

پروژه سیگنال فاز 1

استاد خلع

امیر رضاضامنی آتنی

400101504

سوال (1)

بخش (1)

در این بخش با استفاده از فیلتر گوسی هایپس با پارامتر سیگما برابر 300 و با رابطه زیر تصویر را در حوزه فرکانس فیلتر کردیم و در همان حوزه ضربی از فرکانس های فیلتر شده را به fft2 عکس اضافه کردیم به علت شیفیت دادن fft2 در ابتدا کار فرکانس های پایین به وسط ماتریس منتقل میشوند و هر چه به اطراف حرکت میکنیم فرکانس های بالاتر را خواهیم داشت با ضرب تابع z در ماتریس fft2 ضربی که در وسط قرار دارند در ضریب کوچکی (تقریباً صفر) ضرب میشوند و فرکانس های بالاتر در ضریب نزدیک به 1 ضرب میشوند. سپس ماتریس حاصل را با ضریب kc با fft2 تصویر جمع میکنیم تا فرکانس های بالاتر تقویت شوند.

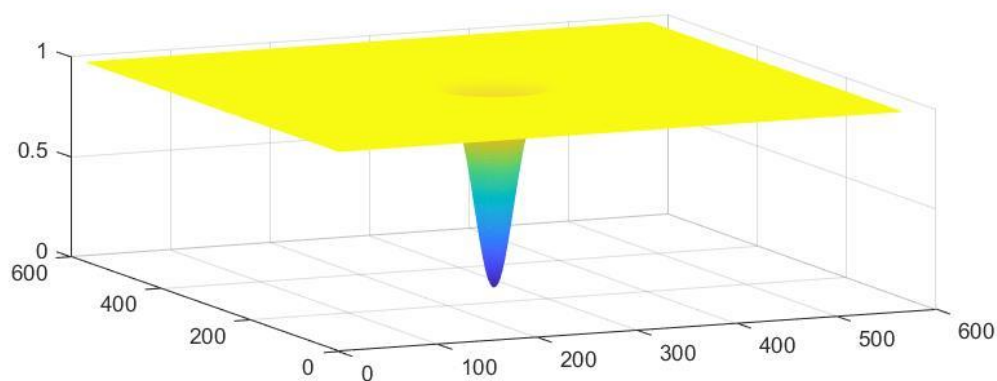
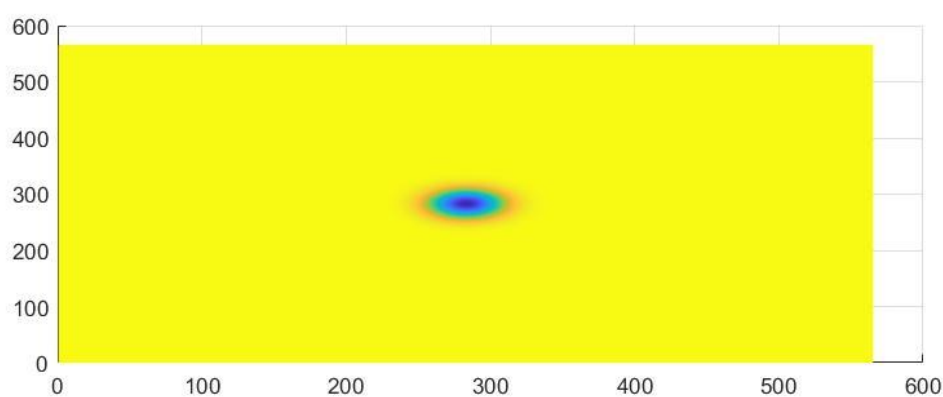
به علت اینکه با جمع کردن فرکانس های بالاتر و در نهایت اسکیل کردن سهم فرکانس های پایین که نقش اصلی را در روشنایی کلی عکس دارند با تقویت فرکانس های بالاتر روشنایی عکس کم میشود (با افزایش ضریب kc روشنایی کم میشود).

مشاهده میکنیم که تصویر شارپ تر میشود و لبه ها معلوم تر و تیز تر شده اند و تغییرات محسوس تر شدند

نمودار و رابطه فیلتر گوسی استفاده شده z:

highpass filter :

$$f(x, y) = \left(\max \left(\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma}} \right) - \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma}} \right) \frac{1}{\max \left(\frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma}} \right)}$$



نتیجه فیلتر به ازای مقادیر مختلف kc : عکس های دقیق تر و بزرگتر با نام گذاری:

Q1_p1_1.jpg, Q1_p1_1.5.jpg, Q1_p1_0.7.jpg

$kc=0.7$:



$kc=1$:



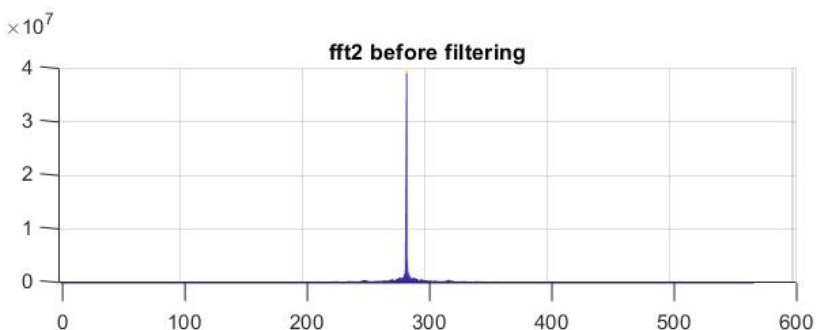
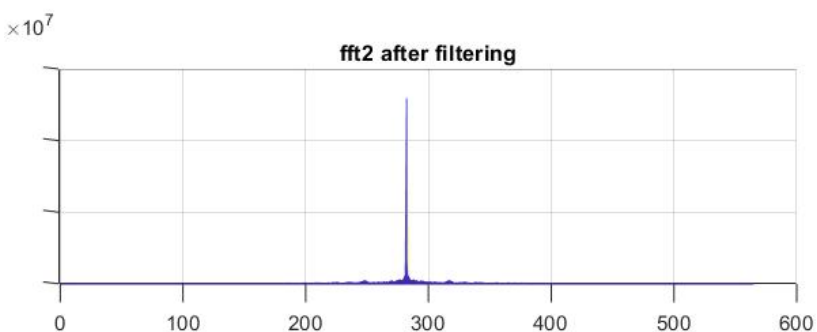
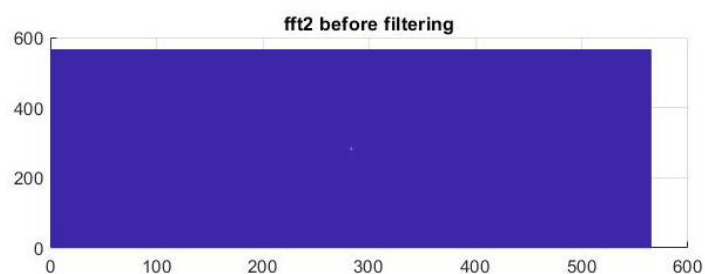
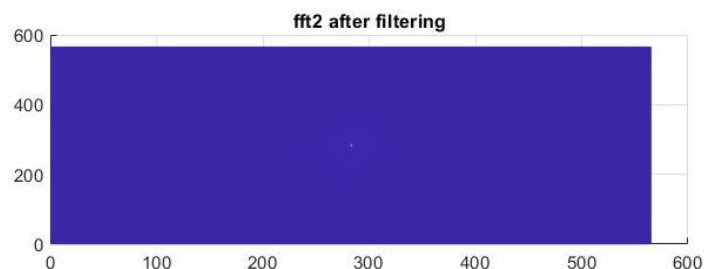
kc=1.5:



به علت کاهش نور یکبار با تغییر روند اسکیلینگ نتیجه ای با نور بیشتر و بهتر گرفتیم به طوری که اسکیلینگ را در رنج بالاتری انجام دادیم. نتیجه را با $kc=1$ در تصویر زیر مشاهده میکنید: سمت راست تصویر فیلتر شده است عکس با نام: Q1_part1_kc1_light.jpg



در نمودار های زیر fft2 عکس را قبل و بعد از فیلتر شدن آورده شده مشاهده میشود بعد از فیلتر شدن تاثیر فرکانس های پایین کمتر شده و فرکانس های بالا تقویت شده اند و بزرگتر شدند این تغییرات از نمای کناری محسوس تر اند.



بخش 2)

در این بخش با استفاده از فیلتر داده شده عکس را فیلتر کرده اما برخلاف بخش قبل در حوزه مکان عکس اصلی را با عکس فیلتر شده جمع میکنیم این روش نسبت به روش قبلی نتیجه ضعیف تر و کم کیفیتی را دارد زیرا وقتی پس از فیلتر از عکس فوریه وارون میگیریم

قسمت هایی که تغییرات بالایی دارند بسیار روشن میشوند صرف نظر از اینکه لبه تغییر روشن یا تیره بود پس وقتی عکس فیلتر شده را با عکس اصلی در حوزه فرکانس جمع میکنیم و سپس اسکیل میکنیم روشنایی عکس کم میشود و همچنین لبه ها صرف نظر از روشن بودن یا نبودن روشن تر میشوند و در نتیجه خطوط تیره عکس به جا اینکه تیره تر و عمیق تر شوند روشن تر میشوند و عمق خود را از دست میدهند و کنتراست کلی عکس کمی کمتر میشود و نتیجه نهایی کیفیت خوبی ندارد مخصوصا اینکه عکس رنگی است این روش در برخی عکس های سیاه و سفید که خیلی بلر هستند میتواند نتیجه قابل قبولی داشته باشد اما همچنان از روش قبلی ضعیف تر است و در عکس این سوال که هم رنگی است و هم به نسبت شارپ است تاثیر مناسبی نمیگذارد.

همانطور که گفته شد اضافه کردن عکس ها در حوزه مکان ایده خوبی برای فیلترینگ نیست

اما توضیح اینکه چرا فیلتر داده شده هایپس است را در زیر آوردیم :

میدانیم وقتی از تصویری $fft2$ میگیریم و شیفیت میدهیم ضرایب فرکانس های پایین در مرکز

قرار میگیرند و ضرایب فرکانس های بالا در کناره ها قرار میگیرند و هرچه به سمت کناره ای

ماتریس $fft2$ حرکت کنیم فرکانس های بالاتر عکس را خواهیم داشت این فیلتر با رابطه

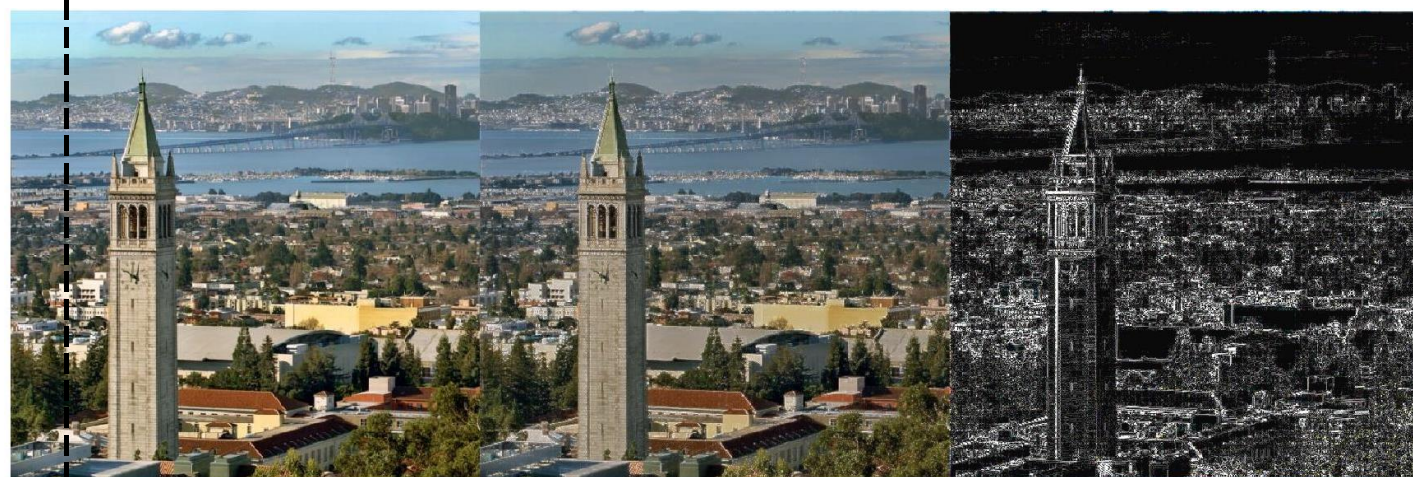
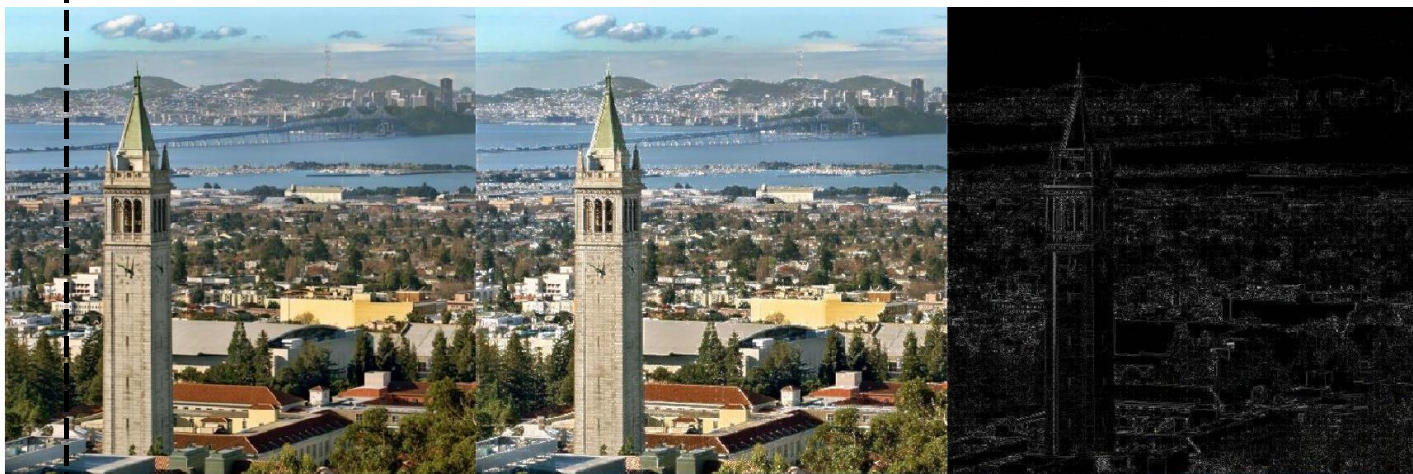
$F(u,v) = u^2 + v^2$ درایه های با فاصله بیشتر از مبدا را بیشتر تقویت میکند در نتیجه فرکانس

های بالا تقویت میشوند نتیجه عکس فیلتر شده را در تصویر سمت راست و تصویر اصلی را

سمت چپ و جمع آنها را در تصویر وسط مشاهده میکنید یک نمونه با اسکیلینگ 2^{55} و دیگری با

جمع توسط تابع $imadd$ متلب (نمونه روشن تر)

تصاویر با نام های : Q1_part2.jpg , Q1_part2_adding_imadd.jpg



سوال 2)

برای افزایش کنتراست روش های متعددی وجود دارد که تعداد زیادی از این روش ها یک هدف دارند آن هم این است که اختلاف رنگ ها و روشنایی را در تصویر افزایش دهند برای اینکار باید در نواحی تیره را کمی روشن کرد و نواحی روشن را کمی تیره کرد به گونه ای که اختلاف روشنایی پیکسل های هر ناحیه بیشتر شود و در نتیجه به طور خلاصه باید نمودار هیستوگرام عکس باز تر شود و باید با اعمال تابع مناسب بر عکس در نواحی از نمودار که تراکم پیکسلی وجود دارد این تراکم را کم کرد و نمودار را باز تر کرد .

تصویر داده شده دارای نواحی بسیار روشن در کنار نواحی بسیار تیره است پس برای این تصویر باید از تابعی استفاده کنیم که به صورت ناحیه ای عمل کند و در نواحی تیره با ضرب ضریب بزرگتر از 1 مناسب به همراه تابع مناسب با روشن تر کردن تصویر فاصله روشنایی پیکسل ها را بیشتر کند و در نواحی روشن با ضرب یک ضریب کوچک تر از 1 و اعمال تابع مناسب روشنایی پیکسل ها را کمتر کند در نتیجه کنتراست کلی عکس بیشتر میشود بدون کاهش کیفیت آن البته باید توجه کنیم که بیش از اندازه کنتراست را زیاد نکنیم تا عکس اشباع نشود .

تابع مورد استفاده در این بخش تابع (adapthisteq) میباشد که ابتدا تصویر را

به نواحی کوچکی به نام (tiles) تقسیم میکند سپس با اعمال تابع مناسب کنتراست و اختلاف پیکسل ها را در هر tile افزایش میدهد و سپس با درونیابی دوخطی tile های کوچک همسایه tile های بزرگتر را تشکیل میدهد و تغییرات روشنایی پیکسل را با توجه به توزیع داده شده انجام میدهد و مرحله این کار را ادامه میدهد تا کنتراست تصویر بهتر شود .

در نهایت نمودار هیستوگرام عکس به توزیع داده شده میل میکند .

توزیع میتواند خطی یا نمایی یا ریلی باشد برای هر نوع عکس باید از توزیع مناسب استفاده کرد مثلا توزیع ریلی برای عکس های زیر آب مناسب است .
در این تصویر ما از توزیع خطی استفاده کردیم .

ورودی ClipLimit مقدار کنتراست را کنترل میکند تا تصویر اشباع نشود .

و هرچه بیشتر باشد کنتراست خروجی بالاتر است و مقدار معمول آن 0.01 میباشد.

ورودی Nbin تعداد خانه های هیستوگرامی که در الگوریتم استفاده میشود را تعیین میکند به طور دیفالت 255 است اما با افزایش آن میتواند کیفیت خروجی و دقت آن را بیشتر کرد زیرا انگار رنج اعداد ما به تعداد بیشتری قسمت تقسیم میشود اما با افزایش Nbin اجرا محاسبات بیشتر طول میکشد .

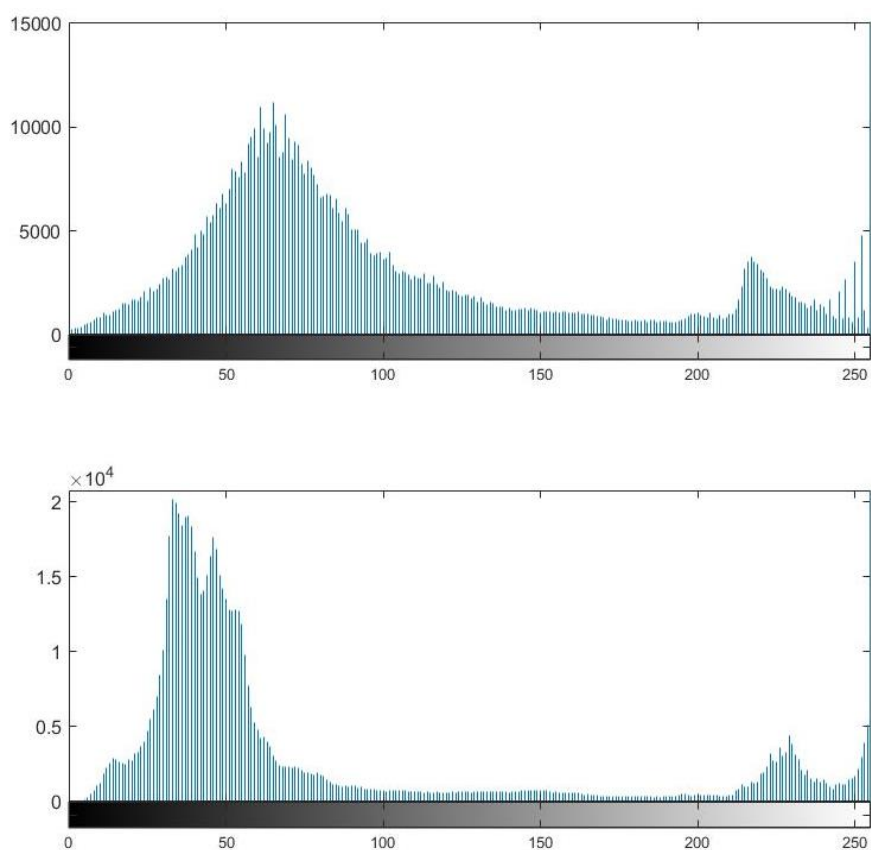
ورودی range رنج پیکسل های خروجی را تعیین میکند که به طور معمول همان 255 یا uint8 برای آن مناسب است .

اگر از توزیع نمایی یا رایلی استفاده میکنیم میتوانیم پارامتر آن را به عنوان Alpha به تابع بدهیم .

همچنین میتوانیم تعداد نواحی را به عنوان NumTiles به تابع بدهیم .

نمودار هیستوگرام عکس قبل (نمودار پایین) و بعد از تنظیم کنتراست (نمودار بالا) را در زیر آورده شده ملاحظه میشود بعد از تغییرات هیستوگرام باز تر شد و تراکم پیکسلی در یک ناحیه خاص نمودار کم شده برخلاف نمودار اولیه .

Question2_imhist.jpg



تصویر خروجی در راست و ورودی در چپ:

Question2_images.jpg



سوال 3)

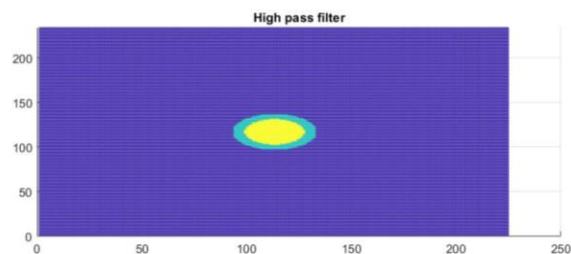
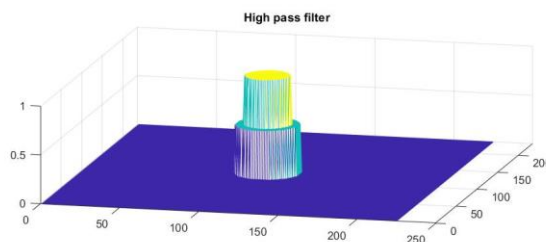
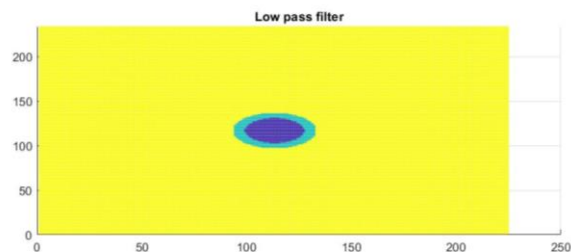
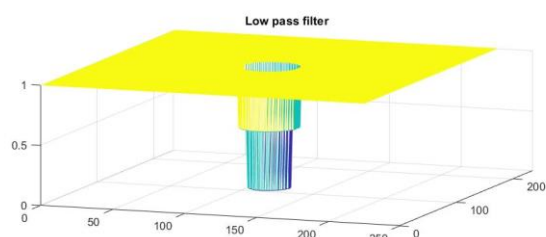
ابتدا تصاویر را از RGB به Gray تبدیل میکنیم و تصویر آلبرت را کمی میچرخانیم تا زاویه آن با مرلین برابر شود.

در این سوال از سه روش استفاده کردیم :

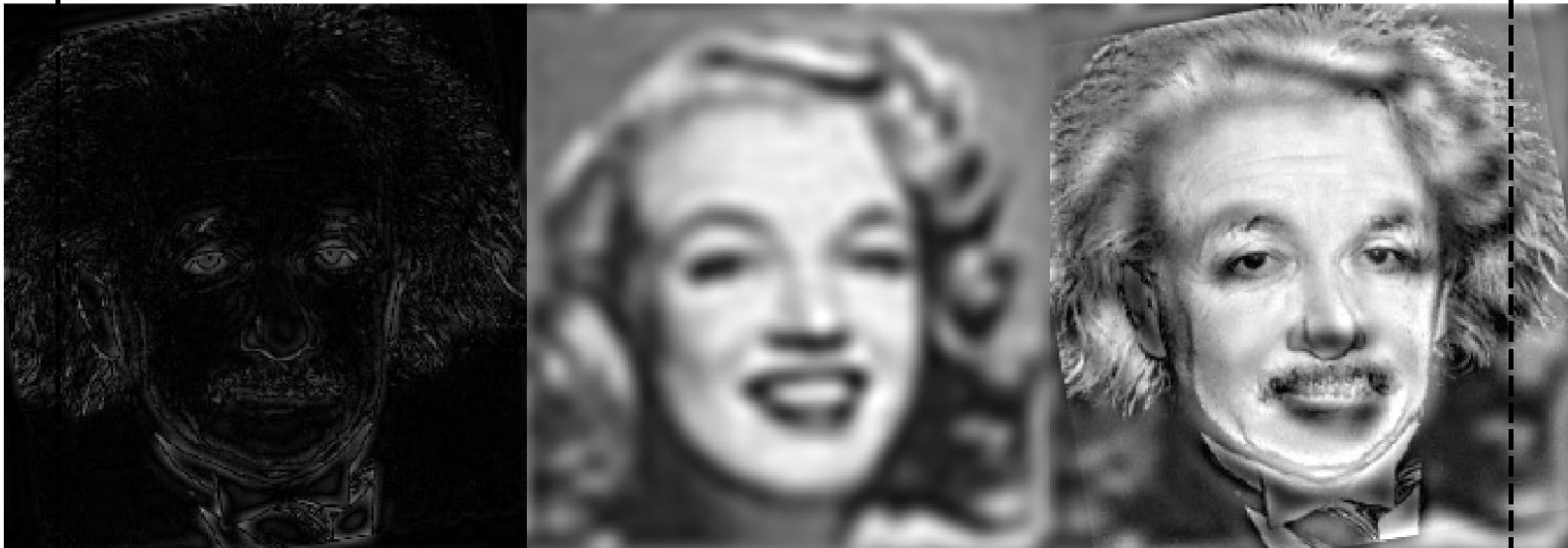
1) در روش اول از دو عکس fft2 گرفتیم و پس از شیفت دادن و انتقال فرکانس های پایین به مرکز در تصویر مرلین فیلتر پایین گذاری را اعمال کردیم که ضرایب داخل دایره به شعاع 200 رادیکال را با ضریب 1 و ضرایب بین دایره به شعاع 200 و 20 را با ضریب $1/2$ نگه داشته و بقیه ضرایب را صفر کرد سپس مکمل فیلتر بالا را بر fft2 تصویر آلبرت اعمال کردیم یعنی فرکانس های بالا (خارج دایره به شعاع 20) را با ضریب 1 و فرکانس های بین دایره به شعاع 200 رادیکال و دایره به شعاع 20 را با ضریب $1/2$ نگه داشتیم و بقیه ضرایب را صفر کردیم و در نهایت دو تصویر را در حوزه فرکانس جمع کردیم (fft2 تصاویر را جمع کردیم و از آنجایی که دو فیلتر مکمل اند تصویر در حوزه زمان خراب نخواهد شد) سپس از حاصل ifft2 گرفتیم

نتایج را در تصاویر زیر مشاهده میکنید:

فیلتر ها:



تصویر بزرگتر با نام (Question3_photos.jpg) در فایل زیپ قرار دارد.



2) در این روش با استفاده از توابع lowpass , highpass که در بخش function در زیر کد تعریف کردیم تصاویر را با استفاده از فیلتر گوسی با رابطه های زیر فیلتر میکنیم و در فیلتر هاپس برای اینکه تصویر برای جمع کردن خیلی تیره نشود بعد از فیلتر کردن کمی تصویر را روشن تر میکنیم .

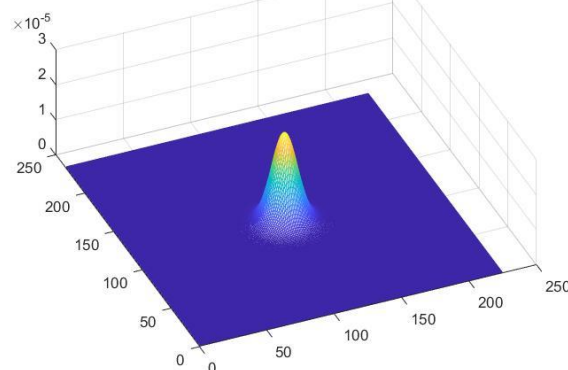
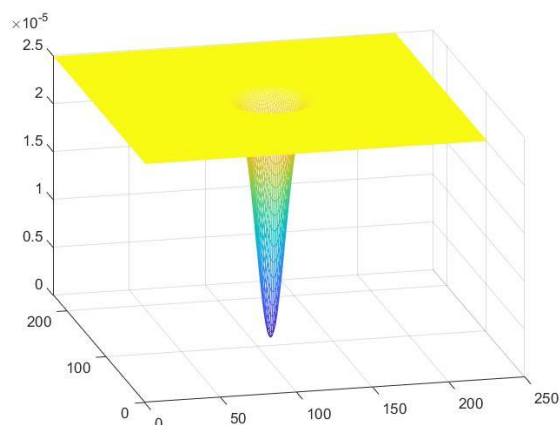
lowpass filter :

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{\frac{-x^2 - y^2}{2\sigma^2}}$$

highpass filter :

$$f(x, y) = \max(\text{lowpass}(x, y)) - \text{lowpass}(x, y)$$

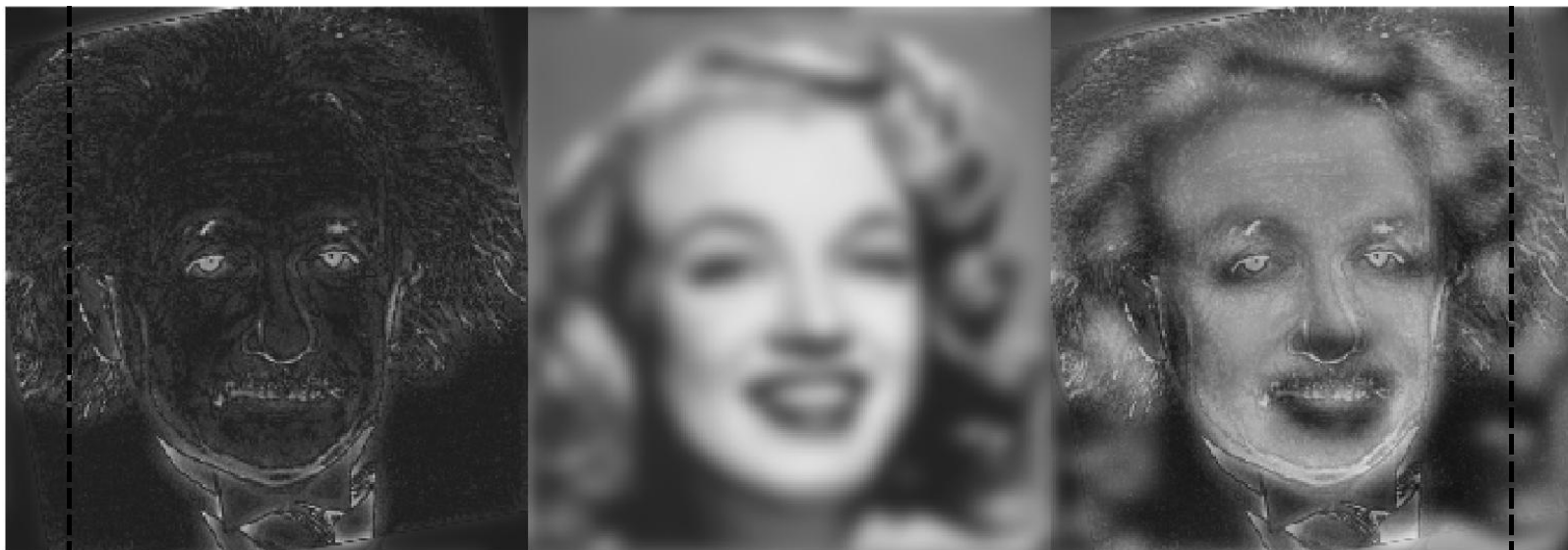
Highpass filter : Lowpass filter



برای فیلتر کردن ابتدا توابع فیلتر را در شیفت یافته fft2 تصاویر ضرب میکنیم سپس از آن ifft میگیریم. (قبل از وارون گیری هم شیفت بک میدهیم)

در نهایت دو تصویر فیلتر شده را با ضریب $1/2$ در حوزه مکان باهم جمع میکنیم و خروجی نهایی را نمایش میدهیم :

تصاویر بزرگتر در (Question3_photos_adding.jpg)



3) در روش آخر از فیلتر آماده گوسی متلب استفاده کردیم و پس از فیلتر کردن دو عکس با تابع آماده فیلتر گوسی (imgaussfilt) در حوزه مکان با ضریب $1/2$ آنها را جمع کردیم. برای هاپس کردن تصویر لوپس شده را از ضربی از عکس کم کردیم.

تصاویر بزرگتر در (Q3_usingmatlabfilters.jpg)

