حوزه ی فرکانس

علیرضا لرستانی

اطلاعات گزارش

تاریخ: 1399/09/21

واژگان کلیدی:
فیلتر
تبدیل فوریه
عکس تبدیل فوریه
طیف
Padding
Dft
Idft
Dc

چکیده

دو حوزه مکان (spatial domain) و فرکانس (frquency domain) حوزه های پردازش تصویر هستند که در گذشته به حوزه ی مکان پرداختیم و در این تمرین تمرکز خود را بر حوزه فرکانس خواهیم گذاشت.

با تبدیل فوریه آشنا خواهیم شد و فیلترینگ در حوزه فرکانس را بررسی خواهیم کرد

۱-مقدمه

نوشتار حاضر، به بررسی و پیاده سازی (با زبان برنامه نویسی متلب) تعدادی عملیات پردازش تصویر در حوزه فرکانسی یا frequency domain میپردازد.

با تبدیل فوریه و بردن یک تصویر به فضای فرکانسی و انجام عملیات های مختلف در این فضا آشنا خواهیم شد.

و در نهایت عملیات هایی را بر روی تعدادی تصویر بررسی و امتحان خواهیم کرد.

۲-شرح تکنیکال

در این تمرین درباره Fourier Transform صحبت خواهیم کرد:

تبدیل فوریه ابزاری مهم در زمینه پردازش تصویر است. از این تبدیل برای تجزیه تصاویر به عناصر سینوسی و کسینوسی استفاده میشود. این تبدیل تصویر را به حوزه فرکانسی میبرد.

تبیل فوریه در زمیینه های فراوانی کاربرد دارد، از جمله: آنالیز تصاویر، فیلتر کردن، بازسازی تصاویر و تجزیه آن ها.

از آنجایی که تصاویر دیجیتال مورد بحث ما هستند، این مبحث را به تبدیل فوریه گسسته (DFT) محدود خواهیم کرد.

* پست الکترونیک نویسنده مسئول:
alirezalorestani2010@gmail.com

1. دانشجو، دانشکده مهندسی، دانشگاه فردوسی مشهد

Dft نیز نوعی از تبدیل فوریه است اما همانطور که مورد از نامش پیداست همه ی فرکانس های تشکیل دهنده ی یک تصویر را شامل نمیشود.

اما این تعداد باید به اندازه ای باشد که بتواند تصویر را به خوبی حوزه مکان توصیف کند.

فیلترهای مکانی که قبلا استفاده میشد، میتواند در حوزه فرکانس نیز انجام شود. در حوزه مکانی، عمل فیلترینگ با کانولوشن تصویر و ماسک فیلتر به دست می آید. در حوزه فرکانس، این کار با ضرب تبدیل فوریه فیلتر و تصویر به دست می آید.

تبدیل فوریه (گسسته) در دو بعد برای یک تصویر با اندازه NxN به صورت زیر تعریف میشود:

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

در این رابطه f (x , y) به تصویر در حوزه مکان، و تابع نمایی که در ادامه مشاهده میکنید نقش اصلی بردن تصویر از حوزه مکان به حوزه فرکانس را ایفا میکند.

این رابطه را میتوان این گونه تفسیر کرد:

هر نقطه در فضای فرکانس از مجموع حاصل ضرب هر نقطه ی تصویر در حوزه ی مکان در تابع نمایی پایه ای به دست می آید.

توابع پایه ای در واقع همان سیگنال های سینوسی و کسینوسی هستند.

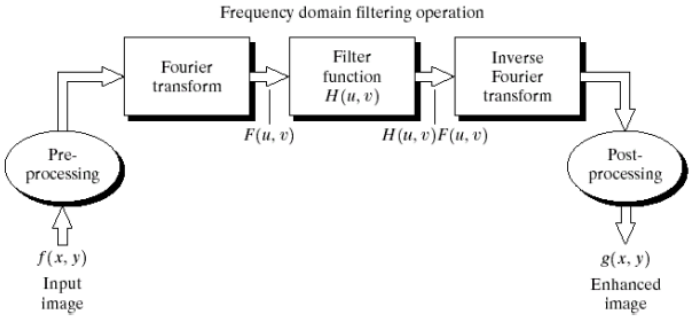
پس از اعمال تبدیل فوریه و انجام عملیات های مورد نظرمان در فضای مذکور به سادگی میتوانیم مجددا تصویر را با اسفاده از رابطه ی زیر به حوزه ی مکان برگردانیم:

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

قابل ذکر است که از 1/MN برای نرمال سازی تصویر استفاده میشود. این ضریب میتواند در فرمول تبدیل فوریه نیز به کار رود اما نکته ی مهم این است که تنها در یکی از این 2 تبدیل باید استفاده شود نه هر دوی آنها.

در ادامه مراحل فیلترینگ در حوزه فرکانس را با هم بررسی خواهیم کرد:

تصویر زیر نشان دهنده ی مراحل فیلترینگ در حوزه فرکانس به طور خلاصه است.



- بدست آوردن P و Q با توجه به اندازه MxN تصویر عموماً P = 2M و Q = 2N انتخاب می شود.
- ساختن تصویر f_p با عمل افزودن صفر (zero padding) به تصویر f.
- ضرب کردنf_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 به دست آوردن g_p
- برداشتن بخشی به اندازه MxN از گوشه ی بالای سمت چپ g_p

فیلتر جداپذیر (separable)

یک فیلتر را جداپذیر گویند اگر بتوان آن را به صورت ضرب یک بردار ستونی در یک بردار سطری نوشت. به عبارت دیگر اگر فیلتر را یک ماتریس در نظر بگیریم، فیلتر جداپذیر یک ماتریس با رتبه 1 است. اگر فیلتر A جداپذیر باشد خواهیم داشت:

A = v * h

فیلتر بالا گذر:

بر روی پیکسل ها با فرکانس پایین تغییرات ایجاد میکنند و فرکانس های بالا را عبور میدهند

فیلتر پایین گذر:

بر روی پیکسل ها با فرکانس بالا تغییرات ایجاد میکنند و فرکانس های پایین را عبور میدهند

۳-شرح نتایج و نتیجه گیری

۴.۱.۱

در این بخش 3 فیلتر a و b و c را بررسی خواهیم کرد:

- فیلتر a :

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

- فیلتر b :

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

- فیلتر c :

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

ابتدا هر یک از این فیلتر ها را با استفاده از تبدیل فوریه به حوزه ی فرکانسی میبریم، این عمل به کمک تابع fft2 انجیام میشود.

سپس برای هر یک مقدار magnitude را با استفاده از تابع fftshift به دست آورده ایم.

برای فیلتر a مقدار محاسبه شده به صورت زیر است:

0.0625	0.2500	0.0625
0.2500	1.0000	0.2500
0.0625	0.2500	0.0625

برای فیلتر b مقدار محاسبه شده به صورت زیر است:

9	9	9
9	0	9
9	9	9

برای فیلتر c مقدار محاسبه شده به صورت زیر است:

7.0000	4.0000	7.0000
4.0000	1.0000	4.0000
7.0000	4.0000	7.0000

سپس برای فهمیدن کاربرد هر یک از این سه فیلتر، تاثیر آن ها بر روی تصویر سیاه و سفید شده Barbara را مشاهده میکنیم.

تصویر اصلی:

با استفاده از تابع rgb2gray تصویر را سیاه و سفید میکنیم:

حال تاثیر هر یک از فیلترها را مشاهده میکنیم:

تصویر Barbara پس از اعمال فیلتر a :

در ضمن از آنجایی که فیلتر a را میتوانیم به صورت زیر بنویسیم میتوان گفت که جدایی پذیر است:

$$\begin{bmatrix}1\\2\\1\end{bmatrix} * [1 \quad 2 \quad 1]$$

تصویر Barbara پس از اعمال فیلتر b :

فیلتر b جدایی پذیر نیست.

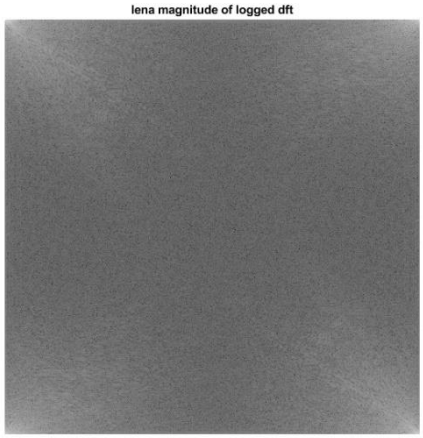
تصویر Barbara پس از اعمال فیلتر c :

فیلتر c نیز جدایی پذیر نیست.

۴.۱.۲

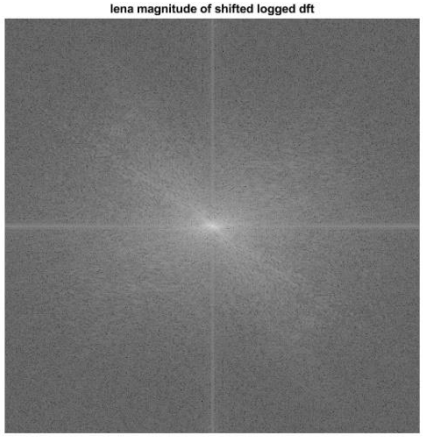
در این بخش ابتدا dft را بر روی هر سه تصویر lena و baboon و f16 اجرا میکنیم. سپس حالت های با شیفت، بدون شیفت، با لگاریتم و بدون لگاریتم را بررسی میکنیم.

lena magnitude of logged dft

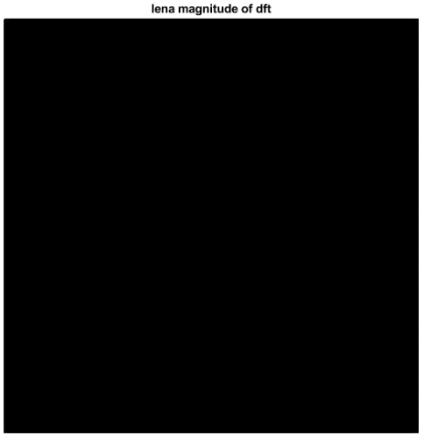


lena original image

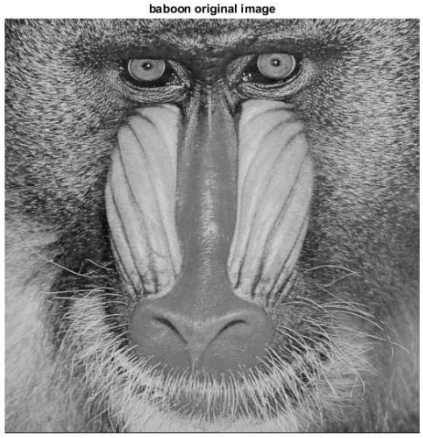
lena magnitude of shifted logged dft



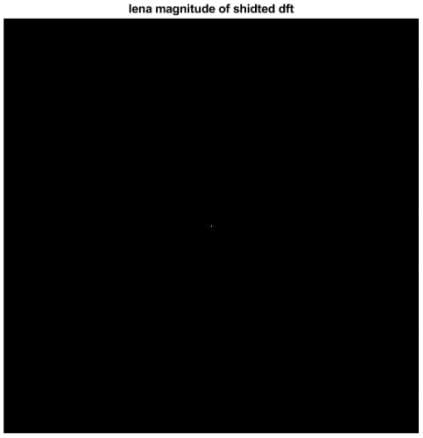
lena magnitude of dft



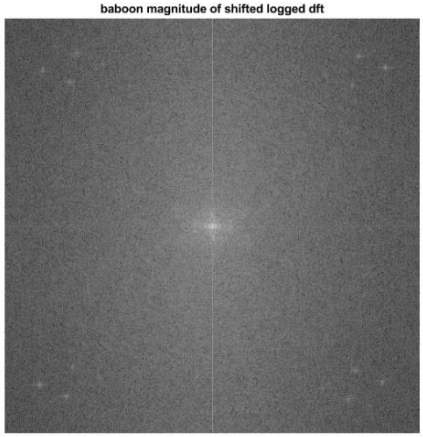
baboon original image



lena magnitude of shidted dft



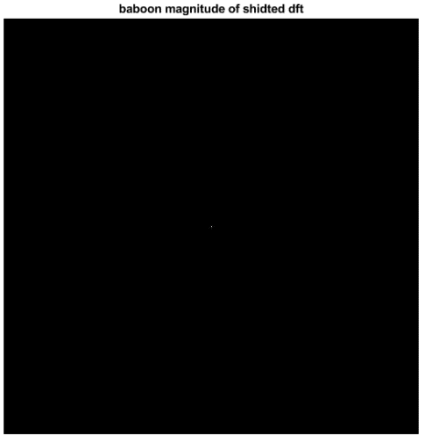
baboon magnitude of shifted logged dft



baboon magnitude of dft

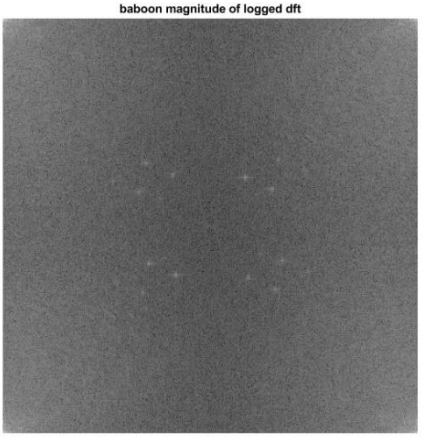
f16 original image

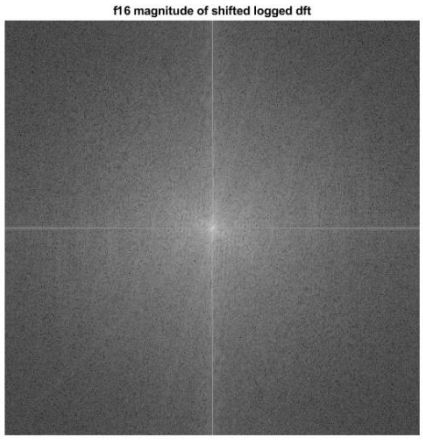
baboon magnitude of shidted dft

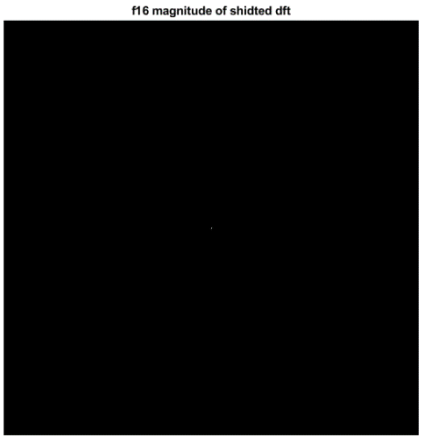


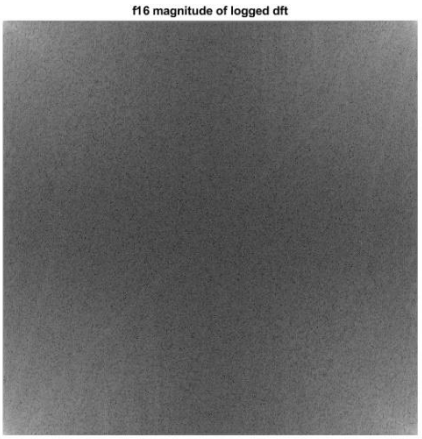
f16 magnitude of dft

baboon magnitude of logged dft









همانطور که مشاهده میشود تصاویر بدون اعمال لگاریتم تصاویری سیاه هستند که پس از شیفت در مرکز آن ها یک نقطه ی سفید مشاهده میشود. علت آن این است که جمع فرکانس تمام پیکسل های تصویر در مرکز آن قرار گرفته است که مقداری بسیار بزرگ است و اختلاف آن با سایر نقاط بسیار زیاد است.

با اعمال لگاریتم این تصویر نرمال تر شده و قابل درک تر میشود.

در تصاویر لگارتیم گرفته شده مشاهده میشود پس از شیفت که فرکانس های بالا همگی در وسط تصویر جمع شده اند (که قبل از شیفت در کناره ها قابل مشاهده هستند) و هر چه از نقطه ی میانی تصویر فاصله میگریم به سمت فرکانس های پایین تر میرویم.

۴.۲.۱

سایز تصویر ورودی 256x256 است بنابراین برای dft ما باید برای آن padding ایجاد کنیم و سایز تصویر را تغییر داده و آن را 512x512 کنیم.

همچنین سایز فیلتر را نیز (512x512) میکنیم تا با اندازه تصویر یکی شود.

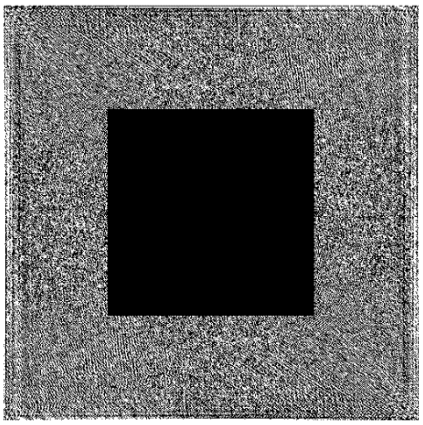
در ادامه میتوانیم به سادگی محاسبات بیان شده را انجام دهیم.

۴.۲.۲

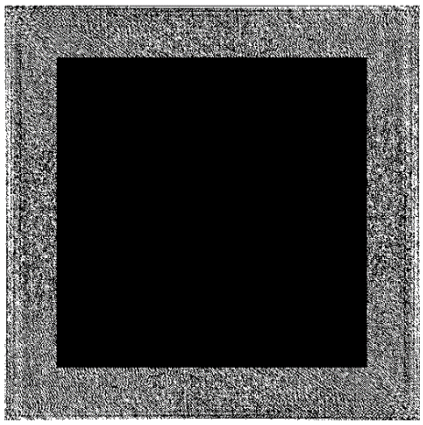
در این بخش تاثیر هر فیلتر را در حوزه فرکانسی و در حوزه مکانی بررسی خواهیم کرد.

ابتدا فیلتر a را اعمال میکنیم.

T=1/4



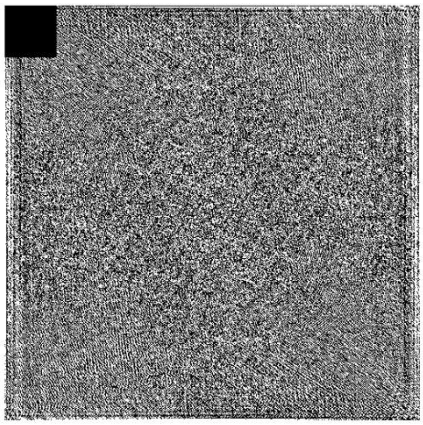
T = 1/8

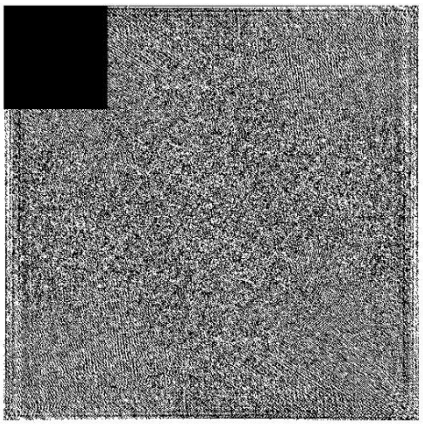


T= 1/8

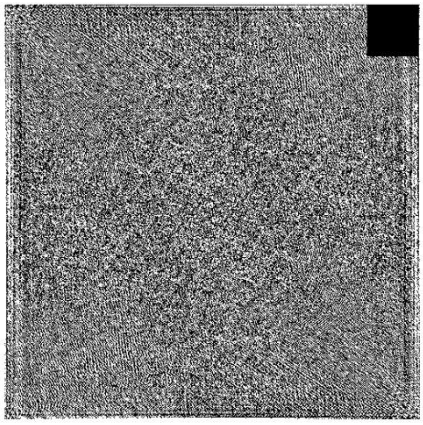
b.1

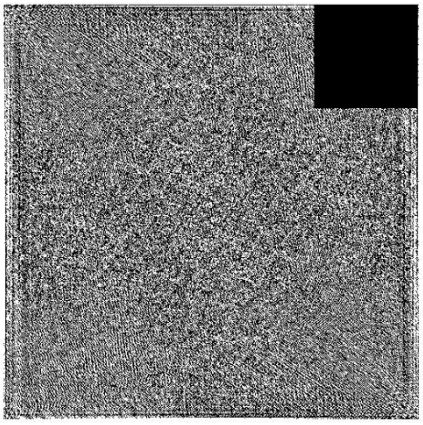
T = 1/4

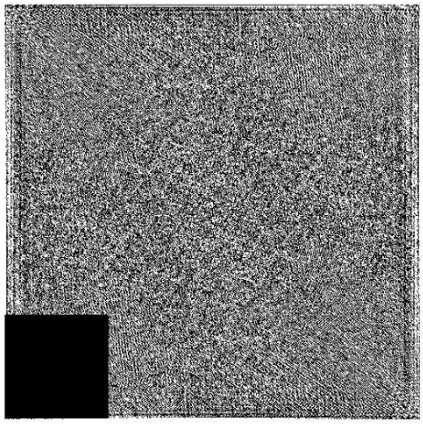


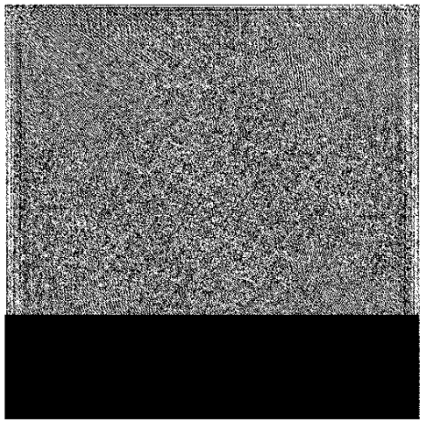


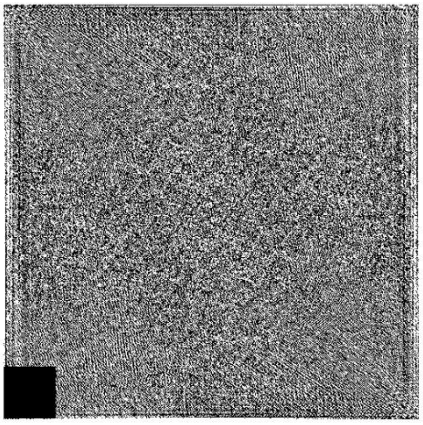
B2

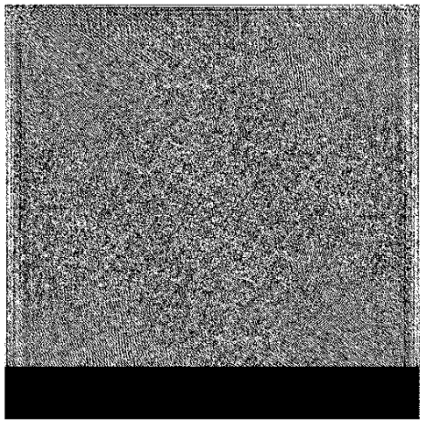












4-پیوست

۴.۱.۱

```
filter_a = ([1,2,1;2,4,2;1,2,1])/16;
filter_b = [-1,-1,-1;-1,8,-1;-1,-1,-1];
filter_c = [0,-1,0;-1,5,-1;0,-1,0];

filter_a_Fourier = fft2(filter_a);
filter_b_Fourier = fft2(filter_b);
filter_c_Fourier = fft2(filter_c);

filter_a_magnitude = abs(fftshift(filter_a_Fourier));
filter_b_magnitude = abs(fftshift(filter_b_Fourier));
filter_c_magnitude = abs(fftshift(filter_c_Fourier));

disp('filter a magnitude');
disp(filter_a_magnitude);
imshow(filter_a_magnitude,[]);

disp('filter b magnitude');
disp(filter_b_magnitude);

disp('filter c magnitude');
disp(filter_c_magnitude);

filter_a_on_barbara = uint8(conv2(single(grayImg),filter_a,'same'));
filter_b_on_barbara = uint8(conv2(single(grayImg),filter_b,'same'));
filter_c_on_barbara = uint8(conv2(single(grayImg),filter_c,'same'));
```

۴.۱.۲

```
lenaGray = rgb2gray(lenaImg);
baboonGray = rgb2gray(baboonImg);
f16Gray = rgb2gray(f16Img);
% dft magnitude
lenaDft = abs(fft2(lenaGray));
baboonDft = abs(fft2(baboonGray));
f16Dft = abs(fft2(f16Gray));

%dft shift magnitude
lenaShiftDft = abs(fftshift(fft2(lenaGray)));
baboonShiftDft = abs(fftshift(fft2(baboonGray)));
f16ShiftDft = abs(fftshift(fft2(f16Gray)));

% dft log shift magnitude
lenaLogShiftDft = log(lenaShiftDft);
baboonLogShiftDft = log(baboonShiftDft);
f16LogShiftDft = log(f16ShiftDft);

% dft log
lenaLogDft = log(lenaDft);
baboonLogDft = log(baboonDft);
f16LogDft = log(f16Dft);
```

۴.۲.۲

```
lenaImg = imread('.\4\Lena.bmp');
lenaGray = rgb2gray(lenaImg);
lenaDft = fft2(lenaGray);
[M,N] = size(lenaGray);

a_filter = ones(M,N);
for k = 1:M
    for l=1:N
        if( k>T*N && l>T*N && k<(1-T)*N && l<(1-T)*N)
            a_filter(k,l) = 0;
        end
    end
end
result_a = lenaDft.*a_filter;
imshow(uint8(result_a),[]);
result_a = ifft2(result_a);
result_a = real(uint8(result_a));
% imshow(result_a,[]);

b1_filter = ones(M,N);
for k = 1:M
    for l=1:N
        if( k <= T*N && l <= T*N && k >= 0 && l >= 0)
            b1_filter(k,l) = 0;
        end
    end
end
result_b1 = lenaDft.*b1_filter;
% imshow(uint8(result_b1),[]);
result_b1 = ifft2(result_b1);
result_b1 = real(uint8(result_b1));
% imshow(result_b1,[]);

b2_filter = ones(M,N);
for k = 1:M
    for l=1:N
        if( k <= T*N && k >= 0 && l >= (1-T)*N && l <= N-1)
            b2_filter(k,l) = 0;
        end
    end
end
result_b2 = lenaDft.*b2_filter;
% imshow(uint8(result_b2),[]);
result_b2 = ifft2(result_b2);
result_b2 = real(uint8(result_b2));
% imshow(result_b2,[]);
```

```
b3_filter = ones(M,N);
for k = 1:M
    for l=1:N
        if( k <= N-1 && k >= (1-T)*N && l >= 0 && l <= T*N)
            b3_filter(k,l) = 0;
        end
    end
end
result_b3 = lenaDft.*b3_filter;
% imshow(uint8(result_b3),[]);
result_b3 = ifft2(result_b3);
result_b3 = real(uint8(result_b3));
% imshow(result_b3,[]);
```

```
b4_filter = ones(M,N);
for k = 1:M
    for l=1:N
        if( k >= (1-T)*N && l <= N-1 )
            b4_filter(k,l) = 0;
        end
    end
end
result_b4 = lenaDft.*b4_filter;
% imshow(uint8(result_b4),[]);
result_b4 = ifft2(result_b4);
result_b4 = real(uint8(result_b4));
% imshow(result_b4,[]);
```