

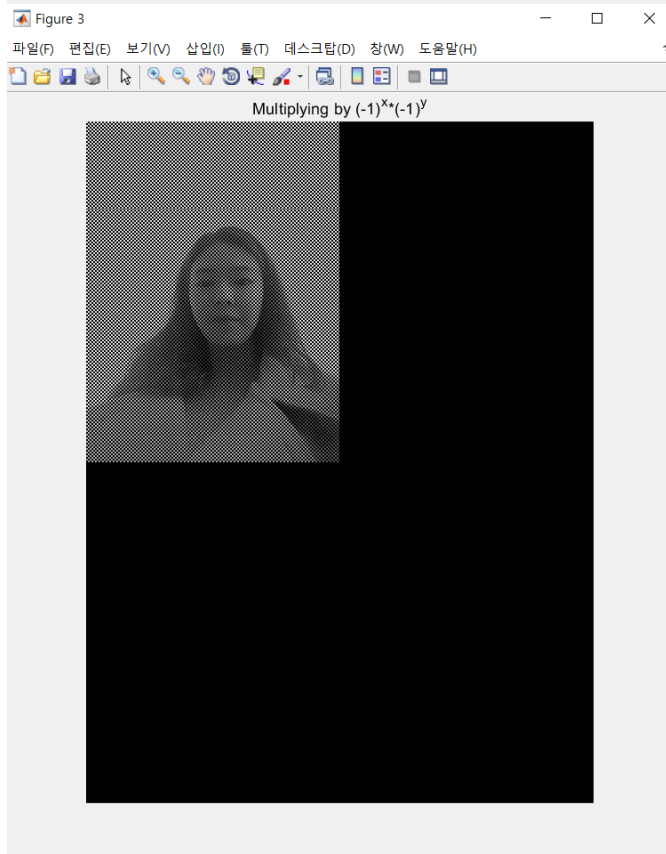
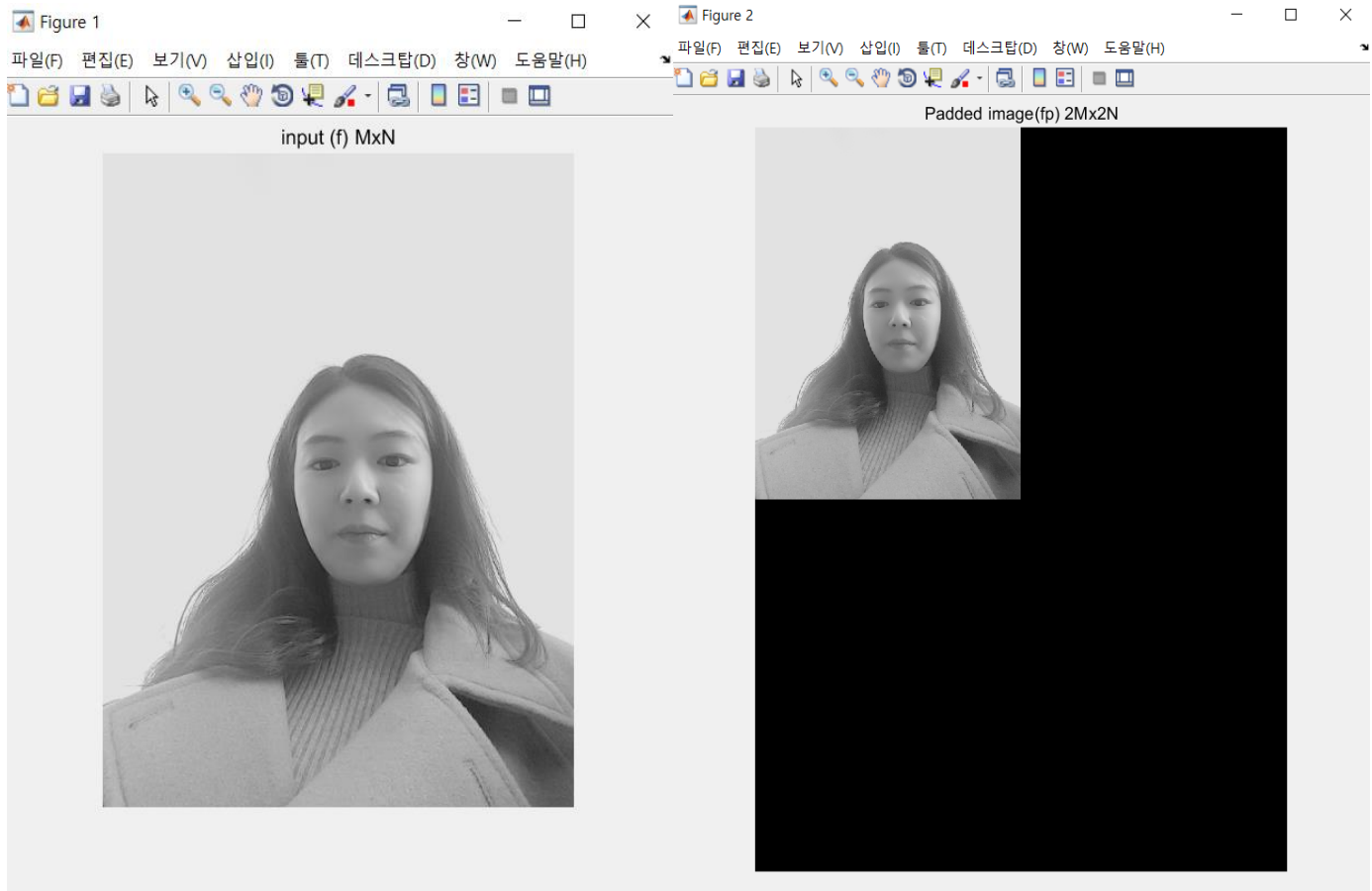
디지털 영상 처리 LAB 7

Filtering in Freq. domain

21611591 김난희

-source code (matlab)

```
1 % 1. input (f) MxN
2 f=imread('myimg.png');
3 f=rgb2gray(f);
4 figure(1); imshow(f); title('input (f) MxN');
5
6 % 2. Padded image(fp) 2Mx2N
7 [M,N]=size(f);
8 Padded_img = zeros(2*M,2*N); % 초기화
9 for i=1:M
10     for j=1:N
11         Padded_img(i,j) = f(i,j); % 할당된 곳 제외하고는 zero padding
12     end
13 end
14 figure(2); imshow(uint8(Padded_img)); title('Padded image(fp) 2Mx2N');
15
16 % 3. Multiplying by  $(-1)^{x+y}$ 
17 center=zeros(2*M,2*N);
18 for i=1:2*M
19     for j=1:2*N
20         center(i,j)=Padded_img(i,j).*((-1).^(i+j)); % 저주파 대역으로 나타냄
21     end
22 end
23 figure(3); imshow(uint8(center)); title('Multiplying by  $(-1)^x*(-1)^y$ ');
```



← 확대해서 보면 또 다른 그림이다.

아래 4번, 문제에서 3번을 넣어주라 하여 넣어주었지만, 왜 그런지 이해가 되지 않는다.

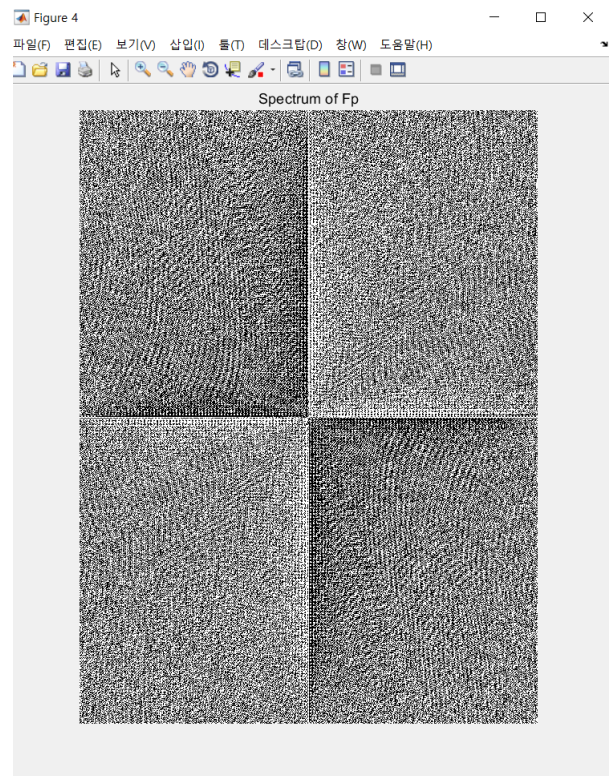
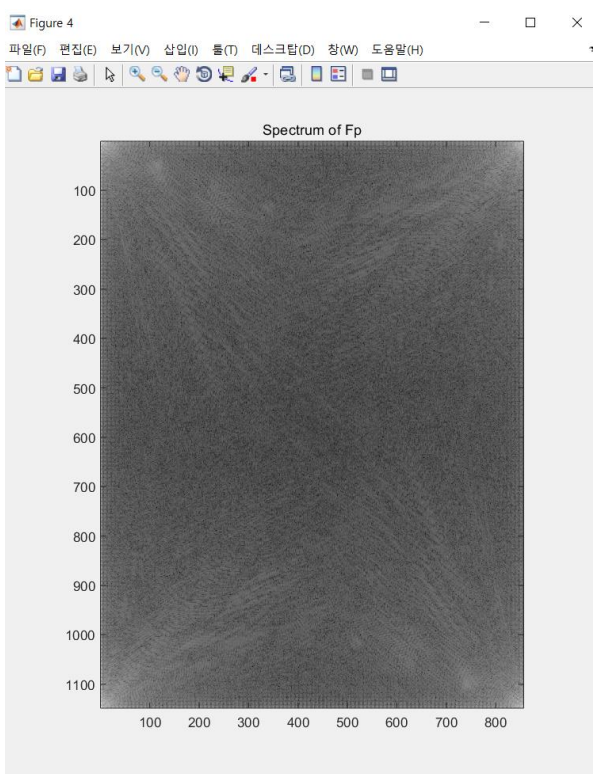
3번은 zero padding을 했으므로, 제대로 된 결과가 나오지 않을 것이라 생각이 들었다.

```
25 % 4. Fp : compute DFT
26 - F=fft2(center); Fcenter=fftshift(F); %원점을 중간으로 옮기는 변환 함수
27 - magF=abs(Fcenter); % real와 imaginaly의 크기
28 - figure(4); imagesc(log(1+magF)); title('Spectrum of Fp'); colormap gray; axis image;
```

여기서 imshow(F); 를 해보면,

원하는 결과가 나오지 않는다. (바로 아래 그림)

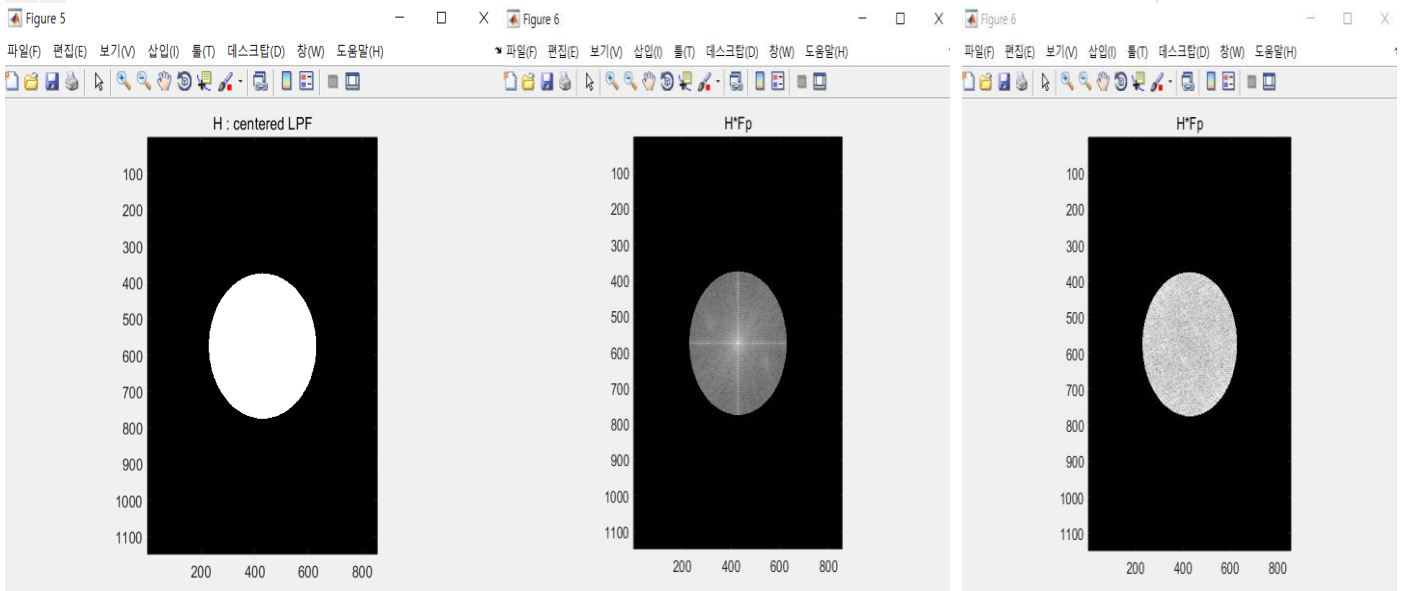
FIGURE 6의 결과를 보면 잘 나온 것이 이해되지 않는다



```

30 % 5. H : centered LPF
31 [M,N]=size(F);
32 u=0:(M-1); v=0:(N-1); % 벡터 생성
33 u=u-M/2; v=v-N/2; % 중심 맞춰줌
34 [V, U]=meshgrid(v, u);
35 D=sqrt(U.^2+V.^2); % 모든 픽셀마다 중심으로 부터 거리 계산하는 D
36 P=200; % Cut-off frequency
37 H = double(D<=P); % 만족하면 1 아니면 0 나오는 logical data를 double data로 변환
38 figure(5); imagesc(H); title('H : centered LPF'); colormap gray; axis image;
39
40 % 6. H*Fp
41 G =H.*F;
42 magG=abs(G);
43 figure(6); imagesc(log(1+magG)); title('H*Fp'); colormap gray; axis image;

```



여기서 아래와 같이 Fcenter를 곱해줄 경우에는 3번째와 같은 결과가 나왔다.

```

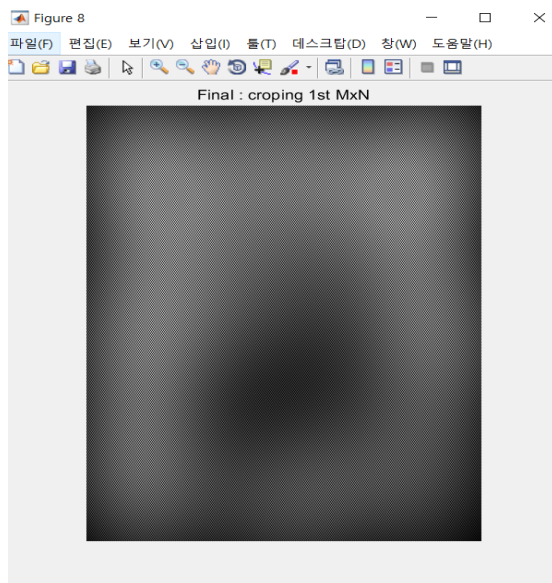
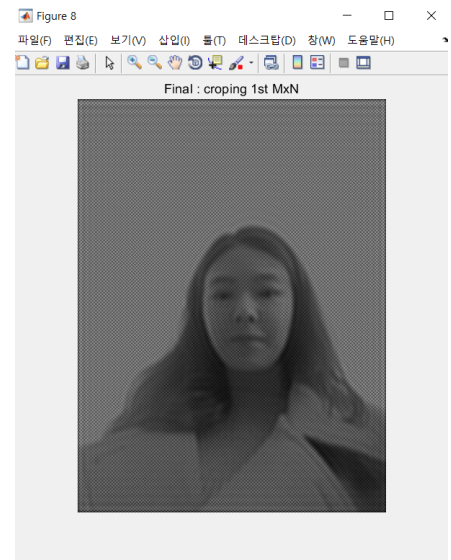
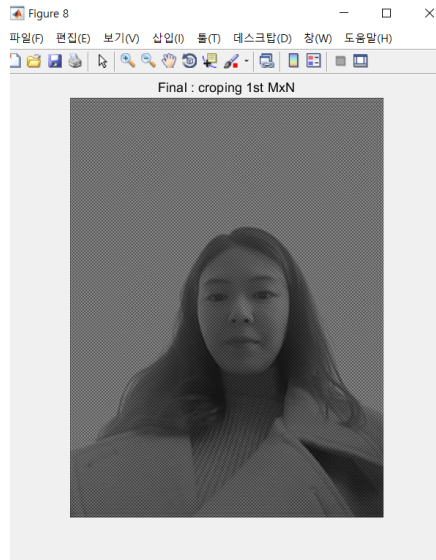
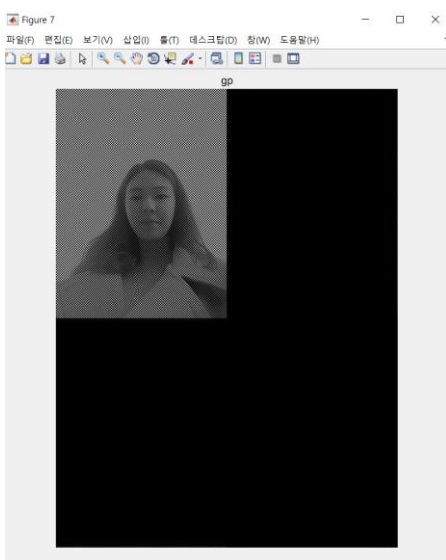
40 % 6. H*Fp
41 G =H.*Fcenter;
42 magG=abs(G);
43 figure(6); imagesc(log(1+magG)); title('H*Fp'); colormap gray; axis image;

```

```

45 % 7. Processed image
46 g=real(ifft2(G)).*((-1).^(i+j));
47 % g=real(ifft2(double(ifftshift(G))));
48 figure(7); imshow(uint8(g)); title('gp');
49
50 % 8. Obtain the final processed result
51 [M,N]=size(f);
52 result= zeros(M,N);
53 for i=1:M
54     for j=1:N
55         result(i,j)=g(i,j);
56     end
57 end
58 figure(8); imshow(uint8(result)); title('Final : cropping 1st MxN');

```

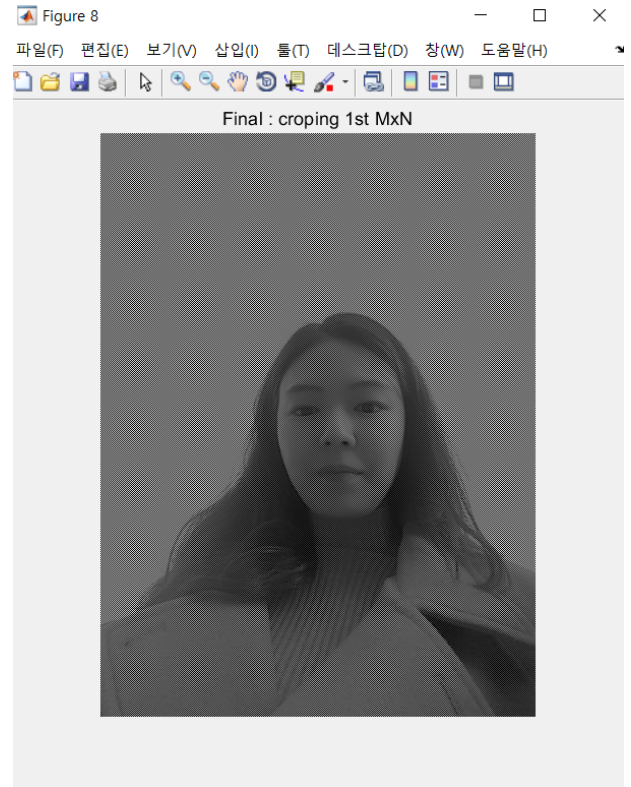
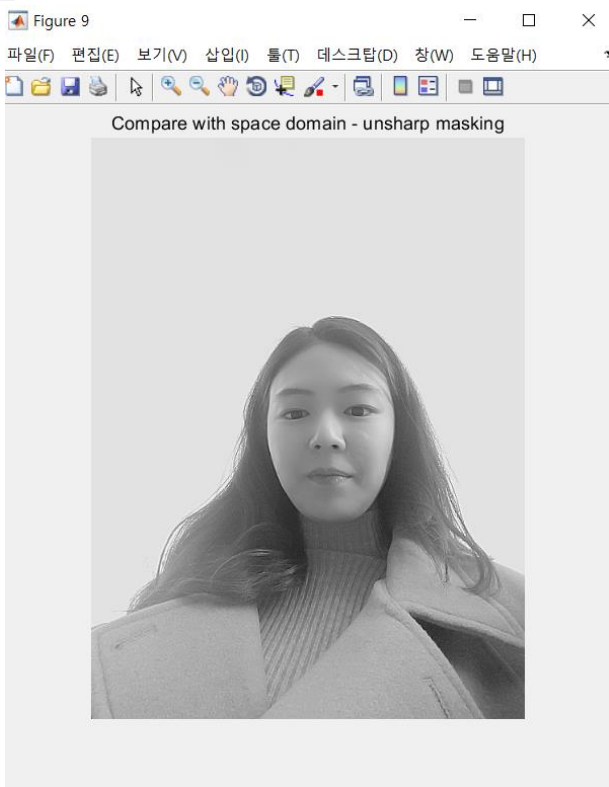


위의 왼쪽 결과 그림은 Cut-off frequency를 200을 적용한 것이고, 위의 오른쪽 결과 그림은 100을 적용한 것이다. 필터의 Cut-off frequency가 작을수록 결과가 더 잘 안 나온 것을 볼 수 있다.

왼쪽 그림은 Cut-off frequency를 5를 적용한 것이다. 거의 형태를 알아볼 수 없다.

마지막으로 이전에 배웠던 공간 상에서 필터링을 적용해 본다.

```
60 % 9. Compare - space domain filtering(Unsharp Masking)
61 w=fspecial('gaussian', [5 5],1);
62 lb=imfilter(f,w,'same','replicate');
63 ls=f+(f-lb); % k =1 : unsharp masking
64 figure(9); imshow(ls); title('Compare with space domain - unsharp masking');
```



왼쪽 사진은 공간 상 필터링을 한 것이고 오른쪽 사진은 주파수 상 필터링을 한 것이다. 오른쪽 사진에 적용된 cut-off frequency의 값은 1000으로 거의 없는 것을 적용하여도 저렇게 어둡게 나온다. 그 이유는 제로 패딩과 섞여서 그럴 것이라는 짐작뿐이다. 더 깊게 공부해야 할 것 같다. 주파수 상에서도 같은 결과가 나와야 한다고 생각했는데 왜 아닌지 의문이 든다.