تمرین ۴

هادی تمیمی
۹۶۲۲۷۶۲۴۰۸

اطلاعات گزارش
تاریخ: ۱۴۰۰/۲/۱۷

واژگان کلیدی:
تبدیل فوریه
حوزه فرکانس
فیلترینگ در حوزه فرکانس
طیف سیگنال
فاز سیگنال

چکیده

در این تمرین به حوزه فرکانس و خواص آن با استفاده از تبدیل فوریه که کار ما را برای پردازش تصویر ساده تر میکند می پردازیم. همچنین فیلترینگ در حوزه فرکانس و تفاوت های آن با فیلترینگ در حوزه مکان، شرح داده شده است و راجع به چند فیلتر خاص بحث می کنیم.

۱-مقدمه

حوزه فرکانس در زمینه پردازش تصویر ابزار قدرتمندی در اختیار ما میباشد.به کمک آن میتوان تصاویر را فشرده سازی کرد.در صورتی که نویز ناشناخته ای در تصویر وجود داشته باشد،در حوزه فرکانس میتوان آن را شناسایی نمود.

همچنین درصورتی که فیلتر بزرگی در حوزه مکان میخواهیم به تصویر اعمال کنیم ،تبدیل و اعمال کردن آن در حوزه فرکانس از نظر پردازشی بهتر خواهد بود.

در تبدیل فوریه، سیگنال سینوسی و کسینوسی به عنوان توابع پایه ای استفاده می شوند. در واقع توابع نمایی هم شکل دیگر توابع سینوسی و کسینوسی است. بنابراین تبدیل فوریه، یک سیگنال در بعد مکان را به مجموعه ای از توابع سینوسی با فرکانس های مختلف تجزیه می کند. برای بازسازی سیگنال اصلی از روی تبدیل فوریه، چون توابع پایه ای ثابت سینوسی می باش دو تنها نیاز به دانستن مجموعه فرکانس هایی است که سیگنال اصلی شامل بوده است. بنابراین، تبدیل فوریه نشان می دهد که سیگنال شامل چه فرکانس هایی می باشد.

۲-شرح تکنیکال

۴.۱ تبدیل فوریه

تبدیل فوریه برای سیگنال های دو بعدی گسسته به صورت زیر تعریف میشود:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

که در آن x,y مختصات سیگنال اصلی و u,v مختصات تبدیل شده آن می باشد.M و N هم در زمینه تصاویر سایز تصویرما میباشند.به همین ترتیب تبدیل فوریه معکوس تعریف میشود:

$$f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(\frac{ux}{M}+\frac{vy}{N})}$$

در صورتی که بخواهیم تبدیل فوریه سیگنال به خوبی قابل مشاهده باشد بهتر است که آن را شیفت دهیم.در صورتی که بخواهیم فوریه سیگنال را شیفت دهیم، میتوانیم با افزایش فاز به سیگنال اصلی این کار را انجام دهیم:

$$f(x,y)e^{j2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{n}\right)} \Leftrightarrow F(u-u_0, v-v_0)$$

اثبات میشود که در صورتی که بخواهیم مبدا تصویر را مرکز آن در نظر بگیریم، باید هر نقطه تصویر را در $(-1)^{x+y}$ ضرب نماییم.این مرحله لازم نیست ولی برای شرح بهتر نتایج انجام میشود.همچنین دقت شود در صورتی که این کار را انجام دادیم سپس به حوزه فرکانس رفتیم و عملیات مورد نظر را انجام دادیم،زمانی که به حوزه مکان بازگشتیم،باید دوباره این ضرب را انجام دهیم.

۴.۱.۱

مراحل فیلترینگ در حوزه فرکانس:

- افزودن صفر به تصویر(zero padding) و به دست آوردن تصویری با دو برابر اندازه تصویر قبلی.
- شیفت دادن، برای انتقال مبدا به مرکز.
- محاسبه تبدیل فوریه.
- افزودن صفر(zero padding) به ماتریس فیلتر و به دست آمدن ماتریسی با دو برابر اندازه های تصویر اولیه.
- شیفت دادن ماتریس فیلتر.
- محاسبه تبدیل فوریه ماتریس شیفت داده شده.
- ضرب فیلتر تبدیل یافته در تصویر تبدیل یافته.
- به دست آوردن معکوس تبدیل فوریه.
- انتخاب بخش حقیقی نتیجهTچون با تغییر ضرایب ممکن است بخش موهومی هم به تصویر اضافه شده باشد که قابل صرف نظر کردن است.
- شیفت دادن نتیجه.

- برداشتن بخشی به اندازه طول و عرض تصویر اولیه از گوشه بالا سمت چپ.

فیلتر اولی که در این بخش به بررسی آن میپردازیم به صورت زیر است:

$$f_1 = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

این فیلتر یک فیلتر میانگین گیر است. معادل تبدیل فوریه طبق مراحل بالا بدست می آوریم.


فیلتر a و نتیجه اعمال آن

این فیلتر یک فیلتر lowpass است. اعمال این فیلتر باعث میشود جزئیات تصویر کمرنگ تر شوند مانند لبه ها و کلیات تصویر را نگه میدارد.

این فیلتر جدا پذیر است و آنرا با دو فیلتر زیر میتوان بدست آورد.

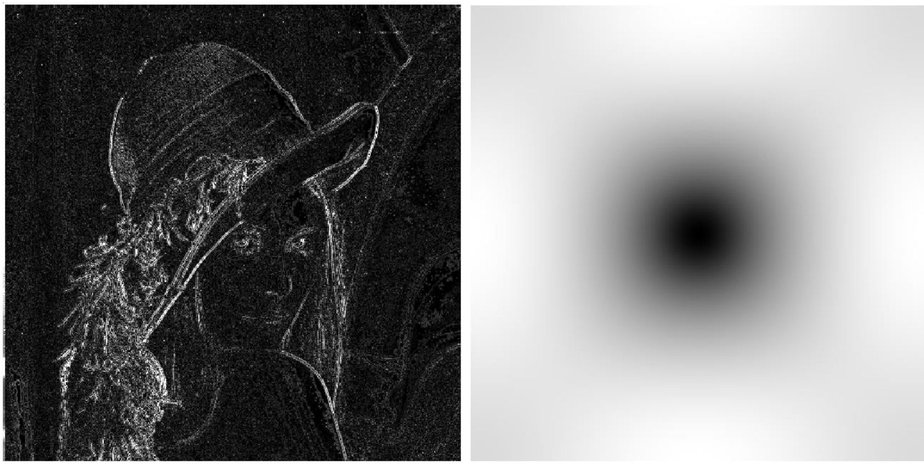
$$\frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

در حوزه فرکانس کافیست ضرب درایه ای تصویر را بر هر دو فیلتر انجام دهیم تا عملکرد فیلتر اصلی را داشته باشند.

فیلتر دوم به صورت زیر است:

$$f_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

این فیلتر برای نقاط یکنواخت مقدار صفر را برای فیلتر مرکزی قرار میدهد اما در نواحی غیر یکنواخت مانند لبه ها مقدارهای بیشتر از صفر خواهد داد.در نتیجه میتوان گفت این یک فیلتر edge detection میباشد. حال معادل تبدیل فوریه آن را بدست می آوریم.



فیلتر b و نتیجه اعمال آن

این یک فیلتر highpass میباشد. اعمال این فیلتر باعث میشود فرکانس های بالا باقی بمانند که لبه ها را در تصویر اصلی شناسایی میکند.

این فیلتر جدا پذیر نیست.

فیلتر سوم به صورت زیر است:

$$f_3 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

این فیلتر در نواحی یکنواخت مقدار پیکسل مرکزی را برای خودش قرار میدهد و در عمل تغییری ایجاد نمیکند.اما در لبه ها شدت پیکسل مرکزی را افزایش میدهد.این کار باعث میشود تصویر sharp تر شود.در عمل این فیلتر یک فیلتر edge enhancement می باشد.

حال معادل تبدیل فوریه آن را بدست می آوریم:

فیلتر c و نتیجه اعمال آن

این فیلتر در دو بعد عمودی و افقی فرکانس ها را کمرنگ تر میکند.همچنین فرکانس های مرکزی را نیز وزن کمتری می دهد.با انجام این کار باعث میشود لبه های مورب شدت بیشتری پیدا کنند و همچنین تصویر جزئیات بیشتری داشته باشد که باعث sharp تر شدن تصویر میشود.

این فیلتر جدا پذیر نیست.

۴.۱.۲

در صورتی که بخواهیم تبدیل فوریه سیگنال به خوبی قابل مشاهده باشد بهتر است که آن را شیفت دهیم.در صورتی که بخواهیم فوریه سیگنال را شیفت دهیم، میتوانیم با افزایش فاز به سیگنال اصلی این کار را انجام دهیم:

$$f(x,y)e^{j2\pi\left(\frac{u_0x}{M}+\frac{v_0y}{n}\right)} \Leftrightarrow F(u-u_0,v-v_0)$$

اثبات میشود که در صورتی که بخواهیم مبدا تصویر را مرکز آن در نظر بگیریم، باید هر نقطه تصویر را در $(-1)^{x+y}$ ضرب نماییم.این مرحله لازم نیست ولی برای شرح بهتر نتایج انجام میشود.همچنین دقت شود در صورتی که این کار را انجام دادیم سپس به حوزه فرکانس رفتیم و عملیات مورد نظر را انجام دادیم،زمانی که به حوزه مکان بازگشتیم،باید دوباره این ضرب را انجام دهیم.

در حالت بدون شیفت مبدا فرکانس در گوشه سمت چپ بالا می باشد که شاید نظر تصویری برای ارائه مناسب نیست ولی به کمک شیفت مبدا مختصات به مرکز تصویر می آید.همچنین در این حالت جزئیات لبه ها نیز مشخص است که برای شرح نتایج تصویر بهتر است.

در فرمول تبدیل فوریه مرکز مختصات جمع تمام پیکسل های تصوی است.در واقع مرکز مختصات میتواند شدت روشنایی تصویر را به ما نشان می دهد. اما در شرح نتایج این خاصیت مرکز مختصات میتواند مشکل زا باشد. برای مثال فرض کنید مقادیر غیر مرکز در رنج ده و صد باشند، در حالی که مرکز در رنج چند هزار مقدار خواهد داشت.اتفاقی که می افتاد این است که در صورتی که quantization انجام دهیم به جز مرکز نقاط دیگر صفر خواهند بود و مرکز بیشترین مقدار را خواهد داشت.لگاریتم این اختلاف زیاد بین نقاط را به خوبی کنترل میکند و اختلاف فاحش بین نقاط کاهش یافته و تصویر تبدیل یافته به خوبی قابل مشاهده می شود.

-در تبدیل فوریه تصویرخطوطی که در راستای افقی هستند، لبه های عمودی تصویر را نشان میدهند. خطوط در راستای عمود،لبه های افقی را نشان میدهند. خطوط مورب لبه های عمود بر راستایشان را نشان میدهند.

۴.۲ فیلترنگ

۴.۲.۱

-در سیگنال و سیستم ثابت میشود که کانولوشن رو حوزه مکان معادل ضرب آرایه ای در حوزه فوریه میباشد:

$$f(x,y)*h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

معنی این عبارت این است که در صورتی که فیلتری در حوزه مکان داشتیم و به کمک عملیات کانولوشن آن را بر روی تصویر اعمال میکردیم در صورتی که فیلتر و تصویر را در حوزه فرکانس به کمک تبدیل فوریه ببریم،میتوان با ضرب درایه ای عکس نهایی را بدست آورد. خروجی فیلترینگ در حوزه فرکانس و در حوزه مکان تفاوتی ندارد چون تمام عملیات های ما بر روی بخش طیف صورت میگیرد و فاز تصویر فیلتر شده با فاز تصویر اصلی یکی است.

عملیات فیلترینگ در حوزه فرکانس به سه قسمت زیر تقسیم میشود:

۱) بدست آوردن طیف عکس و فیلتر

۲) اعمال ضرب درایه ای
۳) بازگرداندن تصویر به حوزه مکان

مراحل فیلترینگ در حوضه فرکانس با جزئیات بیشتر در بخش **۴.۱.۱** گفته شده است.

-در صورتی که فیلتر در حوزه مکان کوچک باشد،عملیات کانولوشن از این سه مرحله پیچیدگی زمانی کمتری خواهد داشت.اما اگر که اندازه فیلتر آنقدر بزرگ باشد که عملیات کانولوشن پیچیدگی زمانی بیشتری داشته باشد میتوان از حوزه فرکانس برای اعمال فیلتر استفاده کرد که از نظر پردازشی بار محاسباتی کمتری خواهد داشت.

-اگر سایز فیلتر $N \times N$ و سایز تصویر $M \times M$ باشد اندازه پدینگ برای فیلترینگ مکانی $\left\lfloor \frac{N}{2} \right\rfloor$ می باشد.حال اگر عملیات کانولوشن را انجام دهیم،$\left\lfloor \frac{N}{2} \right\rfloor$ پیکسل اول و آخر از هر دو جهت پیکسل هایی خواهند بود که در آنها پدینگ لحاظ شده است.در واقع برای این که $\mathrm{Z(m,n)} = \mathrm{Y(m,n)}$ باشد باید داشته باشیم:

$$\left\lfloor \frac{N}{2} \right\rfloor < \{m, n\} < M - \left\lfloor \frac{N}{2} \right\rfloor$$

۴.۲.۲ صفر کردن برخی فرکانس های خاص

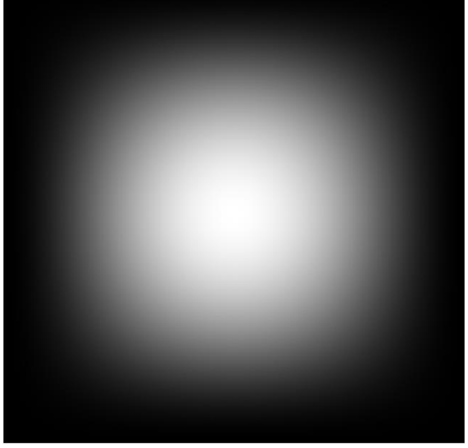
یکی از روش هایی که میتوان برخی عملیات مختلف فیلترینگ را بر روی تصویر اعمال کرد از طریق دستکاری مقادیر فرکانسی خاص می باشد.

برای این کار سه مرحله داریم:

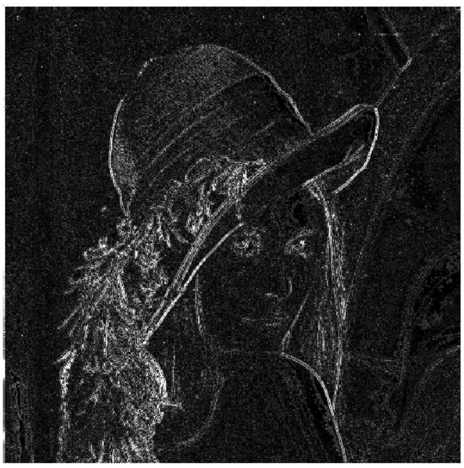
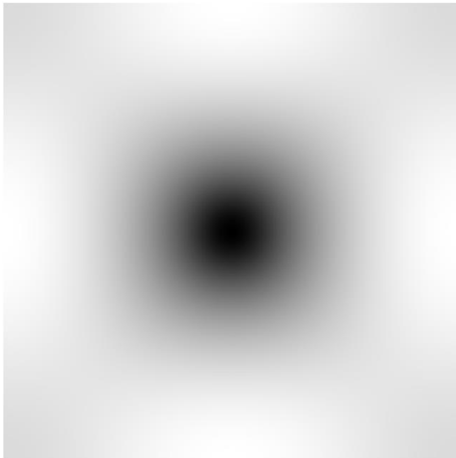
۱) ابتدا تصویر را به کمک تبدیل فوریه به حوزه فرکانس میبریم.
۲) بر روی فرکانس های مورد نظرمان عملیات مورد نظر را اعمال میکنیم(در این مورد صفر کردن فرکانس های مورد نظر)
۳) تصویر تغییر یافته را از حوزه فرکانس دوباره به حوزه مکان می آوریم.

۲-شرح نتایج

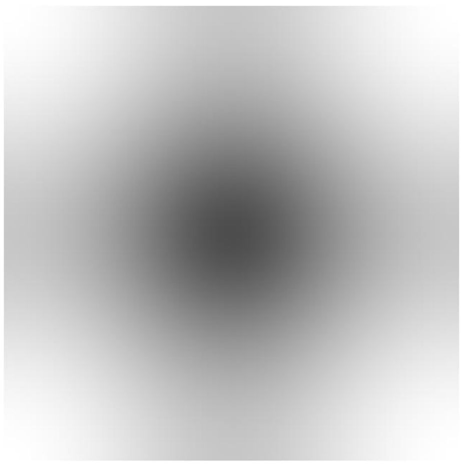
۴.۱.۱



فیلتر a و نتیجه اعمال آن

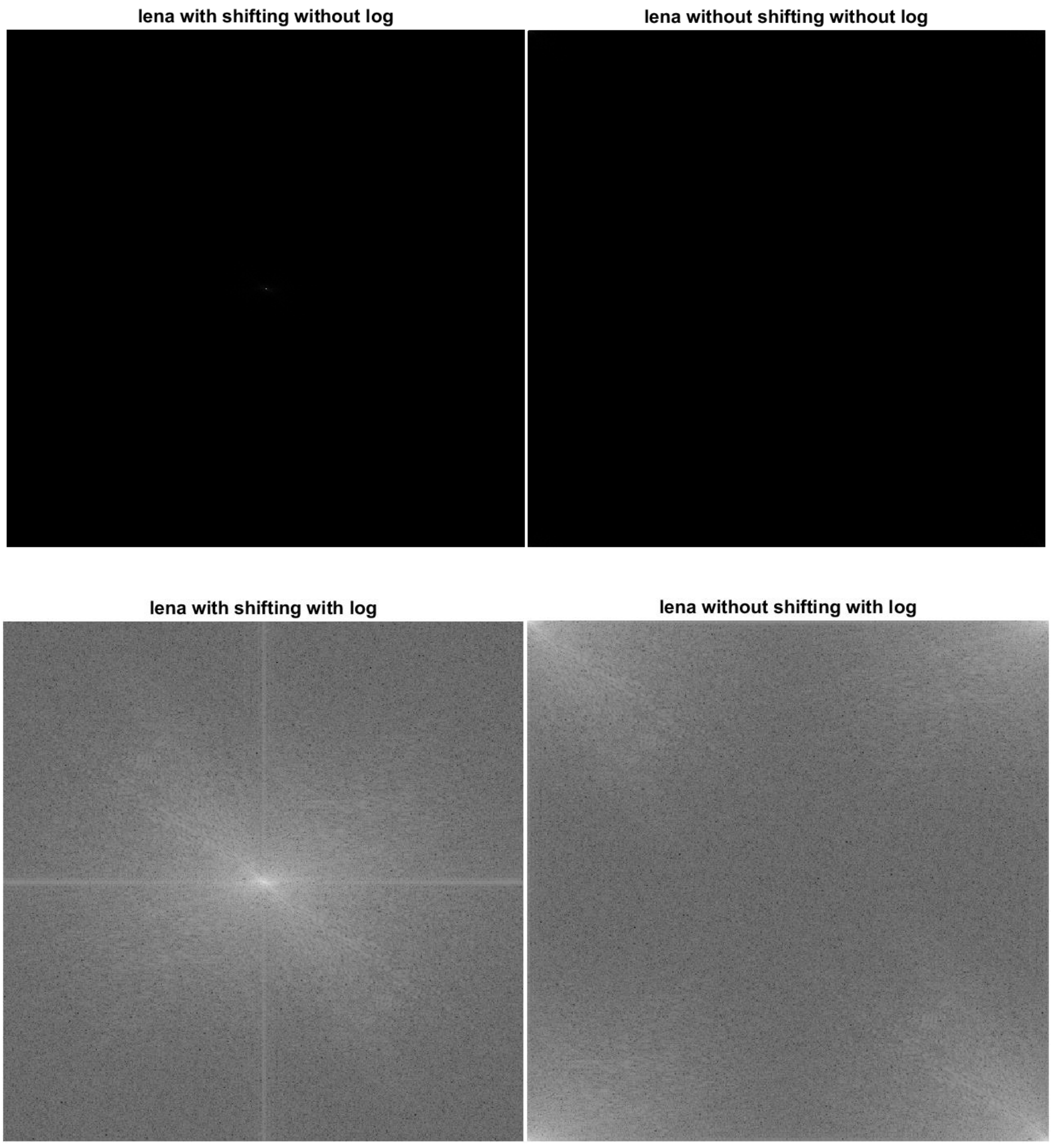
 

فیلتر b و نتیجه اعمال آن

فیلتر c و نتیجه اعمال آن

۴.۱.۲

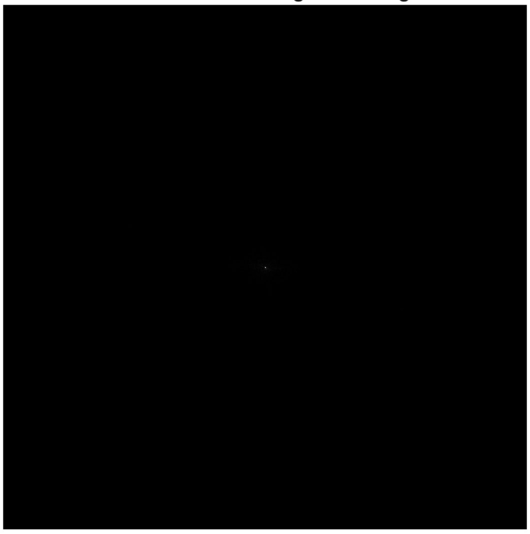


خطوطی که در راستای افقی هستند، لبه های عمودی تصویر را نشان میدهند. خطوط در راستای عمود،لبه های افقی را نشان میدهند. خطوط مورب لبه های عمود بر راستایشان را نشان میدهند. همانطور که در تصویر مشخص است خطوط مورب چپ به راست عمود بر کلاه lena می باشند.

الگوی راه راه موجود در تبدیل فوریه به دلیل الگوی راه راه لباس، رومیزی و صندلی در تصویر barbara است.

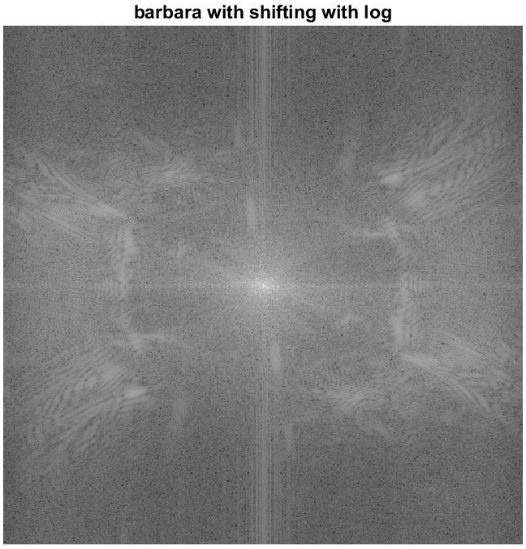
barbara orginal image

barbara with shifting without log

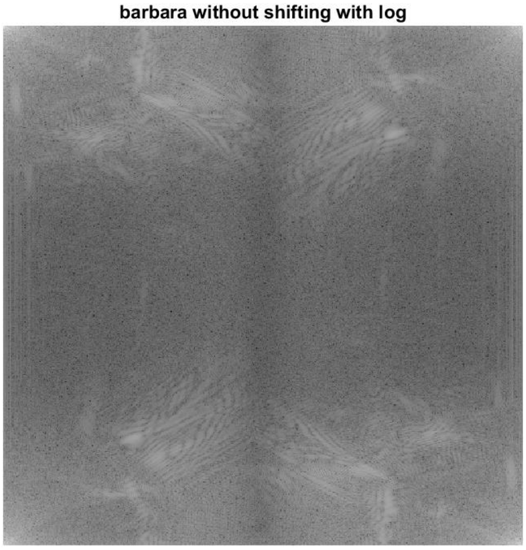


barbara without shifting without log

barbara with shifting with log



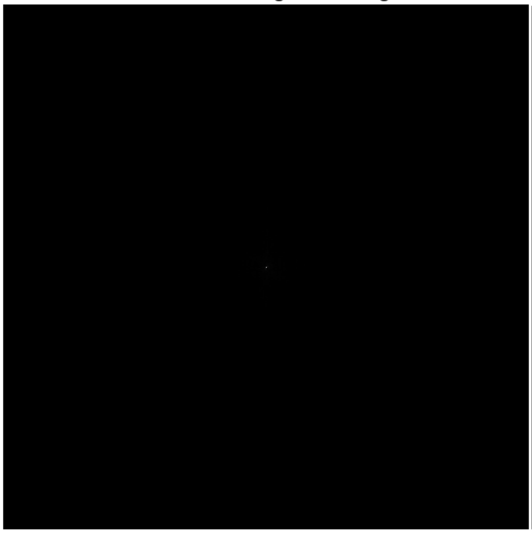
barbara without shifting with log



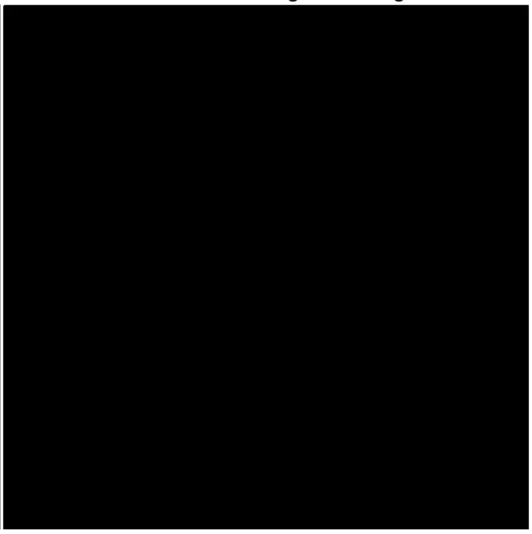
الگوی راه راه موجود در تبدیل فوریه به دلیل الگوی راه راه لباس، رومیزی و صندلی در تصویر barbara است.

f16 orginal image

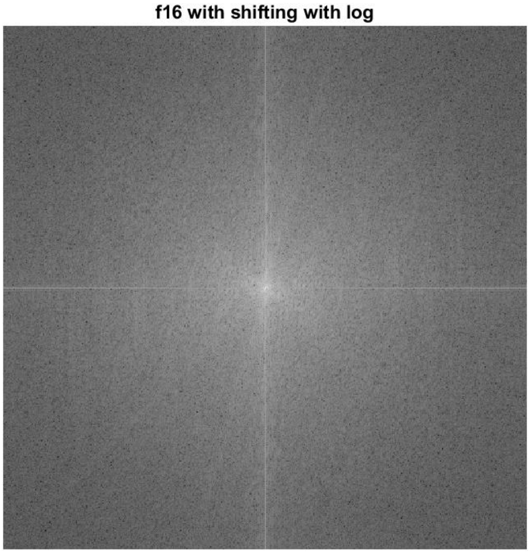
f16 with shifting without log



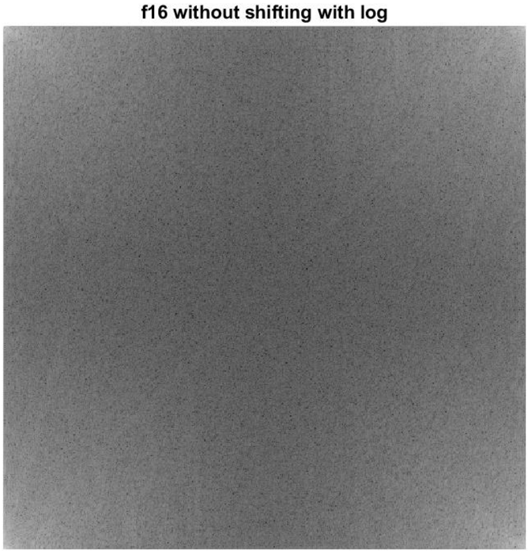
f16 without shifting without log



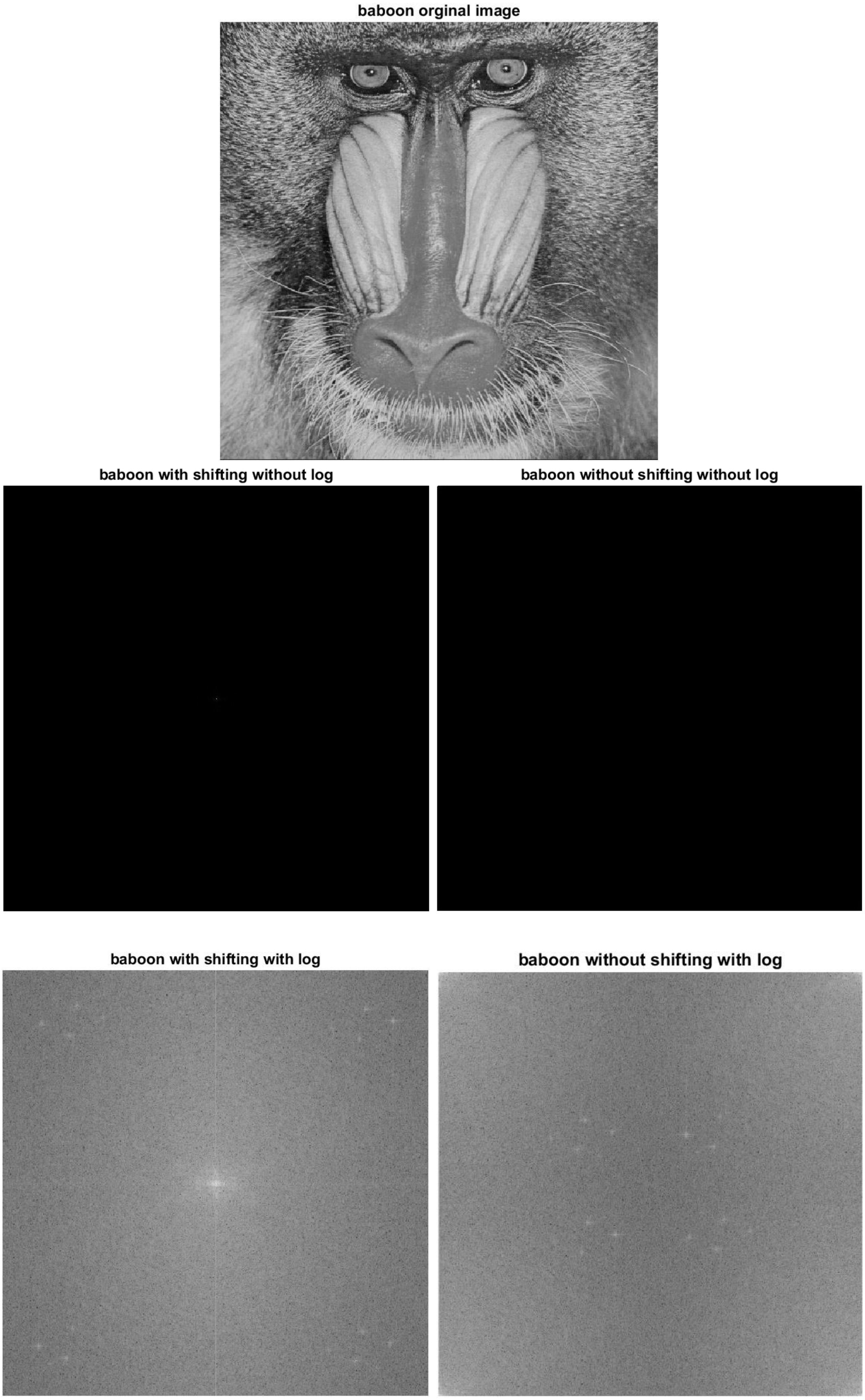
f16 with shifting with log



f16 without shifting with log

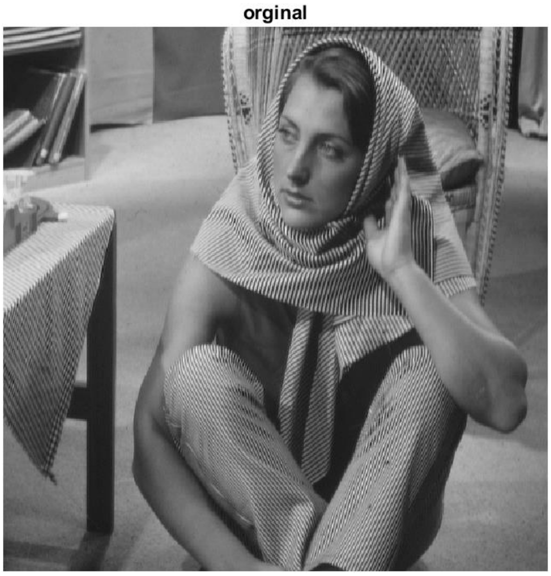


خطوط عمودی و افقی به خاطر کوه ها و بدنه جت است. خطوط موربی در راستای قطر فرعی ماتریس وجود دارند که بخاطر باله عمودی عقب جت است.

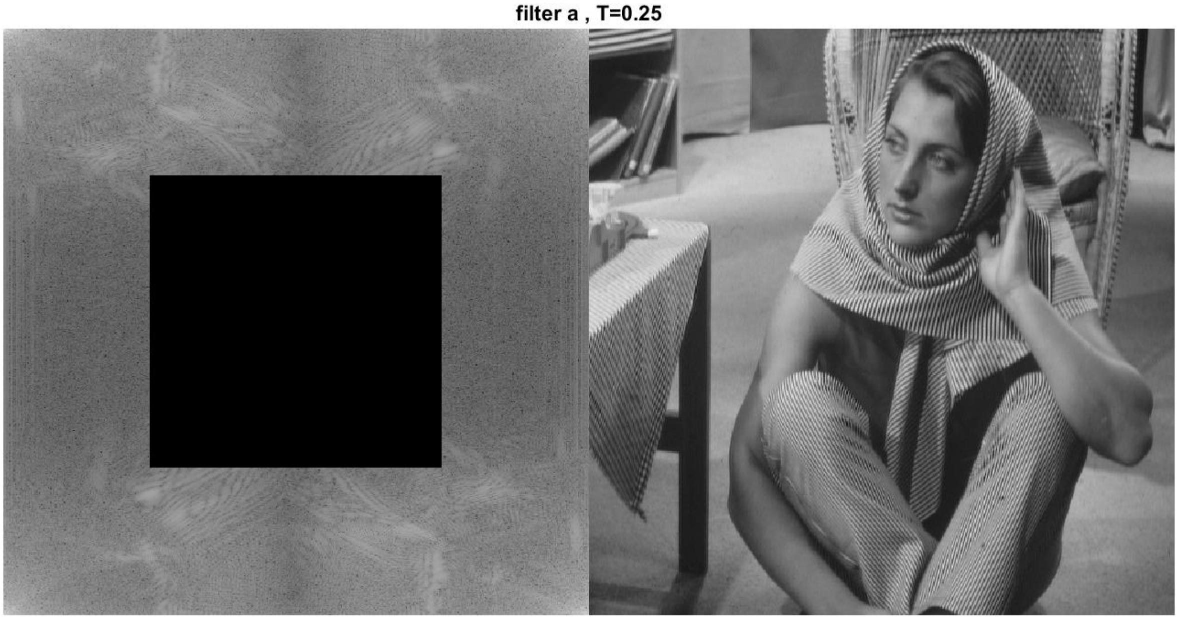


نقاطی در چهار گوشه تصویر در حوزه فرکانس دیده می شود که بنظر بخاطر الگوی ضورت و مو های بابون است.

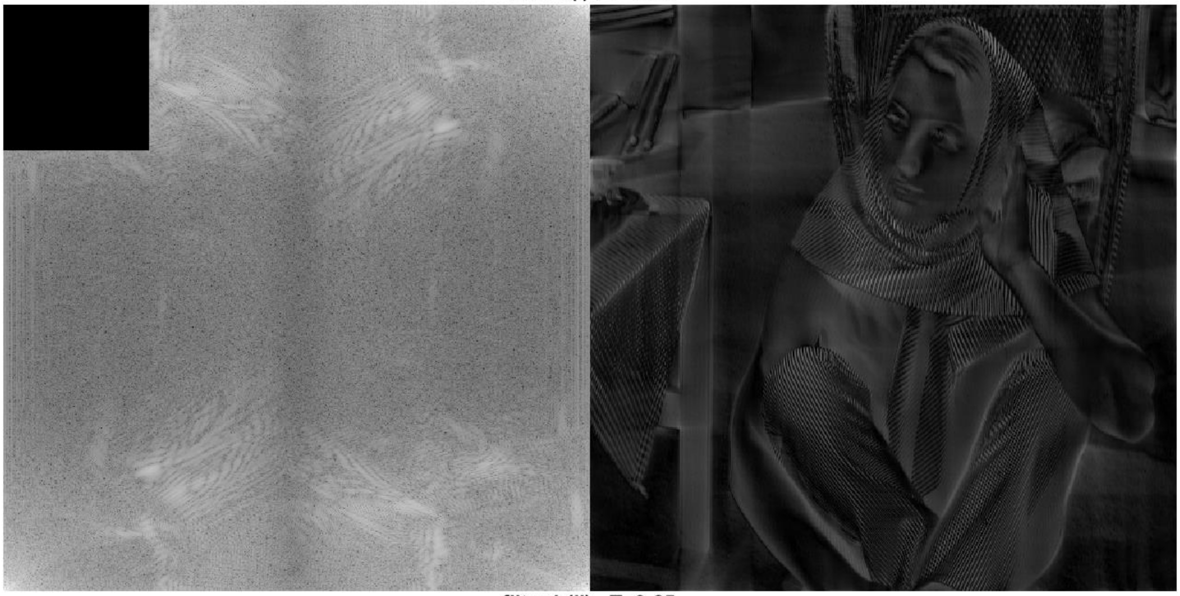
۴.۲.۲



:T=1/4

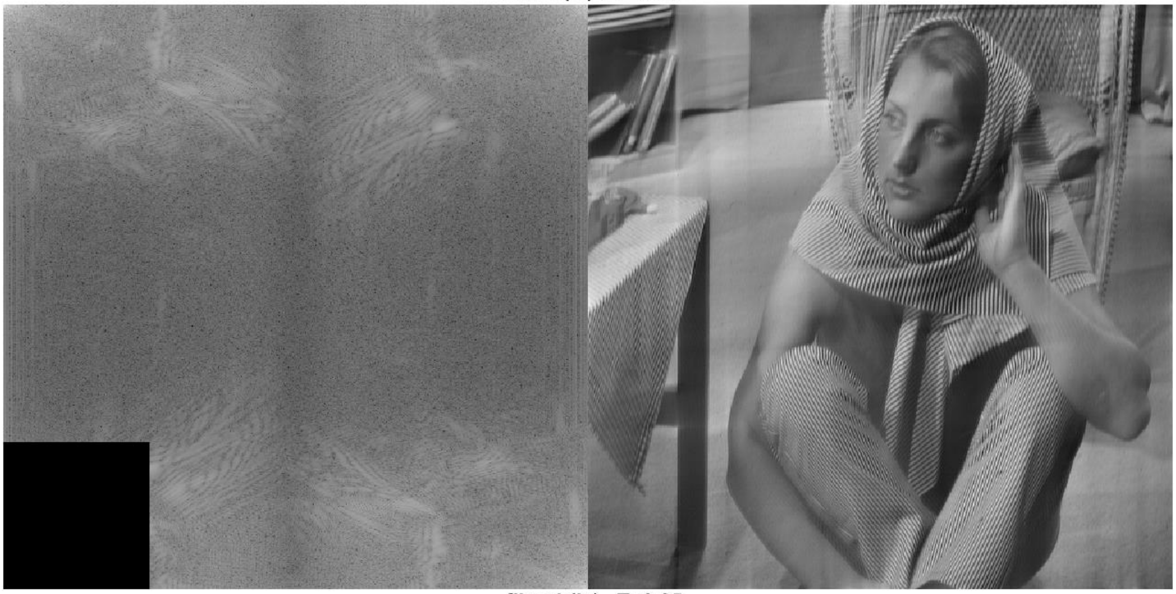


filter b(i) , T=0.25



filter b(ii) , T=0.25

filter b(iii) , T=0.25

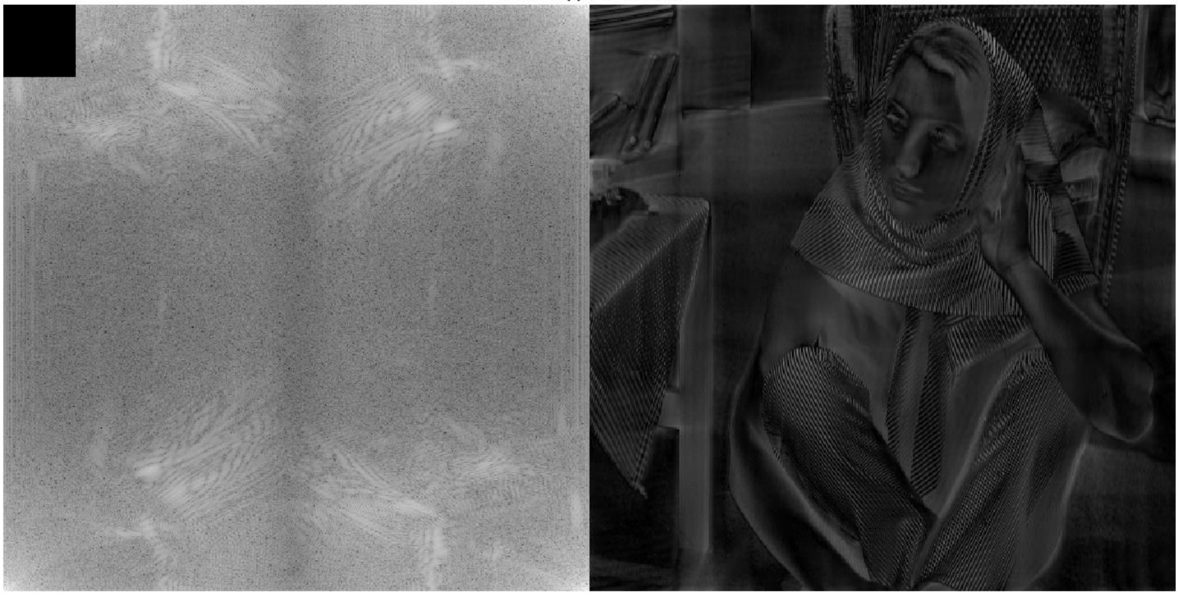


filter b(iv) , T=0.25

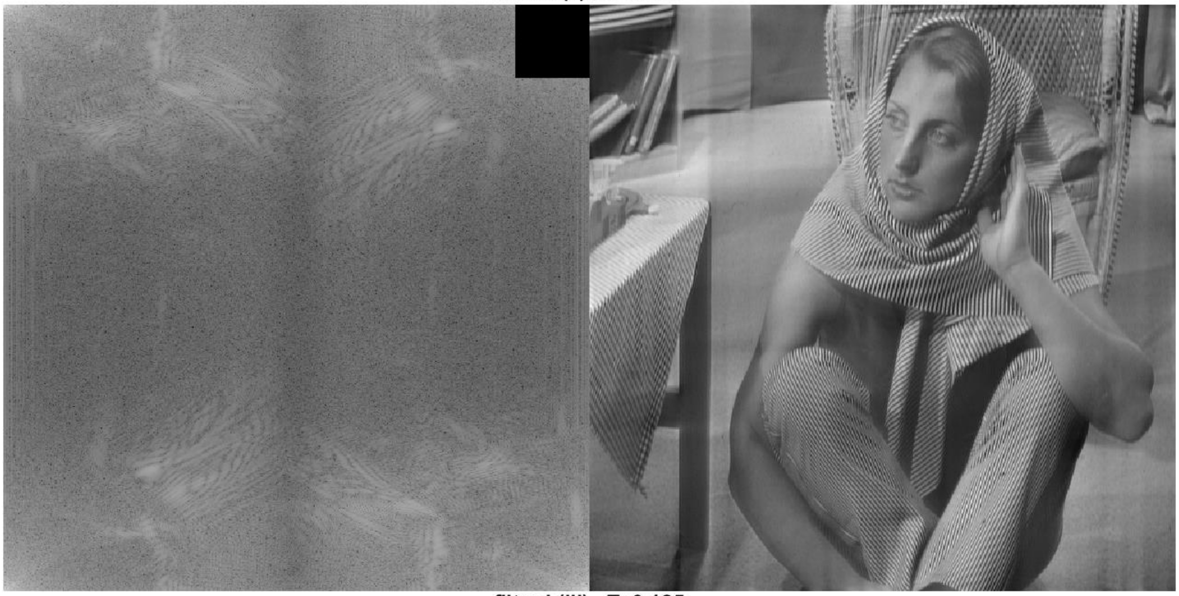
:T=1/8

filter a , T=0.125

filter b(i) , T=0.125

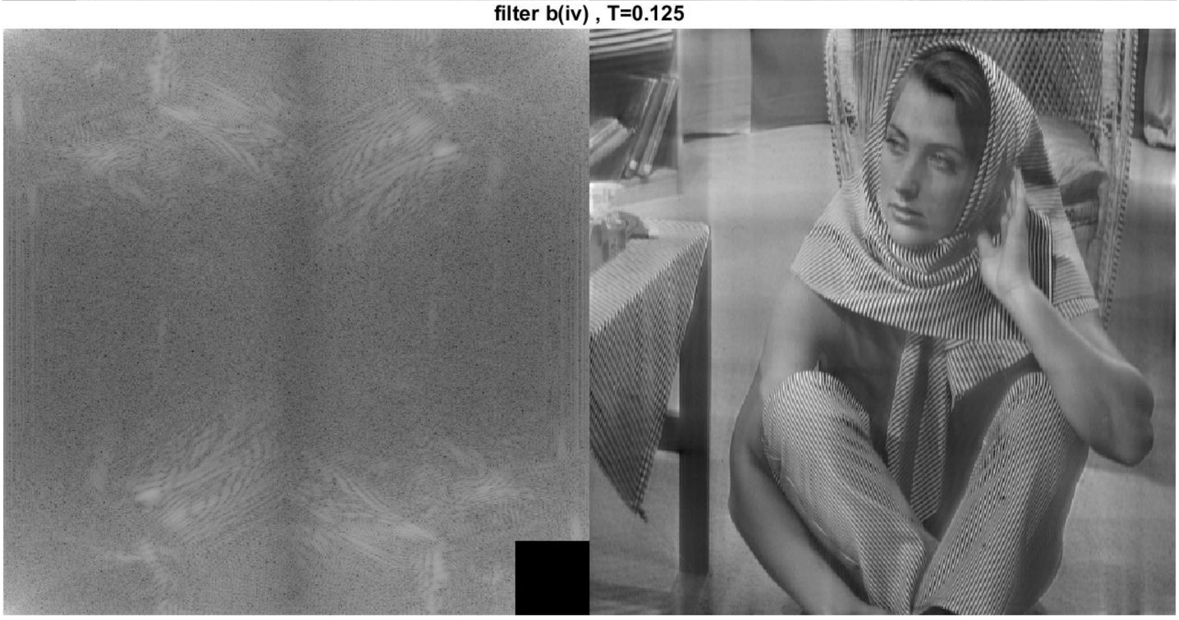


filter b(ii) , T=0.125



filter b(iii) , T=0.125

filter b(iv) , T=0.125



همانطور که مشخص است تفاوتی که T ایجاد میکنند در رنجی است که باعث صفر شدن آن می شوند.
هرچه T بزرگتر باشد ،محدوده فرکانسی که صفر میشود کوچکتر و در نتیجه تاثیرش کم تر می شود.
(اعمال شده روی فرکانس شیفت داده نشده)

-فیلتر a بخش زیادی از فرکانس های بالا را صفر میکند و در واقع یک low pass filter است.این کار باعث میشود جزئیات(لبه ها)تصویر کمتر شوند و تصویر تار شود

- فیلتر b(i) در دلیل آن که مبدا (نقطه(0,0)) در رنج صفره شده هایش می باشد،تیره شده شده است، زیرا همانطور که در قسمت ۴.۱.۲ گفته شد مبدا قبل از شیفت جمع تمام پیکسل های تصویر است در واقع مبدا مختصات میتواند شدت روشنایی تصویر را نشان می دهد.

- فیلتر های b هر کدام بخشی از فرکانس های پایین تصویر را صفر میکنند.این کار باعث میشود که خطوط مورب به صورت نصفه در هرکدام از این تصاویر حذف شود و خطوط و قسمت های تیره ای در تصویر ایجاد می کند.

کد ها:

۴.۱.۱

لود کردن تصویر و فراخوانی توابع و نمایش خروجی:

```
img = imread('Images\4\Lena.bmp');
img = rgb2gray(img);

%change comment for diffret filter
%filter = [1 2 1;2 4 2;1 2 1]/16;
%filter = [-1 -1 -1;-1 8 -1;-1 -1 -1];
filter = [0 -1 0;-1 5 -1;0 -1 0];

[M,N] = size(img);
f = dft_img(filter,M,N);
filtred_img = freq_filtering(img,filter);
filtred_img = uint8(filtred_img);

figure
imshow(f);
figure
imshow(filtred_img)

imwrite(filtred_img,'out.png');
```

تابع dft:

```
function new_img = dft_img(img,M,N)
```

```
    img = double(img);
    img = fft2(img,M,N);
    img = fftshift(img);
    img = abs(img);
    img = log10(img+1);
    Max = max(max(img));

    img = img *(255/Max);
    new_img = uint8(img);
end
```

تابع اعمال فیلتر حوزه فرکانس:

```
function new_img = freq_filtering (image,filter)
        [M,N] = size(image);
        P = 2*M;
        Q = 2*N;

        padded_img = zeros(P,Q,'uint8');
        padded_img(1:M,1:N)=image;
        fourier_img = fft2(padded_img,P,Q);
        fourier_filter = fft2(filter,P,Q);
        fourier_img = fourier_img.*fourier_filter;
        inverse_img = ifft2(fourier_img);
        new_img = abs(inverse_img);
        new_img = new_img(1:M,1:N);


    end
```

۴.۱.۲

لود کردن تصاویر:

```
lena = imread('Images\4\Lena.bmp');
barbara = imread('Images\4\Barbara.bmp');
f16 = imread('Images\4\F16.bmp');
baboon = imread('Images\4\Baboon.bmp');

imgs = {lena ,barbara, f16, baboon};
names=["lena","barbara", "f16", "baboon" ];
```

فراخوانی تابع ها با ترتیب مناسب برای هر مورد برای تصاویر و نمایش خروجی:

```
for i=1:4
img = rgb2gray(imgs{i});
figure
imshow(img);
title(names(i)+" orginal image")

img = double(img);
[M,N] = size(img);

fimg = fft2(img,M,N);

%with shifting with log
img = fftshift(fimg);
img = abs(img);
img = log(img);
Max = max(max(img ));
img = (img)*(255/Max);
```

```
img = uint8(img);
figure
imshow(img);
title(names(i)+" with shifting with log");

%without shifting with log
img = abs(fimg);
img = log(img);
Max = max(max(img ));
img = (img)*(255/Max);
img = uint8(img);
figure
imshow(img);
title(names(i)+" without shifting with log")

%with shifting without log
img = fftshift(fimg);
img = abs(img);
Max = max(max(img ));
img = (img)*(255/Max);
img = uint8(img);
figure
imshow(img);
title(names(i)+" with shifting without log")

%without shifting without log
img = abs(fimg);
Max = max(max(img ));
img = (img)*(255/Max);
img = uint8(img);
figure
imshow(img);
title(names(i)+" without shifting without log")
end
```

۴.۲.۲

لود کردن تصاویر و تعریف متغیر ها:

```
img = imread('Images\4\Barbara.bmp');

img = rgb2gray(img);
figure
imshow(img);
title("orginal");
img = double(img);
T=[1/4 ; 1/8]
```

فراخوانی تابع ها با ترتیب مناسب برای T های مختلف و نمایش خروجی:

```
for i=1:size(T)
x = uint8(freq_filtering(img,0,T(i)));
figure
imshow(x);
title("filter a , T="+T(i));
```

```
x = uint8(freq_filtering(img,1,T(i)));
figure
imshow(x);
title("filter b(i) , T="+T(i));

x = uint8(freq_filtering(img,2,T(i)));
figure
imshow(x);
title("filter b(ii) , T="+T(i));
x = uint8(freq_filtering(img,3,T(i)));
figure
imshow(x);
title("filter b(iii) , T="+T(i));
x = uint8(freq_filtering(img,4,T(i)));
figure
imshow(x);
title("filter b(iv) , T="+T(i));
end
```

تابع اعمال فیلتر های داده شده حوزه فرکانس:

```
function new_img = freq_filtering (image,x,T)
    [M,N] = size(image);

    fourier_img = fft2(image,M,N);

    for k = 1:M
        for l=1:N

            switch x
                case 0
                    if((T*N<k && k<(1-T)*N) &&(T*N<l &&l<(1-T)*N))
                         fourier_img(k,l) = 0;
                    end
                case 1
                    if((0<k && k<T*N) && (0<l && l<T*N))
                        if(k~=0 || l~=0)
                            fourier_img(k,l) = 0;
                        end
                    end
                case 2
                    if((0<=k && k<=T*N)&&((1-T)*N<= l))
                        fourier_img(k,l) = 0;
                    end
                 case 3
                    if((0<=l && l<=T*N)&&((1-T)*N<= k))
                        fourier_img(k,l) = 0;
                    end
                 case 4
                    if((1-T)*N<=k && (1-T)*N<=l)
                        fourier_img(k,l) = 0;
                    end
            end
        end
    end

    x = dft_im(fourier_img);
    inverse_img = ifft2(fourier_img);
```

```
    y = abs(inverse_img);
    new_img = zeros(M,N);
    new_img(1:M,1:N) =x;
    new_img(1:M,N+1:2*N)=y;

end
```

تابع dft:

```
function img = dft_im(fourier_img)
    img = abs(fourier_img);
    img = log10(img+1);
    Max = max(max(img));
    img = img *(255/Max);
    img = uint8(img);

end
```

پایان