Homework 3

ECE 253

Mingxuan Wang A53077257

November 6, 2015

- Problem 1 Median versus Mean Filtering

    1. It’s salt and pepper noise.
    2. The result plots are below:

Figure 1 Original image and noisy image

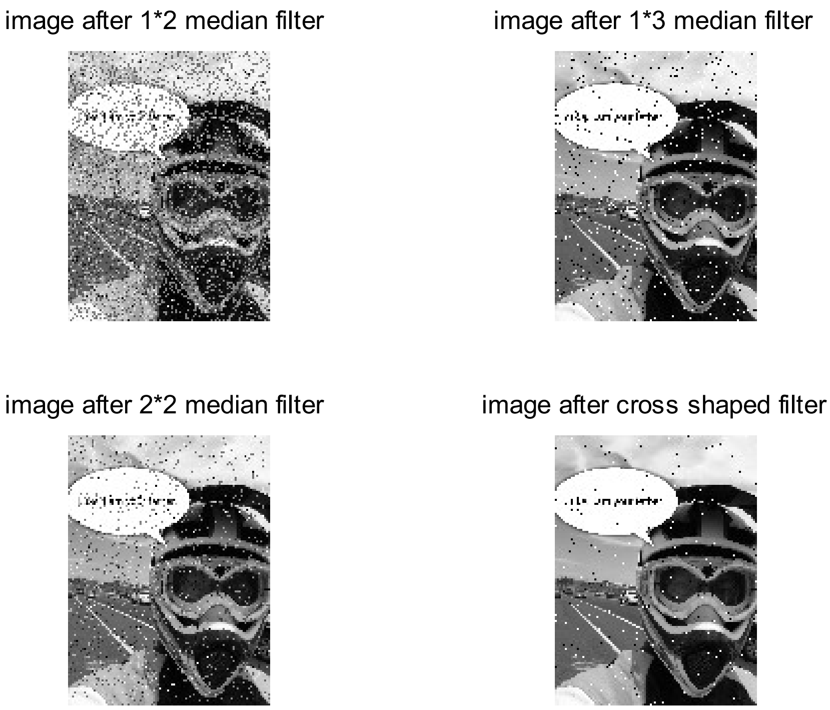

Figure 2 Images after median filter-1

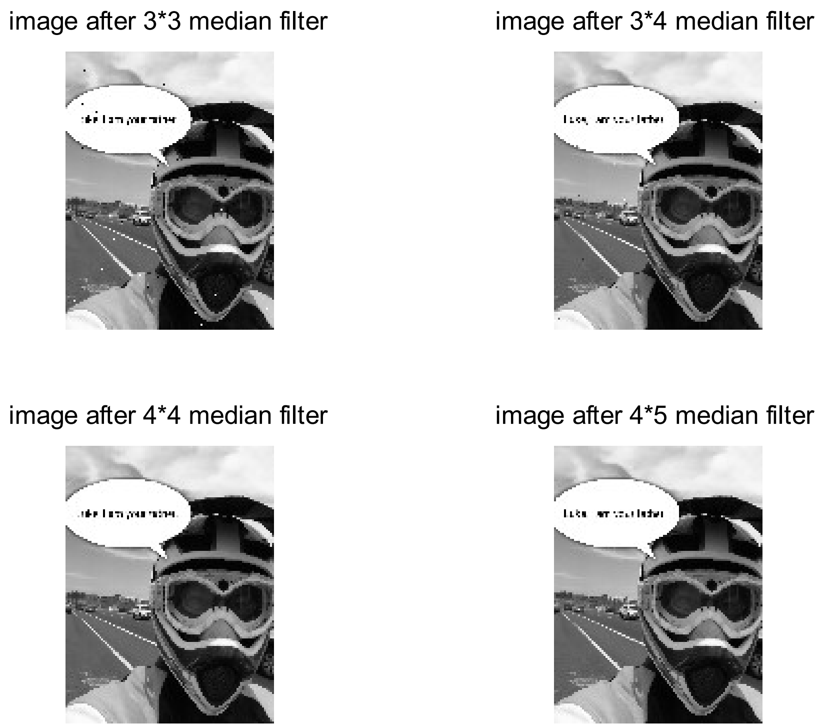

Figure 3 Images after median filter-2

Figure 4 Images after median filter-3

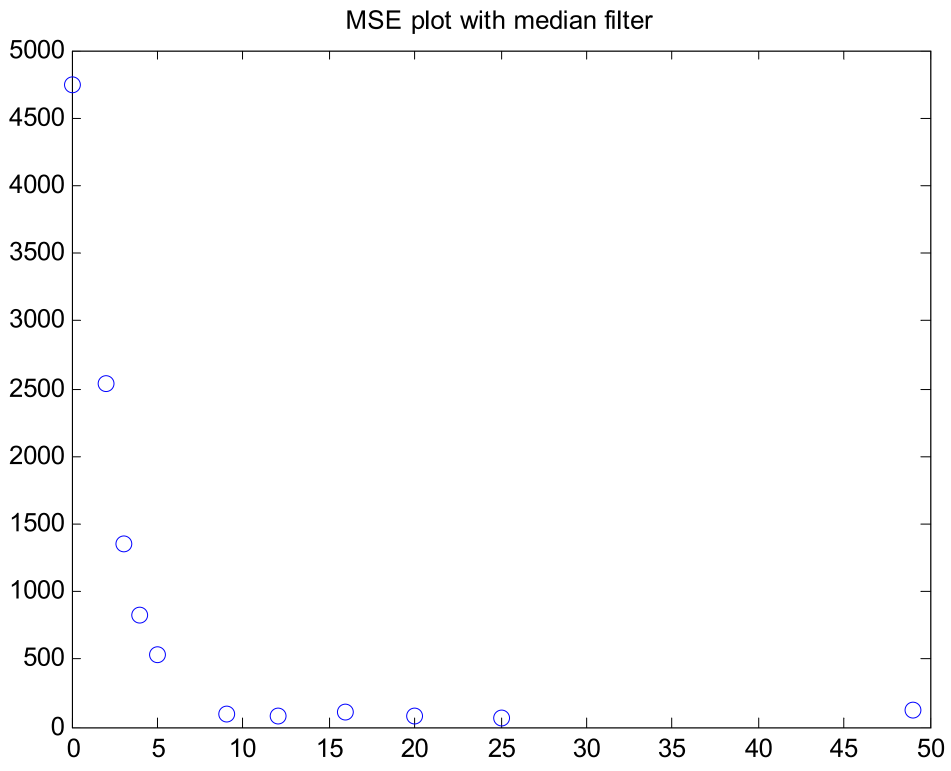

Figure 5 MSE plot with median filter

3. The result plots are below:

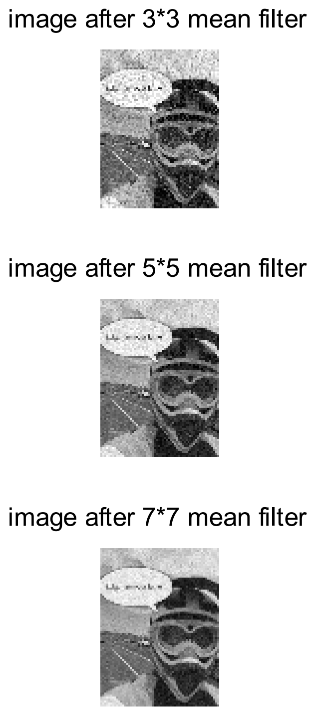

Figure 6 Images after mean filter

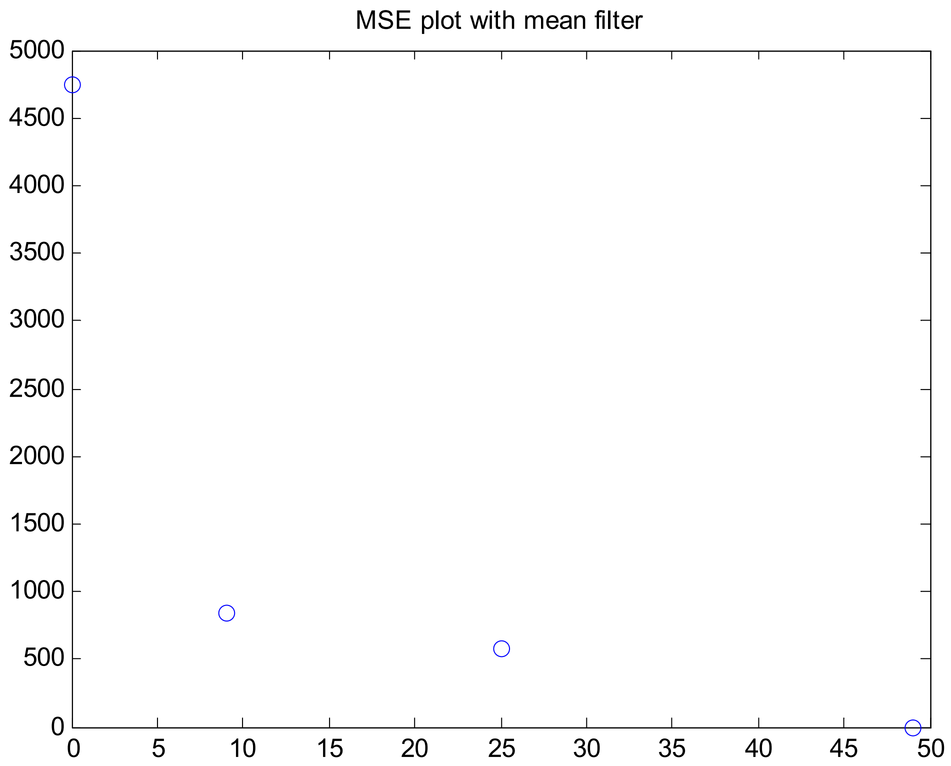

Figure 7 MSE plot with mean filter

Answers:

1. According to my observation, it appears to be better with median filter in small and regular size of the filter. This is because for salt and pepper noise, they are white and black pixels and for median filter these noise pixels won't affect the result too much while the for the mean filter, they just affect a lot when calculate the mean value. But when the size of filter is big enough, it may happen that the white and black pixels affect a little to the mean value. So the mean filter becomes better with larger size of filter.
2. According to my result, 5*5 kernel performs best for this noisy image. As we know for this image, the noise ratio is relatively high so that small filters like 1*3 will not be sufficient. These tiny filters will include a majority of pixels that are noisy within the kernel. But also larger filters like 7*7 is not needed for this image.

- Problem 2 Edge Sharpening

1. We have:
   HPF + LPF = IF …… (1)
   HBF = IF + aHPF …… (2)
   Substitute (1) to (2), we have:
   HBF = IF + a(IF - LPF) = (1+a)IF – aLPF
   That what we want.
2. MaskB is an identity filter. MaskC is maskB-maskA, so it is a high pass filter. With maskD = maskB + weight*maskC, we have maskD as a high boost filter. And maskD can sharpen the edge of the image with weight.

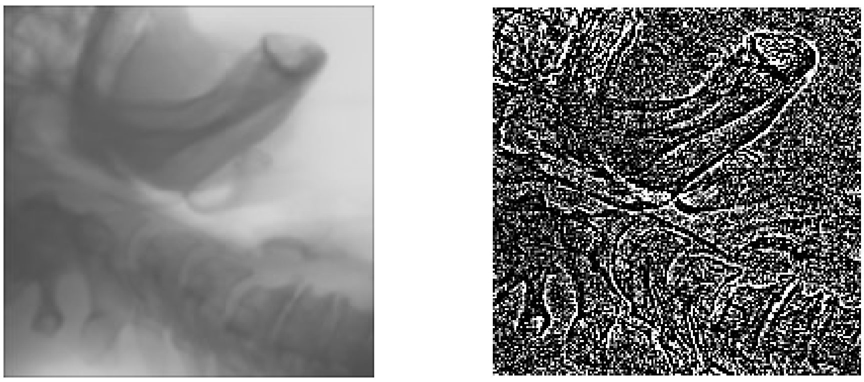

Figure 8 Low-pass(l) component and high-pass(r) component of an image.

3. The results:
   a) Different filter size:

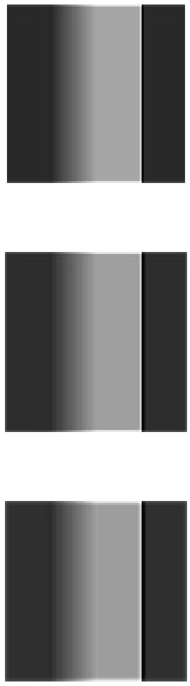

Figure 9 Result of different filter size. 3*3, 5*5, 7*7 from top to bottom

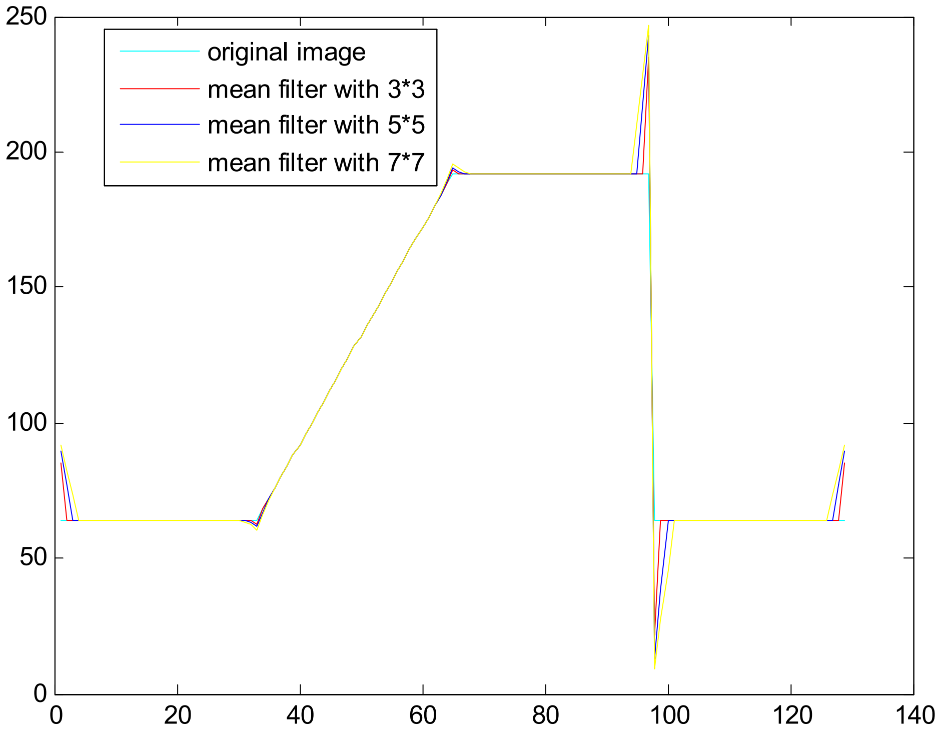

Figure 10 Slice of different filter size

For different size of the mean filter, the larger the size is, the wider the edge is. For the slice, the gap of the change becomes wider when the size getting larger.

b) Different types of filter:

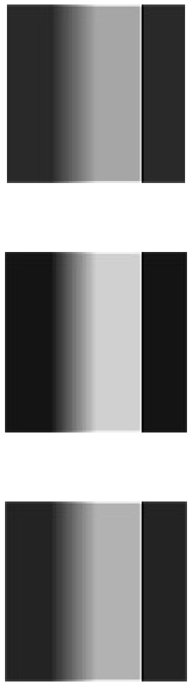

Figure 11 Result of different filter types. Mean, Gaussian, Weighted mean from top to bottom.

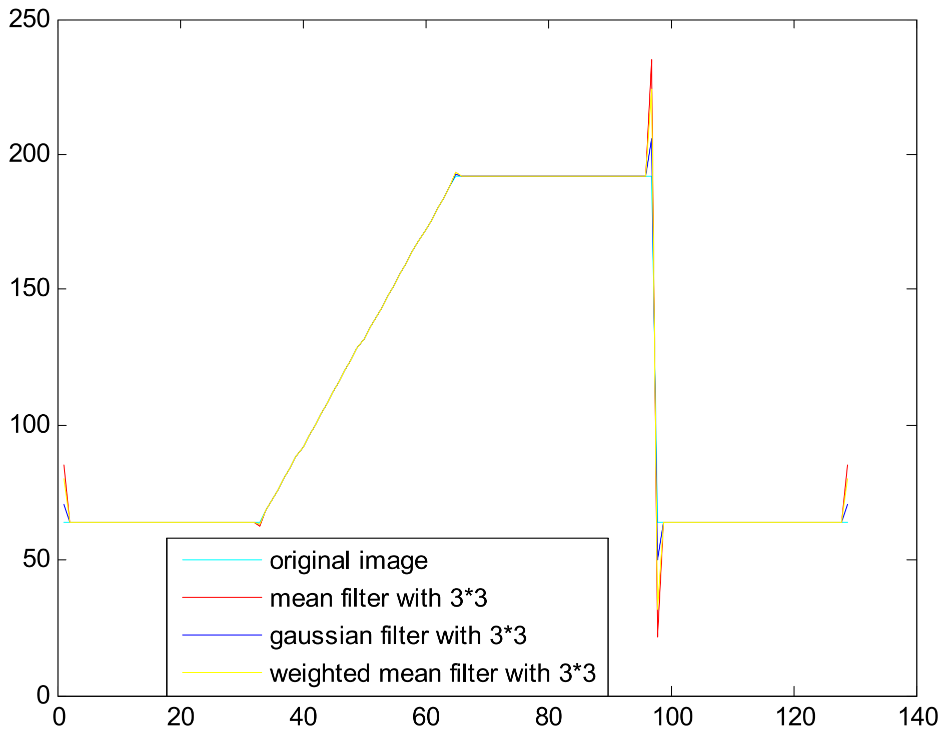

Figure 12 Slice of different filter types.

We can see that for Gaussian, the white part is brighter and the gap in the slice in smaller.

c) Different weight of high pass part:

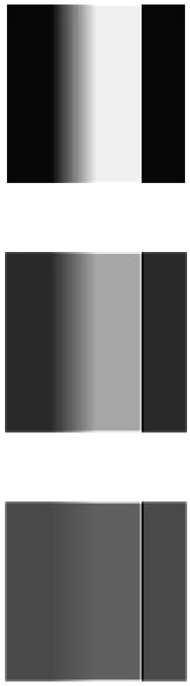

Figure 13 Results of different weight. 0.1, 1, 10 from top to bottom.

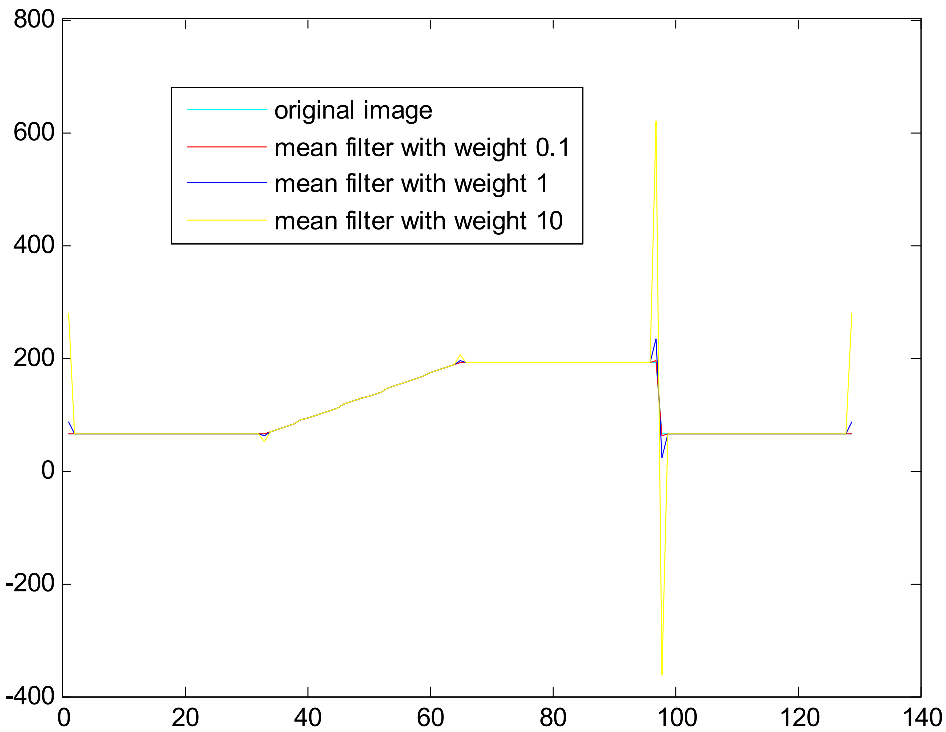

Figure 14 Slice of different weight.

We can see that the larger the weight of high-pass part, the sharper the edge is. And the gap in slices image when value changing is different. When the weight is larger, the gap is wider and sharper.

4. I do this by the mean filter with 5*5 size and 25 weight. I don’t do truncate or rescale. The result is:

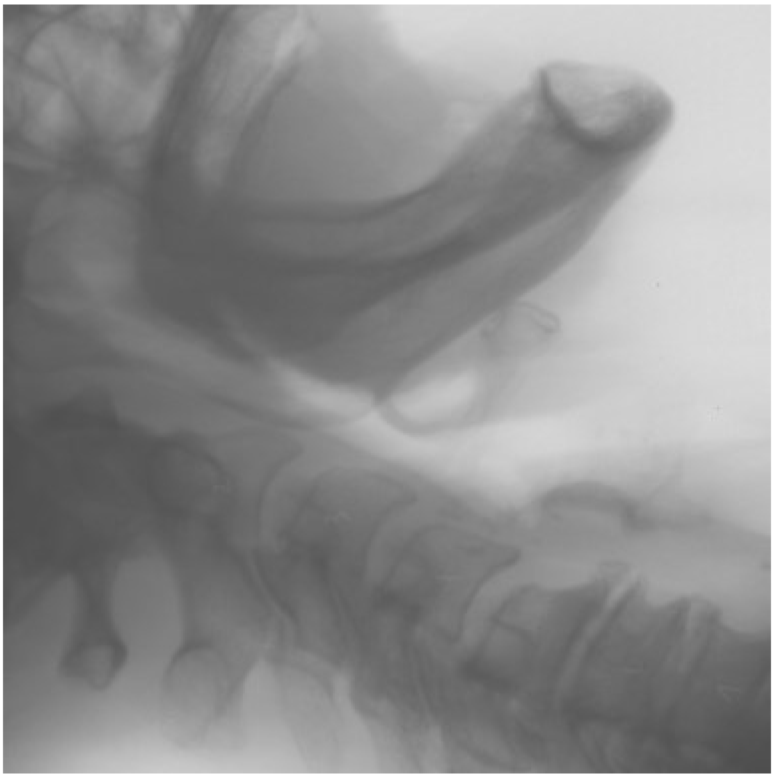

Figure 15 Original Image

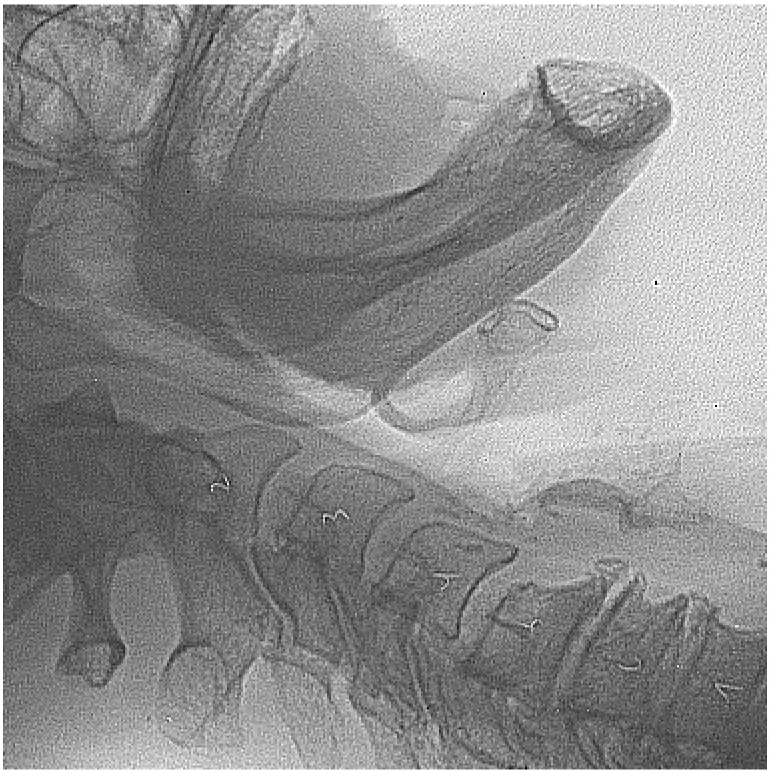

Figure 16 Enhanced Image

For the enhanced image, the edge is sharper, the bone is clear, the noise level is increased and the number on the bone is 234567.

- Problem 3 Binary Morphology
  1. Isolated Pixel Cleaning:

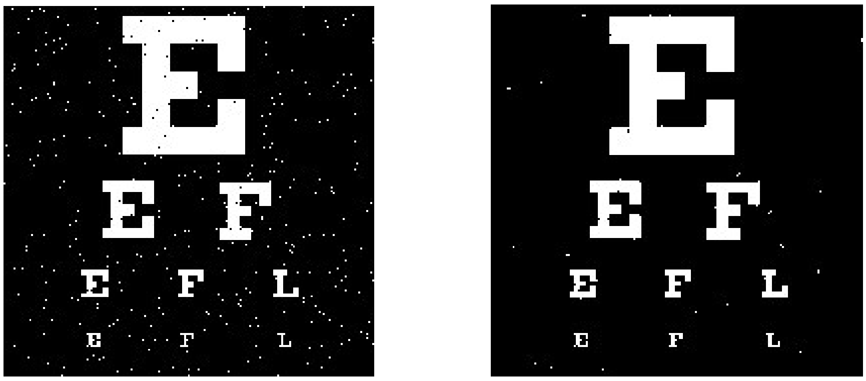

Figure 17 Isolated Pixel Cleaning for 1%. Left: 1% noisy image. Right: Image after cleaning.

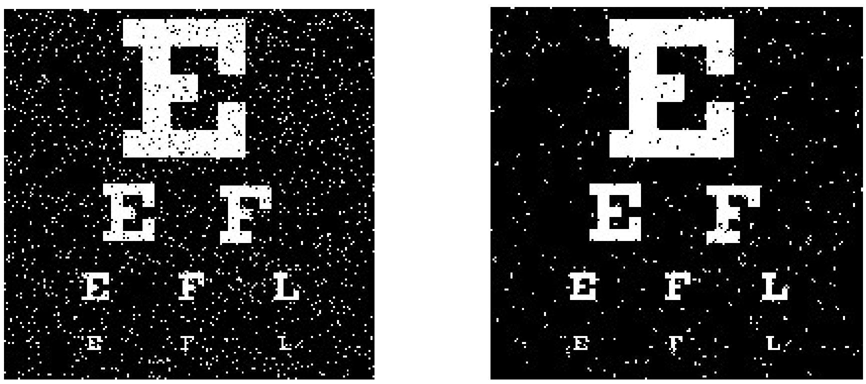

Figure 18 Isolated Pixel Cleaning for 5%. Left: 5% noisy image. Right: Image after cleaning.

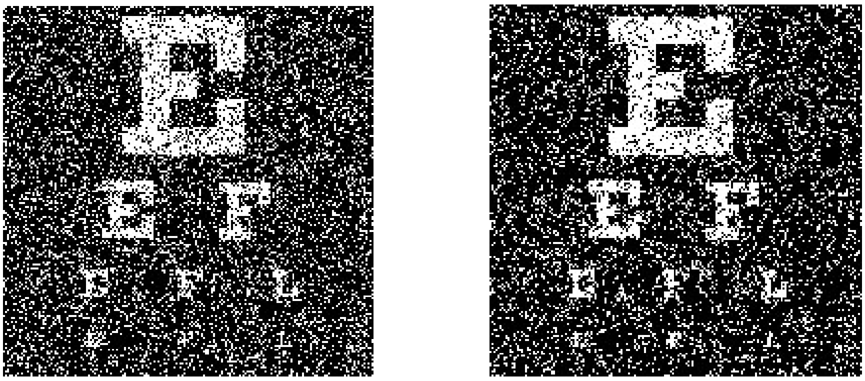

Figure 19 Isolated Pixel Cleaning for 20%. Left: 20% noisy image. Right: Image after cleaning.

2. Cleaning using opening and closing

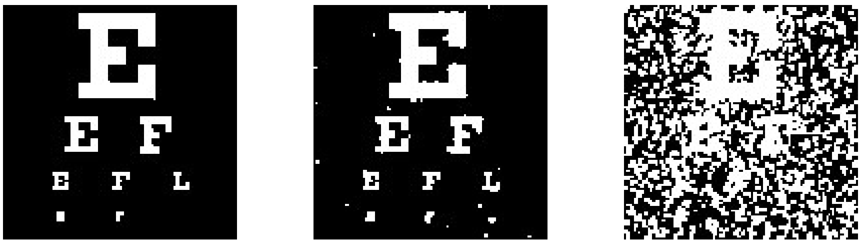

Figure 20 Images after closing-opening cleaning. 1: 1%. 2: 5%. 3:20%.

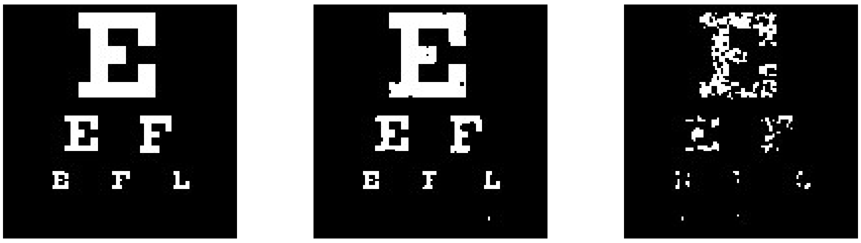

Figure 21 Images after opening-closing cleaning. 1: 1%. 2: 5%. 3:20%.

Actually the two operations are different. If the first operation just get rid of or delete some parts of the image, then it will not come back again. But if the first operation is in the opposite direction, the pixel may connect to the noise and becomes obvious. And these pixels may not be deleted for the second operation.

It appears to be better than Isolated Pixel Cleaning. But there is still some problem for the 20% noisy image.

Noise ratio	MSE before cleaning	MSE after Isolated Pixel Cleaning	MSE after Closing-Opening Cleaning	MSE after Opening-Closing Cleaning
1%	0.0105	0.0013	0.0019	0.0030
5%	0.0495	0.0179	0.0093	0.0073
20%	0.1991	0.1664	0.3775	0.0613

- ## Appendix

Hw3.m:

```
clear;clc;
%% P1
helmet_ori = imread('D:\ucsd\ece253\hw3\Helmet.jpg');
helmet = rgb2gray(helmet_ori);
noise = rand(size(helmet));
noisy_image = uint8(double(helmet) .* (noise > 0.2) + 255 .* (noise < 0.1)); % salt and pepper noise
figure;
subplot(211);
imshow(helmet);
title('original image');
subplot(212);
imshow(noisy_image);
title('noisy image');

% median filter
[x, y] = filters(helmet, noisy_image);
figure;
plot(x, y, 'o');
title('MSE plot with median filter');

% mean filter
[x1, y1] = mean_filters(helmet, noisy_image);
figure;
plot(x1, y1, 'o');
title('MSE plot with mean filter');

%% P2
image_synth = ones(128,1)*[64*ones(1,32) (64:4:192) 192*ones(1,32) 64*ones(1,32)];
m1 = fspecial('average', 3);
m2 = fspecial('average', 5);
m3 = fspecial('average', 7);
m4= fspecial('gaussian', 3, 0.5);
m5= 1/12*[1,1,1; 1,4,1; 1,1,1];
weight = [0.1, 1, 10];
% imshow(image_synth, [min(min(image_synth)), max(max(image_synth))]);

im_out_1_1 = uint8(unsharp(image_synth, m1, weight(1)));
im_out_1_2 = unsharp(image_synth, m1, weight(2));
im_out_2_2 = unsharp(image_synth, m2, weight(2));
im_out_3_2 = unsharp(image_synth, m3, weight(2));
im_out_4_2 = unsharp(image_synth, m4, weight(2));
```

```
im_out_5_2 = unsharp(image_synth, m5, weight(2));
im_out_1_3 = unsharp(image_synth, m1, weight(3));

figure;
subplot(311); imshow(im_out_1_2, [min(min(im_out_1_2)), max(max(im_out_1_2))]);
subplot(312); imshow(im_out_2_2, [min(min(im_out_2_2)), max(max(im_out_2_2))]);
subplot(313); imshow(im_out_3_2, [min(min(im_out_3_2)), max(max(im_out_3_2))]);
figure;
plot(image_synth(64, :), 'c');hold on
plot(im_out_1_2(64, :), 'r');hold on
plot(im_out_2_2(64, :), 'b');hold on
plot(im_out_3_2(64, :), 'y');
legend('original image', 'mean filter with 3*3', 'mean filter with 5*5', 'mean filter with 7*7');


figure;
subplot(311); imshow(im_out_1_2, [min(min(im_out_1_2)), max(max(im_out_1_2))]);
subplot(312); imshow(im_out_4_2, [min(min(im_out_4_2)), max(max(im_out_4_2))]);
subplot(313); imshow(im_out_5_2, [min(min(im_out_5_2)), max(max(im_out_5_2))]);
figure;
plot(image_synth(64, :), 'c');hold on
plot(im_out_1_2(64, :), 'r');hold on;
plot(im_out_4_2(64, :), 'b');hold on;
plot(im_out_5_2(64, :), 'y');
legend('original image', 'mean filter with 3*3', 'gaussian filter with 3*3', 'weighted mean filter with
3*3');

figure;
subplot(311); imshow(im_out_1_1, [min(min(im_out_1_1)), max(max(im_out_1_1))]);
subplot(312); imshow(im_out_1_2, [min(min(im_out_1_2)), max(max(im_out_1_2))]);
subplot(313); imshow(im_out_1_3, [min(min(im_out_1_3)), max(max(im_out_1_3))]);
figure;
plot(image_synth(64, :), 'c');hold on
plot(im_out_1_1(64, :), 'r');hold on;
plot(im_out_1_2(64, :), 'b');hold on;
plot(im_out_1_3(64, :), 'y');
legend('original image', 'mean filter with weight 0.1', 'mean filter with weight 1', 'mean filter with
weight 10');

xray_ori = imread('D:\ucsd\ece253\hw3\xray.tif');
me = fspecial('gaussian', 5, 1);
xray_enh = uint8(unsharp(xray_ori, m1, 25));
figure;
imshow(xray_ori);
```

```
figure;
imshow(xray_enh);

%% P3
eyechart_ori = imread('D:\ucsd\ece253\hw3\eyechart.tif');
binary_ori = 1 - round(double(eyechart_ori)/255);
eyechart_1 = imread('D:\ucsd\ece253\hw3\eyechart_1.tif');
binary_1 = 1 - round(double(eyechart_1)/255);
eyechart_5 = imread('D:\ucsd\ece253\hw3\eyechart_5.tif');
binary_5 = 1 - round(double(eyechart_5)/255);
eyechart_20 = imread('D:\ucsd\ece253\hw3\eyechart_20.tif');
binary_20 = 1 - round(double(eyechart_20)/255);

clean_1 = ipc(binary_1);
clean_5 = ipc(binary_5);
clean_20 = ipc(binary_20);

figure;
subplot(121);
imshow(binary_1);
subplot(122);
imshow(clean_1);
figure;
subplot(121);
imshow(binary_5);
subplot(122);
imshow(clean_5);
figure;
subplot(121);
imshow(binary_20);
subplot(122);
imshow(clean_20);

mse_before = zeros(1, 3);
mse_before(1) = img_mse(binary_ori, binary_1);
mse_before(2) = img_mse(binary_ori, binary_5);
mse_before(3) = img_mse(binary_ori, binary_20);
mse_after_ipc = zeros(1, 3);
mse_after_ipc(1) = img_mse(binary_ori, clean_1);
mse_after_ipc(2) = img_mse(binary_ori, clean_5);
mse_after_ipc(3) = img_mse(binary_ori, clean_20);

co_1 = bwmorph(bwmorph(binary_1, 'close'), 'open');
co_5 = bwmorph(bwmorph(binary_5, 'close'), 'open');
```

```
co_20 = bwmorph(bwmorph(binary_20, 'close'), 'open');
figure;
subplot(131);
imshow(co_1);
subplot(132);
imshow(co_5);
subplot(133);
imshow(co_20);
mse_after_co = zeros(1, 3);
mse_after_co(1) = img_mse(binary_ori, co_1);
mse_after_co(2) = img_mse(binary_ori, co_5);
mse_after_co(3) = img_mse(binary_ori, co_20);

oc_1 = bwmorph(bwmorph(binary_1, 'open'), 'close');
oc_5 = bwmorph(bwmorph(binary_5, 'open'), 'close');
oc_20 = bwmorph(bwmorph(binary_20, 'open'), 'close');
figure;
subplot(131);
imshow(oc_1);
subplot(132);
imshow(oc_5);
subplot(133);
imshow(oc_20);
mse_after_oc = zeros(1, 3);
mse_after_oc(1) = img_mse(binary_ori, oc_1);
mse_after_oc(2) = img_mse(binary_ori, oc_5);
mse_after_oc(3) = img_mse(binary_ori, oc_20);
```

---

filters.m

```
function [ x, y ] = filters( original, noisy )
x = [0 2 3 4 5 9 12 16 20 25 49];
y = zeros(1, 11);
y(1) = img_mse(original, noisy);

filtered_1 = medfilt2(noisy, [1 2]);
y(2) = img_mse(original, filtered_1);

filtered_2 = medfilt2(noisy, [1 3]);
y(3) = img_mse(original, filtered_2);

filtered_3= medfilt2(noisy, [2 2]);
y(4) = img_mse(original, filtered_3);

filtered_4 = uint8(median_filter_5c(noisy));
```

```
y(5) = img_mse(original, filtered_4);

filtered_5 = medfilt2(noisy, [3 3]);
y(6) = img_mse(original, filtered_5);

filtered_6 = medfilt2(noisy, [3 4]);
y(7) = img_mse(original, filtered_6);

filtered_7 = medfilt2(noisy, [4 4]);
y(8) = img_mse(original, filtered_7);

filtered_8 = medfilt2(noisy, [4 5]);
y(9) = img_mse(original, filtered_8);

filtered_9 = medfilt2(noisy, [5 5]);
y(10) = img_mse(original, filtered_9);

filtered_10 = medfilt2(noisy, [7 7]);
y(11) = img_mse(original, filtered_10);

figure;
subplot(221)
imshow(filtered_1);
title('image after 1*2 median filter')
subplot(222)
imshow(filtered_2);
title('image after 1*3 median filter')
subplot(223)
imshow(filtered_3);
title('image after 2*2 median filter')
subplot(224)
imshow(filtered_4);
title('image after cross shaped filter')

figure;
subplot(221)
imshow(filtered_5);
title('image after 3*3 median filter')
subplot(222)
imshow(filtered_6);
title('image after 3*4 median filter')
subplot(223)
imshow(filtered_7);
title('image after 4*4 median filter')
```

```
subplot(224)
imshow(filtered_8);
title('image after 4*5 median filter')

figure;
subplot(211)
imshow(filtered_9);
title('image after 5*5 median filter')
subplot(212)
imshow(filtered_10);
title('image after 7*7 median filter')
end
```

---

median_filter_5c.m

```
function output_img = median_filter_5c( input_img )
    [rows, cols] = size(input_img);
    output_img = zeros(rows, cols);
    arr = zeros(5);
    for i = 2:rows-1
        for j = 2:cols-1
            arr = [input_img(i,j), input_img(i-1,j), input_img(i,j-1), input_img(i+1,j), input_img(i,j+1)];
            output_img(i, j) = median(arr);
        end
    end

end
```

---

mean_filters.m

```
function [ x, y ] = mean_filters( original, noisy )
   x = [0 9 25 49];
   y = zeros(1, 4);
   y(1) = img_mse(original, noisy);

   kernel = 1/9*ones(3,3);
   im_1 = uint8(conv2(double(noisy), kernel, 'same'));
   y(2) = img_mse(original, im_1);

   kernel = 1/25*ones(5,5);
   im_2 = uint8(conv2(double(noisy), kernel, 'same'));
   y(3) = img_mse(original, im_2);

   kernel = 1/49*ones(7,7);
   im_3 = uint8(conv2(double(noisy), kernel, 'same'));
```

```
    y(3) = img_mse(original, im_3);

    figure;
    subplot(311);
    imshow(im_1);
    title('image after 3*3 mean filter');
    subplot(312);
    imshow(im_2);
    title('image after 5*5 mean filter');
    subplot(313);
    imshow(im_3);
    title('image after 7*7 mean filter');

end
```

---

img_mse.m

```
function [ mse ] = img_mse( img1, img2 )
     mse = (sum(sum((double (img1) - double (img2)).^2)))/numel(img1) ;
end
```

---

unsharp.m

```
function im_out = unsharp( im_in, maskA, weight )
[a,b] = size(maskA);
maskB = zeros( size(maskA) );
maskB(ceil(a/2), ceil(b/2)) = 1;
maskC = maskB - maskA;
maskD = maskB + weight * maskC;
im_out = conv2(double(im_in), double(maskD), 'same');
% figure;
% subplot(121);
% imshow(uint8(conv2(im_in, maskA, 'same')));
% subplot(122);
% imshow(conv2(im_in, maskC, 'same'));
End
```

Ipc.m

```
function output_img = ipc( noisy )
mask = [ 1 8 64; 2 16 128; 4 32 256];
im_tmp = filter2(mask,noisy);
tmp = (im_tmp ~= 16);
output_img = noisy .* tmp;
tmp = (im_tmp == 495);
output_img = output_img + tmp;
end
```