape[1]):
            a =
cmath.rect(A[i,j],P[i,j].real)
            F[i,j] = a
    return F

```

```

import cv2
import numpy as np
import cmath
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

#compute magnitude and phase of input
iamgine array
def toPolar(F):
    FA = abs(F)
    FP = np.zeros(F.shape,dtype=float)
    for i in range(0,F.shape[0]):
        for j in range(0,F.shape[1]):
            FP[i,j] =
cmath.phase(F[i,j])
    return FA,FP

#convert ampitude and phase to imagine
and real parts
def toRect(A,P):
    F = np.zeros(A.shape,dtype=complex)
    for i in range(0,F.shape[0]):
        for j in range(0,F.shape[1]):
            a =
cmath.rect(A[i,j],P[i,j].real)
            F[i,j] = a
    return F

#normalize image
def normalizeimg(img,min,max):
    img = (img-min)/(max-min)*255
    return img

picName = 'Capture3'
img =
cv2.imread('../images/{ }.png'.format(picName))
img =
cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

```

4.2.2 کد قسمت

```
import numpy as np
import cv2
import cmath
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

#compute magnitude and phase of input
iamgine array
def toPolar(F):
    FA = abs(F)
    FP = np.zeros(F.shape, dtype=float)
    for i in range(0, F.shape[0]):
        for j in range(0, F.shape[1]):
            FP[i, j] =
cmath.phase(F[i, j])
    return FA, FP

#convert amplitude and phase to imagine
and real parts
def toRect(A, P):
    F = np.zeros(A.shape, dtype=complex)
    for i in range(0, F.shape[0]):
        for j in range(0, F.shape[1]):
            a =
cmath.rect(A[i, j], P[i, j].real)
            F[i, j] = a
    return F

#normalize image
def normalizeimg(img, min, max):
    img = (img-min)/(max-min)*255
    return img

#compute and return notch filter without
shift
def notchFilter(type, T, M, N):
    H = np.ones((N, M))
    if type=='a':
        H[int(T*N):int((1-
T)*N), int(T*N):int((1-T)*N)] = 0
    elif type=='bi':
        H[0:int(T*N), 0:int(T*N)+1] = 0
    elif type=='bii':
        H[0:int(T*N), int((1-T)*N):N] = 0
    elif type=='biii':
        H[int((1-T)*N):N, 0:int(T*N)+1] =
0
    elif type=='biv':
        H[int((1-T)*N):N, int((1-T)*N):N]
= 0
    elif type=='bv':
        H = np.zeros((N, M))
        H[0:N, 0] = 1
        H[0, 0:M] = 1
        H[N-1, 0:M] = 1
        H[0:N, M-1] = 1

    return H

M=512
N =512
picName = 'Lena'
img =
cv2.imread('../images/{0}.bmp'.format(pic
Name))
```

```
img = cv2.imread('../images/Lena.bmp')
img =
cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
N = img.shape[0]
M = img.shape[1]
p = 1*N
q = 1*M

fp = np.pad(img, (0, 0))
Fuv = np.fft.fft2(fp)
FuvA, FuvP = toPolar(Fuv)
FuvA = np.fft.fftshift(FuvA)

filtera = [[-1, -1, -1], [-1, 8, -1], [-1, -1, -
1]]
filtsizeX = 3
filtsizeY = 3
filt = filtera
filtname = 'filtera'
filtgain = 1/16

paddvalueX = p -filtsizeX
paddvalueY = p - filtsizeY
filt = filtgain * np.asarray(filt)
filtP =
np.pad(filt, ((0, paddvalueY), (0, paddvalue
x)))
H = np.fft.fft2(filtP)
HA, HP = toPolar(H)
HA = np.fft.fftshift(HA)

RESA = FuvA * HA
# RESA += 1
# showRES = np.log(RESA)
# min = np.min(RESA)
# max = np.max(RESA)
# showRES = normalizeimg(RESA, min, max)
#
cv2.imwrite('../images/hw4.2.1/RESA{0}.png'
.format(filtname), showRES)

RESA = np.fft.ifftshift(RESA)
RES = toRect(RESA, FuvP)
resp = np.fft.ifft2(RES)
res =
abs(resp[0:img.shape[0], 0:img.shape[1]])
min = np.min(res)
max = np.max(res)
res = normalizeimg(res, min, max)
cv2.imwrite('../images/hw4.2.1/res{0}.png'
.format(filtname), res)

lapmask = laplcian_filter(img, filt)
min = np.min(lapmask)
max = np.max(lapmask)
lapmask = normalizeimg(lapmask, min, max)
cv2.imwrite('../images/hw4.2.1/lap.png',
lapmask)
mmse =
np.square(np.subtract(res, lapmask)).mean()
#compute mmse between filter in
freq domain and spatial domain
print(mmse)

sub = np.subtract(res, lapmask)
max = np.max(sub)
min = np.min(sub)
sub = normalizeimg(sub, min, max)
cv2.imwrite('../images/hw4.2.1/sub.png',
sub)
```



```

mmse =
np.square(np.subtract(img,res)).mean()
d.append(mmse)
data.append(d)
fig.savefig('../images/hw4.2.2/spatial.png')

#display mmse table
col_labels = (" ",
"a","bi","bii","biii","biv","bv")
fig, ax = plt.subplots(dpi=300,
figsize=(5,1))
ax.axis('off')
ax.table(cellText=data,
colLabels=col_labels, loc='center')
fig.savefig('../images/hw4.2.2/resTable.png')

mmse =
np.square(np.subtract(img,res)).mean()

#for compute deferent between origin
image and filtered image uncomment
# im1 = cv2.imread('../images/Lena.bmp')
# im1 =
cv2.cvtColor(im1,cv2.COLOR_RGB2GRAY)
# im2 =
cv2.imread('../images/hw4.2.2/resa-
0.125.png')
# im2 =
cv2.cvtColor(im2,cv2.COLOR_RGB2GRAY)
# sub = np.subtract(im1,im2)
#
cv2.imwrite('../images/hw4.2.2/Leananda.png',sub)

```

```

img =
cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
Fuv = np.fft.fft2(img)
FuvA,FuvP = toPolar(Fuv)

T1 = [1/4,1/8]
types =
['a','bi','bii','biii','biv','bv']
data = []
uu, vv = np.meshgrid(range(N),range(M))
fig = plt.figure(figsize=(10,10))
i=1
for T in T1:
    d = [T]
    for type in types:
        H = notchFilter(type,T,M,N)
        showH = abs(np.fft.ifft2(H))
        ax =
fig.add_subplot(6,2,i,projection='3d')
i += 1

ax.plot_surface(uu,vv,showH[uu,vv],alpha
=1)
ax.view_init(30, 0)
RES = FuvA*H
#for showing frequency domain
numcomment below
# RES += 1
# showRES = np.log10(RES)
# min = np.min(showRES)
# max = np.max(showRES)
# showRES =
normalizeimg(showRES,min,max)
#
cv2.imwrite('../images/hw4.2.2/magnitude
{}-{}.png'.format(type,T),showRES)
RES = toRect(RES,FuvP)
res = np.fft.ifft2(RES)
res = abs(res)

cv2.imwrite('../images/hw4.2.2/res{}-
{}.png'.format(type,T),res)

#compute mmse between orginal
img and filtered imag

```

مراجع

کتاب گنزالس رافائل

اسلایدهای دکتر امیرحسین طاهری نیا

با کمک های حل تمرین درس آقای بلویان و خانم رستمی و آقای کرمی