

## 영상처리 과제 #6

마감기한: 11월 15일 일요일 23:59까지

학과	전자정보통신공학과	과목명	영상처리(001)
학번	18010697	이름	김해리

### # 1

#### MATLAB code

```
% 1. 'pelicans.tif' 영상에 대해 아래에 아래의 문제를 해결하세요.
img = imread("sample_images/pelicans.tif");

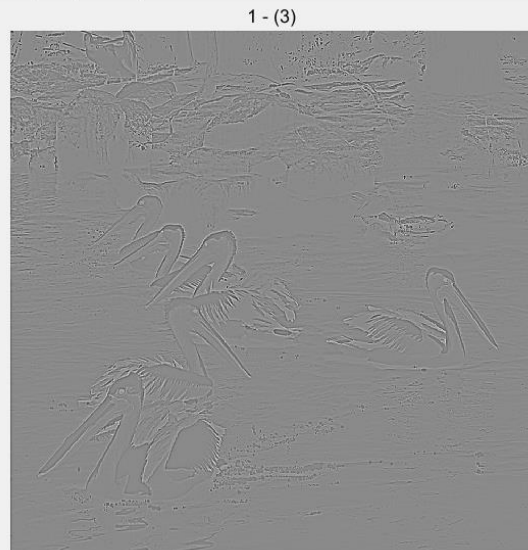
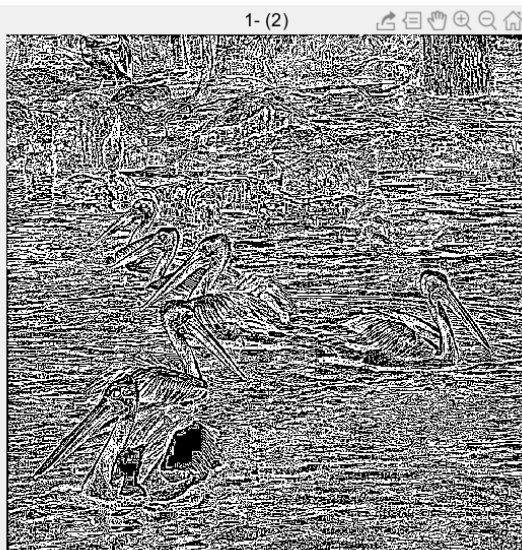
% (1) fspecial 함수를 사용하여 LoG 필터를 생성하여 pelicans 영상을 필터링하세요.
kernel = fspecial("log");
f_img = filter2(kernel, img);

% (2) (1)의 결과에서 0보다 작은 값은 0으로 255보다 큰 값은 255로 만들어 디스플레이하세요.
cut_img = f_img; % 필터링한 이미지 복사
cut_img(cut_img < 0) = 0; % 0보다 작은 값은 0으로 맞추기
cut_img(cut_img > 255) = 255; % 255보다 큰 값은 255로 맞추기
figure(1), imshow(cut_img); % 디스플레이
title("1- (2)");

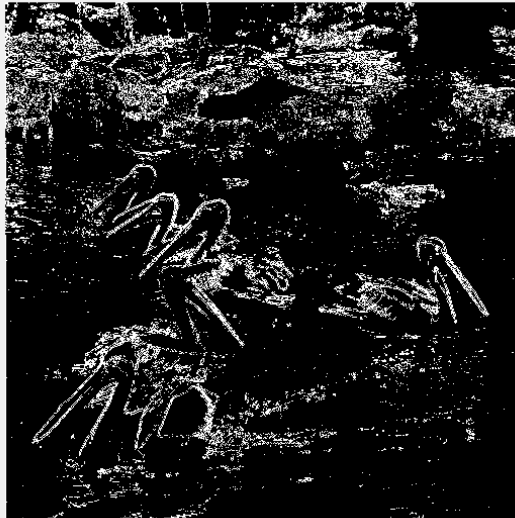
% (3) (1)의 결과에서 최소값을 0으로 최대값을 1로 스케일링 변환하여 디스플레이하세요.
scale_img = double(f_img); % 필터링한 이미지 복사
mx = max(max(scale_img)); % 최대값 구하기
mn = min(min(scale_img)); % 최소값 구하기
scale_img = (scale_img - mn) / (mx - mn); % 0-1 사이 값으로 맞춰주기 위해 최소값 빼주고 전체 스케일로 나눔
figure(2), imshow(scale_img);
title("1 - (3)");

% (4) (1)의 결과에서 절대값이 100보다 큰 값을 1로 작거나 같은 값을 0으로 변환 후 결과 디스플레이하세요.
bin_img = zeros(size(f_img));
bin_img(abs(f_img) > 100) = 1.;
bin_img(abs(f_img) <= 100) = 0.;
figure(3), imshow(bin_img);
title("1 - (4)");
```

#### display



1 - (4)



## # 2

### MATLAB code

```
1 — clear all
2 — clc
3
4 — % 2. 'caribou.tif' 영상에 대해 아래에 아래의 문제를 해결하세요.
5 — img = imread("sample_images/caribou.tif");
6
7 — % (1) fspecial 함수를 사용하여 크기가 5x5이고 표준편차가 2인 가우시안 필터를 만들고,
8 — % 이 필터를 사용하여 caribou 영상을 필터링하세요.
9 — kernel = fspecial("gaussian", [5 5], 2); % h = fspecial('gaussian',hsize,sigma)
10 — blur_img = filter2(kernel, img);
11
12 — % (2) (1)의 결과를 low-pass filtering 결과로 사용하여 caribou.tif 영상에 대한 언샤프 마스크를 수행하고, 결과
13 — % 를 디스플레이하세요. (scale은 1.5를 사용하고, 결과를 xu라고 할 때, uint8(xu*3)으로 디스플레이하세요.)
14 — unsharp_img1 = double(img) - blur_img/1.5;
15 — figure(1), imshow(uint8(unsharp_img1 * 3));
16 — title("2 - (2): unsharped with scale 1.5");
17
18 — % (3) 언샤프 마스크의 scale을 1.1과 10으로 변경해서 결과의 변화를 관찰하세요.
19 — % (강의자료 31쪽에서 xf/1.1과 xf/10로 설정하라는 의미입니다. 여기에서 산출된 결과를 각각 xu1, xu2라고
20 — % 할 때, 각각을 uint8(xu1*8), uint8(xu2)으로 디스플레이하세요.)
21 — unsharp_img2 = double(img) - blur_img/1.1;
22 — unsharp_img3 = double(img) - blur_img/10;
23 — figure(2), imshow([uint8(unsharp_img2 * 8), uint8(unsharp_img3)]);
24 — title("2 - (3): unsharped with scale 1.1(left), 10(right)|");
25
26 — % (4) Scale을 10, 1.5, 1.1으로 변경할 때의 변화를 에지 강조 측면에서 설명하세요.
```

display

2 - (2): unsharped with scale 1.5



2 - (3): unsharped with scale 1.1(left), 10(right)



### 2 - (4) Scale을 10, 1.5, 1.1으로 변경할 때의 변화를 에지 강조 측면에서 설명하세요

이 예제에서는 edge 검출을 위해 gaussian filter라는 low pass filter를 이용해 저주파 성분을 추출해 일정 scale로 나누어 전체 영상에서 빼 주었습니다.

scale 값이 작을수록 전체 영상에서 삭제할 저주파 성분이 더 큼니다.

## # 3

### MATLAB code

```
1 - clear all
2 - clc
3
4 - % 3. 'newborn.tif' 영상에 대해 아래에 아래의 문제를 해결하세요.
5 - img = imread("sample_images/newborn.tif");
6
7 - % (1) nlfiter 함수를 사용하여 영상에 5x5 최소값 필터를 적용하고 결과를 디스플레이 한 후, 영상의 전체적인
8 - % 밝기가 어떻게 변하는지와 그 이유를 설명하세요. (강의자료 36~38쪽)
9 - min_img = nlfiter(img, [5, 5], 'min(x(:))');
10 - figure(1), imshow(min_img);
11 - title("3 - (1) minimum");
```

% (2) ordfilt2 함수를 사용하여 영상에 5x5 최대값 필터를 적용하고 결과를 디스플레이 한 후, 영상의 전체적인 밝기가 어떻게 변하는지와 그 이유를 설명하세요. (강의자료 36~38쪽)

```
max_img = ordfilt2(img, 25, ones(5,5));  
figure(2), imshow(max_img);  
title("3 - (2) maximum");
```

% (3) ordfilt2 함수를 사용하여 영상에 5x5 Median filter 적용하세요. (강의자료 36~38쪽)

```
med_img = ordfilt2(img, 13, ones(5,5));  
figure(3), imshow(med_img);  
title("3 - (3) median");
```

#### display



#### (1)에 대한 설명

최소값 필터를 적용하면 필터 영역 안에서 가장 밝기 값이 낮은 값을 선택합니다. 따라서 전체적으로 영상의 밝기가 어두워지며, zero padding을 했기 때문에 가장자리에 검은 테두리가 생깁니다.

#### (2)에 대한 설명

최대값 필터를 적용하면 필터 영역 안에서 가장 밝기 값이 높은 값을 선택합니다. 따라서 전체적으로 영상의 밝기가 밝아집니다.

## # 4

### MATLAB code

% (1) 이상적인 저역통과 필터링을 주파수 영역에서 수행하세요. 디스플레이할 내용은 다음과 같습니다: 원본  
% 영상, 필터, 주파수 도메인에서의 필터링 결과, 영상 도메인에서의 필터링 결과 (강의자료 47~49쪽)

```
[x, y] = meshgrid(-128:127, -128:127);  
z = sqrt(x.^2 + y.^2);  
c = (z<15);  
cf = fftshift(fft2(img));  
cfl = cf.*c;  
cfl_i = ifft2(cfl);  
  
figure(1); imshow(img);  
figure(2); imshow(c);  
figure(3); fftshow(cfl, 'log');  
figure(4); fftshow(cfl_i, 'abs');
```

% (2) 이상적인 고역통과 필터링을 주파수 영역에서 수행하세요. 디스플레이할 내용은 다음과 같습니다: 원본  
% 영상, 필터, 주파수 도메인에서의 필터링 결과, 영상 도메인에서의 필터링 결과 (강의자료 50~52쪽)

```
[x, y] = meshgrid(-128:127, -128:127);  
z = sqrt(x.^2 + y.^2);  
c = (z>15);  
cf = fftshift(fft2(img));  
cfh = cf.*c;  
cfh_i = ifft2(cfh);  
  
figure(1); imshow(img);  
figure(2); imshow(c);  
figure(3); fftshow(cfh, 'log');  
figure(4); fftshow(cfh_i, 'abs');
```

% (3) 가우시안 저역통과 필터링을 주파수 영역에서 수행하세요. 디스플레이할 내용은 다음과 같습니다: 원본  
% 영상, 필터, 주파수 도메인에서의 필터링 결과, 영상 도메인에서의 필터링 결과 (강의자료 62~63쪽)

% - 가우시안 필터의 최대값을 1로 만들어 주기 위해 fspecial로 필터 g를 만들고  $g = g / \max(g(:))$ 를 해주세요

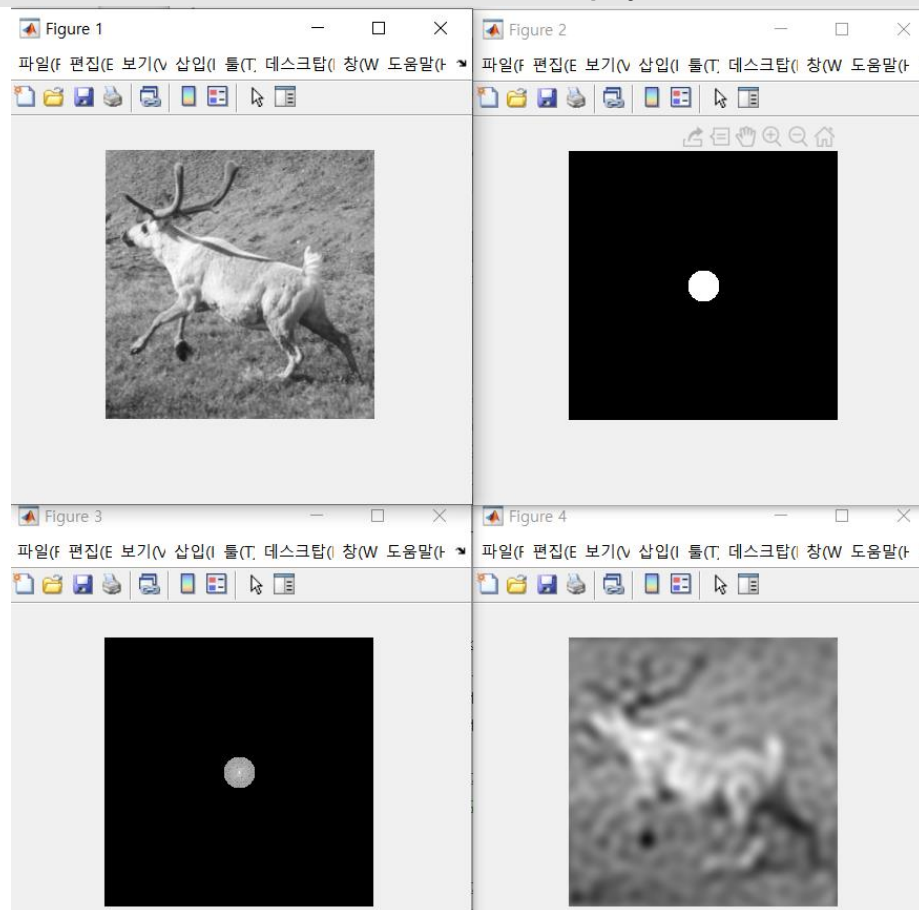
```
g = fspecial('gaussian', 256, 10);  
g = g/max(g(:));  
cf=fftshift(fft2(img));  
cfg=cf.*g;  
cfg_i = ifft2(cfg);  
  
figure(1); imshow(img);  
figure(2); imshow(g);  
figure(3); fftshow(cfg, 'log');  
figure(4); fftshow(cfg_i, 'abs');
```

% (4) 가우시안 고역통과 필터링을 주파수 영역에서 수행하세요. 디스플레이할 내용은 다음과 같습니다: 원본  
% 영상, 필터, 주파수 도메인에서의 필터링 결과, 영상 도메인에서의 필터링 결과 (강의자료 65쪽)  
% - 가우시안 필터의 최대값을 1로 만들어 주기 위해 fspecial로 필터 g를 만들고  $g = g / \max(g(:))$ 를 해주세요

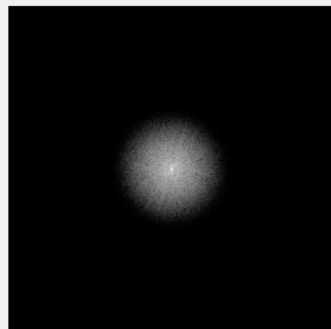
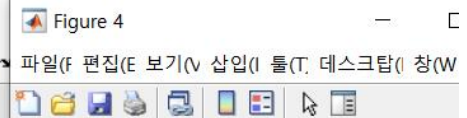
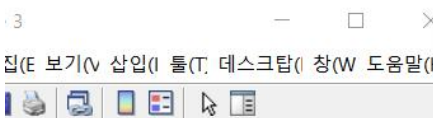
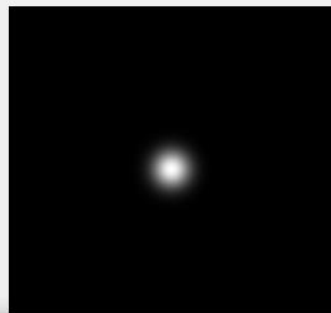
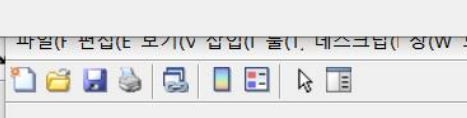
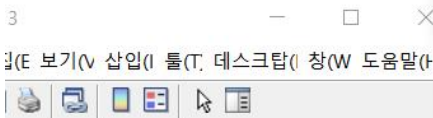
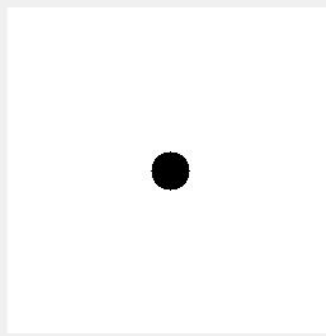
```
g = fspecial('gaussian', 256, 10);
g = 1 - g/max(g(:));
cf=fftshift(fft2(img));
cfg=cf.*g;
cfgi = ifft2(cfg);
```

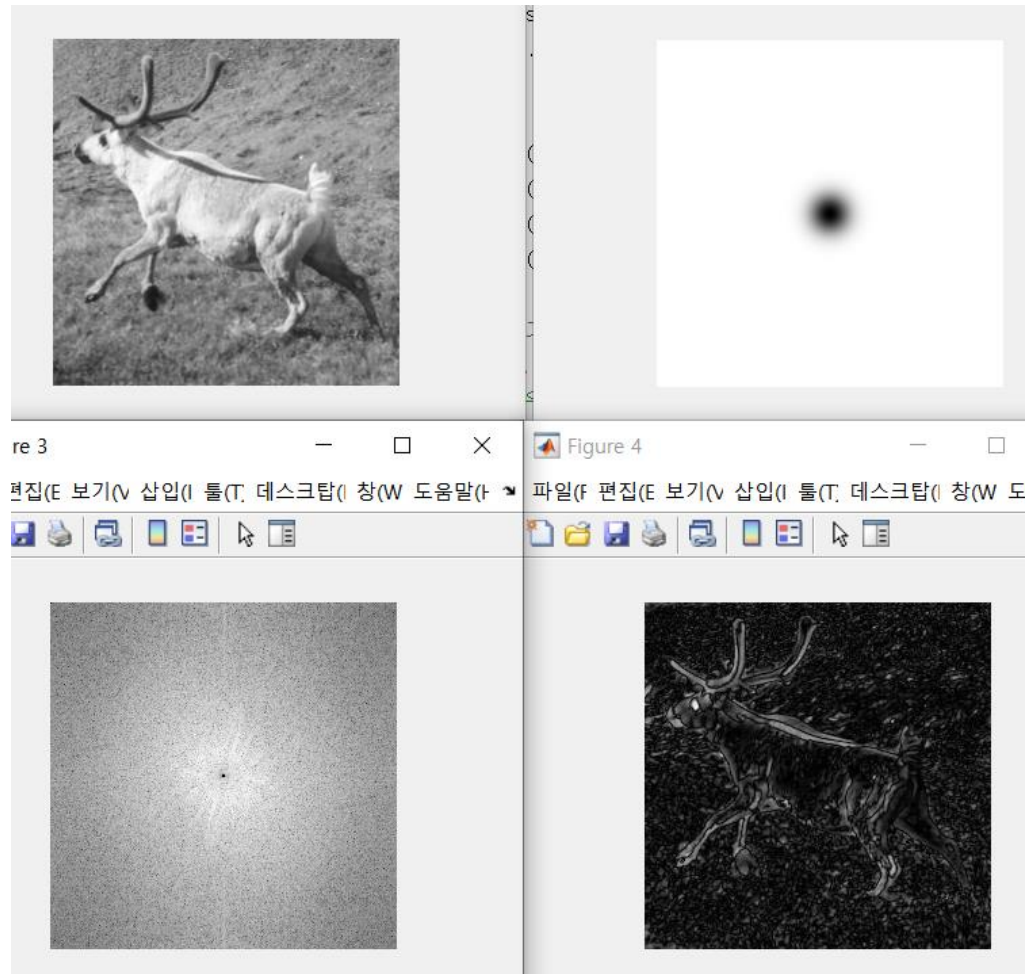
```
figure(1); imshow(img);
figure(2); imshow(g);
figure(3); fftshow(cfg, 'log');
figure(4); fftshow(cfgi, 'abs');
```

### display









**(5) 저역통과 필터링과 고역통과 필터링의 효과에 대해 각각 한 줄 이상으로 설명하세요.**

저역통과 필터링을 거칠 경우 DC 성분을 포함해 밝기 값의 변화가 크지 않은, 면에 해당하는 성분들만 남고 edge와 같이 밝기 값 변화가 심한 부분은 제거됩니다. 따라서 이미지가 전체적으로 뿌옇고 흐려 보입니다.

반면 고역통과 필터를 거쳤을 경우 우선 DC 성분이 제거되기 때문에 전체적으로 검은색을 띕니다. 또한 밝기 값 차이가 심한 면과 면의 경계 부분에 해당하는 성분들이 주로 남아, 이미지의 디테일을 뚜렷하게 확인할 수 있습니다.