第一题

给定图像‘barb.png’，利用一阶Butterworth低通滤波器进行频域滤波，当D0 = 10,20,40,80时，给出相应滤波图像，并分别以频域和空域的观点解释有关滤波结果。

算法描述和文档：

总流程：

图像乘以(-1)^(x+y) 中心变换

图像傅里叶变换

生成Butterworth低通滤波器

低通滤波

DFT反变换后取实部

图像乘以(-1)^(x+y) 反中心变换

具体代码和解释：

%读入图像

src=imread('barb.png');

%获得图像size

[m,n]=size(src);

src=double(src);

%这里我使用meshgrid生成网格矩阵，然后用得到的网格矩阵的值来当做中心变换的(-1)^(x+y)的x和y的值。

%这里要注意，第一个参数是1:n而不是1:m，因为第一个参数是针对横坐标，就是得到的矩阵的列数。

[N,M]=meshgrid(1:n,1:m);

%图像乘以(-1)^(x+y) 中心变换

src\_trans=src.\*(-1).^(M+N);

% fft2 傅里叶变换

FF = fft2(src\_trans);

P=m/2;

Q=n/2;

D=[10,20,40,80];

rank=1;

G=zeros(m,n);

Butterworth\_filter=zeros(m,n);

figure;

for i=1:length(D)

G = FF;

%生成Butterworth低通滤波器

Butterworth\_filter = 1./ ( 1 + ( sqrt((M-P).^2 + (N-Q).^2) ./ D(i) ) .^(2\*rank) ) ;

%低通滤波

G = G .\* Butterworth\_filter;

%DFT反变换后取实部

G=ifft2(G); G=real(G);

%图像乘以(-1)^(x+y) 反中心变换

G=G.\*(-1).^(M+N);

subplot(2,2,i);

imshow(G,[]);title( sprintf('D0=%d',D(i)) );

end

有关图像：

D0=10



D0=20



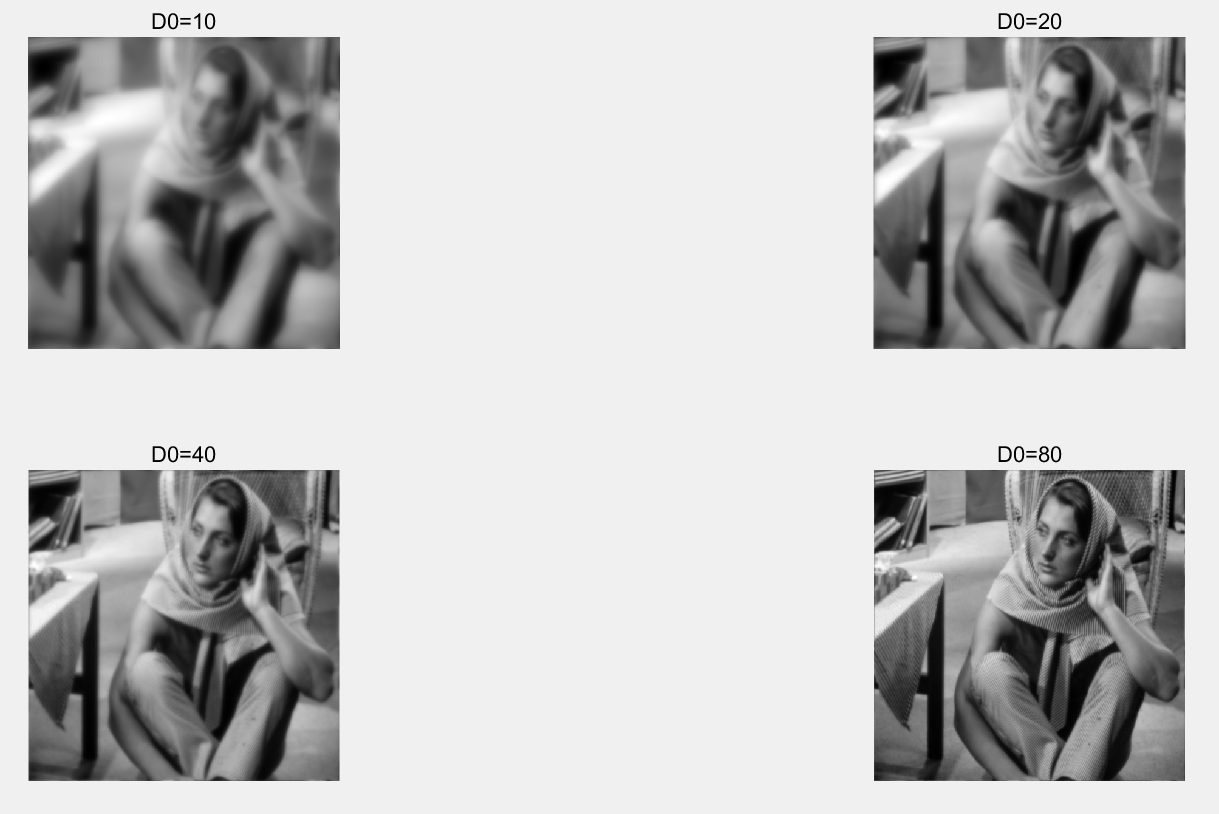
D0=40



D0=80



四张图放在一起比较：



分别以频域和空域的观点解释有关滤波结果。

频域观点：

图像中，边缘和其他尖锐的灰度转变对其傅里叶变换的高频内容有贡献。

低通滤波器，衰减高频成分，保留低频成分。使得图像平滑、模糊。

因为大于D0的频率衰减的较多，D0越小，较多的高频成分被滤除，则图像比较模糊。

所以D0=10处理后的图像，高频成分最少。D0=80处理后的图像，高频成分最多。

空域观点：

图像中，边缘和其他尖锐的灰度转变对其傅里叶变换的高频内容有贡献。

一个图像的高频成分越少，它的细节就越少，就越模糊。

D0=10处理后的图像，高频成分最少，则细节最少，最模糊。

D0=80处理后的图像，高频成分最多，则细节最多，最清晰。

第二题

采用同态滤波来增强图像‘office.jpg’细节。

算法描述和文档：

总流程：

原图像像素值+1后取对数。

傅里叶变换

生成同态滤波器

同态滤波

DFT反变换

先取指数，再取实部，最后-1，因为求对数的时候+1

具体代码和解释：

src\_rgb=imread('office.jpg');

%rgb图像转成gray

src=double(rgb2gray(src\_rgb));

%取对数

src=log(src+1);

[m,n]=size(src);

[N,M]=meshgrid(1:n,1:m);

% fft2 傅里叶变换

FF = fft2(src);

P=m/2;

Q=n/2;

D=[100,500,1000,2000];

rh=2.0; rl=0.25; c=1.0;

rank=1;

G=zeros(m,n);

Homomorphic\_filter=zeros(m,n);

figure;

for i=1:length(D)

G = FF;

%生成同态滤波器

Homomorphic\_filter = (rh-rl) .\* ( 1 - exp ( (-c) .\* ( ( (M-P).^2 + (N-Q).^2 ) ./ D(i).^2 ) ) ) + rl;

%同态滤波

G = G .\* Homomorphic\_filter;

%DFT反变换

G=ifft2(G);

%先取指数，再取实部，最后-1，因为求对数的时候+1

G=real(exp(G))-1;

%将图像像素值映射到[0,255]

MAX=max(max(G));

MIN=min(min(G));

G=255.\*(G-MIN) ./ (MAX-MIN);

G=uint8(G);

subplot(2,2,i);

%imshow(G,[]);title( sprintf('D0=%d',D(i)) );

imshow(G);title( sprintf('D0=%d',D(i)) );

end

尝试不同的D0，发现D0大于1000后，再增加D0的大小，结果没有明显变化。

所以认为D0=1000可以得到最好的结果。

有关图像：





D0=1000的时候的图像：



如将滤波器替换为一阶Butterworth高通滤波器，比较滤波结果。

算法描述和具体解释同题目一，只是滤波器函数改变：

感觉Butterworth高通滤波器不如同态滤波器效果好。

原图像较黑的部分（左下角），很容易滤波后就看不清楚。

