

# 5 Neighborhood Processing 2

소프트웨어융합 김태민

## 연습문제 7

7. 다음에 있는 매개변수를 가진 가우시안 필터로 문제 6을 반복해서 풀기 위해서 “for loop”를 사용하는 M-파일을 작성하여 사용하라.

```

1 % apply_gaussian_filters.m
2
3 % mandrill.mat 파일을 불러옴
4 load('mandrill.mat');
5
6 % 그레이스케일 영상 준비
7 m = im2uint8(ind2gray(X, map));
8
9 % 가우시안 필터 매개변수 설정
10 filter_sizes = [3, 7, 11, 21];
11 standard_deviations = [0.5, 1, 1, 1];
12 threshold_values = [1, 3, 4, 5];
13
14 % 필터 적용 및 subplot에 출력
15 figure;
16 subplot(1, 5, 1);
17 imshow(m);
18 title('Original Image');
19
20 for i = 1:length(filter_sizes)
21     hsize = [filter_sizes(i), filter_sizes(i)];
22     sigma = threshold_values(i);
23     G = fspecial('gaussian', hsize, sigma);
24
25     % 가우시안 필터 적용
26     filtered_image = imfilter(m, G, 'same');
27
28     % 결과 subplot에 출력
29     subplot(1, 5, i + 1);
30     imshow(filtered_image);
31     title(['Size = ', num2str(hsize(1)), ', ', num2str(hsize(2)), ', ', \sigma = ', num2str(sigma)]);
32 end

```

```

1 % mandrill.mat 파일을 불러옴
2 load('mandrill.mat');
3
4 % 그레이스케일 영상 준비
5 m = im2uint8(ind2gray(X, map));
6
7 % 가우시안 필터 매개변수 설정
8 filter_sizes = [3, 7, 11, 21];
9 standard_deviations = [0.5, 1, 1, 1];
10 threshold_values = [1, 3, 4, 5];

```

이름을 잘못 써서 for loop로 바꿨습니다.

Figure 1 ✕

Original Image



Size = [3,3],  $\sigma = 1$



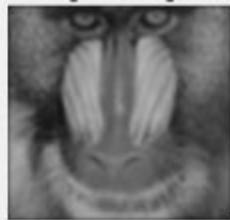
Size = [7,7],  $\sigma = 3$



Size = [11,11],  $\sigma = 5$



Size = [21,21],  $\sigma = 5$



명령 창

```
>>  
>> apply_gaussian_filters  
..
```

## 연습문제 8

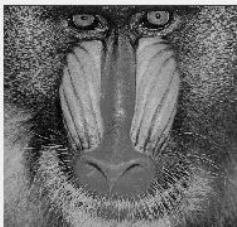
8. 평균 필터링과 가우시안 필터를 사용한 경우에 각각 결과 간의 차이를 볼 수 있는가?

```

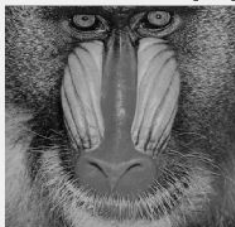
compare_filters.m
1      % mandrill.mat 파일을 불러옴
2      load('mandrill.mat');
3
4      % 그레이스케일 영상 준비
5      m = im2uint8(ind2gray(X, map));
6
7      % 필터 크기 설정
8      filter_sizes = [3, 7, 11, 21];
9
10     % figure 생성
11     figure;
12     subplot(2, 5, 1);
13     imshow(m);
14     title('원본 이미지');
15
16     % 평균 필터 적용 및 subplot에 출력
17     for i = 1:length(filter_sizes)
18         hsize = [filter_sizes(i), filter_sizes(i)];
19         avg_filter = fspecial('average', hsize);
20
21         % 평균 필터 적용
22         avg_filtered_image = imfilter(m, avg_filter, 'same');
23
24         % 결과 subplot에 출력
25         subplot(2, 5, i + 1);
26         imshow(avg_filtered_image);
27         title(['평균 필터: 크기 = ', num2str(hsize(1)), ', ', num2str(hsize(2)), '']);
28     end
29
30     % 원본 영상을 다시 첫 번째 위치에 표시
31     subplot(2, 5, 6);
32     imshow(m);
33     title('원본 이미지');
34
35     % 가우시안 필터 적용 및 subplot에 출력
36     for i = 1:length(filter_sizes)
37         hsize = [filter_sizes(i), filter_sizes(i)];
38         gauss_filter = fspecial('gaussian', hsize);
39
40         % 가우시안 필터 적용
41         gauss_filtered_image = imfilter(m, gauss_filter, 'same');
42
43         % 결과 subplot에 출력
44         subplot(2, 5, i + 6);
45         imshow(gauss_filtered_image);
46         title(['가우시안 필터: 크기 = ', num2str(hsize(1)), ', ', num2str(hsize(2)), '']);
47     end

```

원본 이미지



평균 필터: 크기 = [3,3]



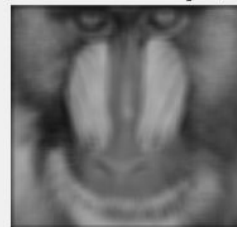
평균 필터: 크기 = [7,7]



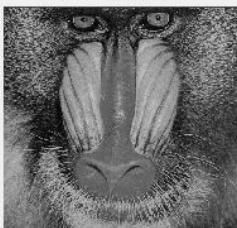
평균 필터: 크기 = [11,11]



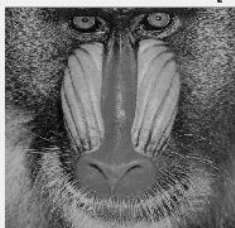
평균 필터: 크기 = [21,21]



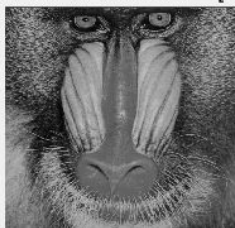
원본 이미지



가우시안 필터: 크기 = [3,3]



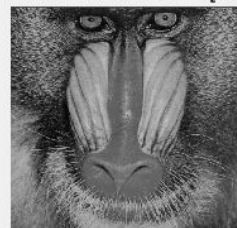
가우시안 필터: 크기 = [7,7]



가우시안 필터: 크기 = [11,11]



가우시안 필터: 크기 = [21,21]



명령 창

```
>> compare_filters
```

평균 필터에 비해 더 자연스럽게 처리가 되는거 같다.

결과값의 비교는 필터 크기가 커질수록 두드러진다.

## 연습문제 9

9. `fspecial` 함수의 도움 페이지를 읽고, cameraman 영상과 mandrill 영상에 대해서 몇 가지 다른 필터를 적용해 보라

`fspecial` 함수의 도움 페이지 : <https://kr.mathworks.com/help/images/ref/fspecial.html>



```

1 % cameraman.tif 파일을 불러옴
2 cameraman_image = imread('cameraman.tif');
3
4 % mandrill.mat 파일을 불러옴
5 load('mandrill.mat');
6 mandrill_image = im2uint8(ind2gray(X, map));
7
8 % 다양한 필터 설정
9 average_filter = fspecial('average', [5, 5]);
10 disk_filter = fspecial('disk', 10);
11 gaussian_filter = fspecial('gaussian', [5, 5]);
12 laplacian_filter = fspecial('laplacian', 0.2);
13 motion_filter = fspecial('motion', 20, 45);
14 sobel_filter = fspecial('sobel');
15
16 % cameraman 이미지에 필터 적용
17 cameraman_average = imfilter(cameraman_image, average_filter);
18 cameraman_disk = imfilter(cameraman_image, disk_filter);
19 cameraman_gaussian = imfilter(cameraman_image, gaussian_filter);
20 cameraman_laplacian = imfilter(cameraman_image, laplacian_filter);
21 cameraman_motion = imfilter(cameraman_image, motion_filter);
22 cameraman_sobel = imfilter(cameraman_image, sobel_filter);
23
24 % mandrill 이미지에 필터 적용
25 mandrill_average = imfilter(mandrill_image, average_filter);
26 mandrill_disk = imfilter(mandrill_image, disk_filter);
27 mandrill_gaussian = imfilter(mandrill_image, gaussian_filter);
28 mandrill_laplacian = imfilter(mandrill_image, laplacian_filter);
29 mandrill_motion = imfilter(mandrill_image, motion_filter);
30 mandrill_sobel = imfilter(mandrill_image, sobel_filter);
31

```

```

48 subplot(2, 7, 4);
49 imshow(cameraman_gaussian);
50 title('가우시안 필터');
51
52 subplot(2, 7, 5);
53 imshow(cameraman_laplacian);
54 title('라플라시안 필터');
55
56 subplot(2, 7, 6);
57 imshow(cameraman_motion);
58 title('모션 필터');
59
60 subplot(2, 7, 7);
61 imshow(cameraman_sobel);
62 title('소벨 필터');
63
64 % mandrill 결과 출력
65 subplot(2, 7, 8);
66 imshow(mandrill_image);
67 title('원본 mandrill');
68
69 subplot(2, 7, 9);
70 imshow(mandrill_average);
71 title('평균 필터');
72
73 subplot(2, 7, 10);
74 imshow(mandrill_disk);
75 title('원형 필터');
76
77 subplot(2, 7, 11);
78 imshow(mandrill_gaussian);

```

compare\_filters2의 M파일을 만들어 cameraman과 mandrill의 각종 필터 사용 결과 비교를 진행했다.

```
>> compare_filters2
```

원본 cameraman



평균 필터



원형 필터



가우시안 필터



라플라시안 필터



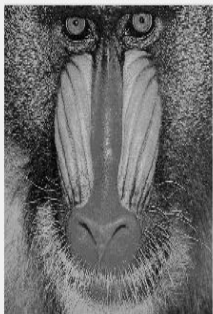
모션 필터



소벨 필터



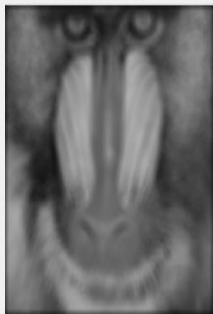
원본 mandrill



평균 필터



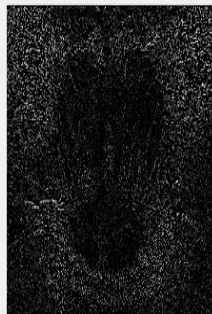
원형 필터



가우시안 필터



라플라시안 필터



모션 필터



소벨 필터



## 연습문제 10

10. cameraman과 mandrill 영상에 대해서 서로 다른 라플라시안 필터들을 적용해 보라. 어떤 필터가 가장 좋은 에지 영상을 생성하는가?

```
filters2.m × compare_filter3.m * × +
% cameraman.tif 파일을 불러옴
cameraman_image = imread('cameraman.tif');

% mandrill.mat 파일을 불러옴
load('mandrill.mat');
mandrill_image = im2uint8(ind2gray(X, map));

% 라플라시안 필터의 alpha 값 설정
alpha_values = [0, 0.2, 0.5, 0.7, 1.0];

% cameraman 이미지에 필터 적용 및 결과 저장
cameraman_results = cell(length(alpha_values), 1);
for i = 1:length(alpha_values)
    laplacian_filter = fspecial('laplacian', alpha_values(i));
    cameraman_results{i} = imfilter(cameraman_image, laplacian_filter, 'replicate');
end

% mandrill 이미지에 필터 적용 및 결과 저장
mandrill_results = cell(length(alpha_values), 1);
for i = 1:length(alpha_values)
    laplacian_filter = fspecial('laplacian', alpha_values(i));
    mandrill_results{i} = imfilter(mandrill_image, laplacian_filter, 'replicate');
end

% 결과 출력
figure;
```

```
% 결과 출력
```

```
figure;
```

```
% cameraman 결과 출력
```

```
subplot(2, length(alpha_values) + 1, 1);
```

```
imshow(cameraman_image);
```

```
title('원본 cameraman');
```

```
for i = 1:length(alpha_values)
```

```
    subplot(2, length(alpha_values) + 1, i + 1);
```

```
    imshow(cameraman_results{i});
```

```
    title(['alpha = ', num2str(alpha_values[i])]);
```

```
end
```

```
% mandrill 결과 출력
```

```
subplot(2, length(alpha_values) + 1, length(alpha_values) + 2);
```

```
imshow(mandrill_image);
```

```
title('원본 mandrill');
```

```
for i = 1:length(alpha_values)
```

```
    subplot(2, length(alpha_values) + 1, length(alpha_values) + 2 + i);
```

```
    imshow(mandrill_results{i});
```

```
    title(['alpha = ', num2str(alpha_values[i])]);
```

```
end
```



```
>> compare_filter3
```

원본 cameraman



alpha = 0



alpha = 0.2



alpha = 0.5



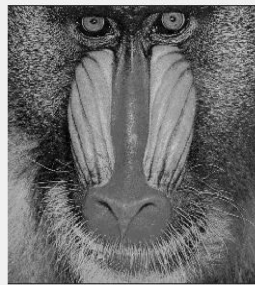
alpha = 0.7



alpha = 1



원본 mandrill



alpha = 0



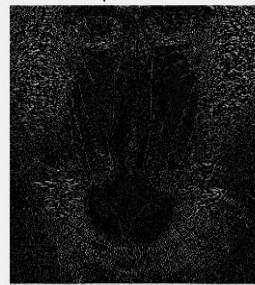
alpha = 0.2



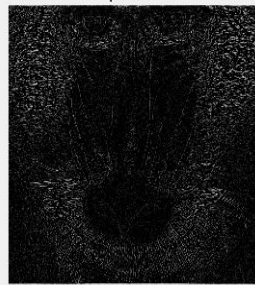
alpha = 0.5



alpha = 0.7



alpha = 1



- $\alpha = 0$ : 기본적인 4-이웃 라플라시안 필터로서, 가장 단순한 에지 검출을 수행함
- $\alpha = 0.2$ : 에지 검출이 좀 더 강하게 나타날 수 있으며, 잡음이 강조됨
- $\alpha = 0.5$ : 에지 검출이 더욱 뚜렷하게 나타나며, 세부 정보도 강조됨
- $\alpha = 0.7$ : 에지 검출이 매우 강하게 나타나지만, 잡음이 많이 강조됨
- $\alpha = 1.0$ : 가장 강한 에지 검출을 수행하며, 이미지의 대부분의 세부 정보와 잡음이 매우 뚜렷하게 나타남

## 연습문제 18

18. 블러링 효과를 역으로 사용하는데 언샤프 마스크를 사용할 수 있는가? 3\*3  
평균 필터링 처리 후에 언샤프 마스크 필터를 적용하고 그 결과를 설명해라

```
1 % cameraman.tif 파일을 불러옴
2 cameraman_image = imread('cameraman.tif');
3
4 % 3x3 평균 필터 생성
5 average_filter = fspecial('average', [3, 3]);
6
7 % 3x3 평균 필터를 사용하여 이미지 블러 처리
8 blurred_image = imfilter(cameraman_image, average_filter, 'replicate');
9
10 % 블러 이미지와 원본 이미지의 차이 (고주파 성분)
11 high_pass = double(cameraman_image) - double(blurred_image);
12
13 % 언샤프 마스크 필터 적용 (원본 이미지 + 고주파 성분)
14 sharp_image = double(cameraman_image) + high_pass;
15
16 % 결과를 uint8 형식으로 변환
17 sharp_image = uint8(sharp_image);
18
19 % 결과 출력
20 figure;
21
22 % 원본 이미지 출력
23 subplot(1, 3, 1);
24 imshow(cameraman_image);
25 title('원본 이미지');
26
27 % 블러 처리된 이미지 출력
28 subplot(1, 3, 2);
29 imshow(blurred_image);
30 title('3x3 평균 필터');
31
32 % 언샤프 마스크 결과 이미지 출력
33 subplot(1, 3, 3);
34 imshow(sharp_image);
35 title('언샤프 마스크 필터');
```



원본 이미지



3x3 평균 필터



연섀프 마스크 필터



원본 이미지에서 3\*3 평균 필터링을 하여 블러처리(고주파 성분(세부 정보)를 줄이고, 저주파 성분(흐림)을 강조)을 하고, 연섀프 마스크를 통해 블러 이미지를 원본 이미지에서 빼서 고주파 성분(세부 정보)를 얻고 이를 원본 이미지에 더해 이미지를 가장자리와 세부 정보를 강조, 시각적으로 더 선명하게 만든다.