School of Data and Computer Science, Sun Yat-sen University

PROJECT 03-05
I - Enhancement Using the Laplacian
and
PROJECT 03-06
II - Unsharp Masking

17341137 Zhenpeng Song

September 18, 2019

Abstract

Implementation of two experiments:

- I - Enhancement Using the Laplacian to get an image with some grey edge lines and catastrophe points upon a dark background
- II - Unsharp Masking to clarify a blurred image.

1 Technical discussion

- I - Enhancement Using the Laplacian

  First of all, please refer to the definitions blow:

  - mask: the filter for laplacian filtering.
  - scale: to normalize the output to avoid excessive 'black' area.
  - spatialfilter(I, mask): the function written in matlab for generic use to do spatial filter on I with mask.

  Laplacian Filtering is conducted with the following steps:

  1. Choose proper Mask: In this project, I choose two regular mask:

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, M_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

  2. Choose proper Scale: In this project, I do two groups of research:
     - scales < 0 and scales > 0
       Research on: scales = [-2 -1 1 2]

     - Gradient negative scales
       Research on: scales = [-4.0 -3.5 ... -1.0 -0.5]
  3. According to the equation:

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

- II - Unsharp Masking

  In this Project, I use Laplacian-based high-boosting filter, which is defined as:

$$f_{hb}(x,y) = \begin{cases} Af(x,y) - \nabla^2 f, & mid < 0 \\ Af(x,y) + \nabla^2 f, & mid > 0 \end{cases}$$

  That is to say, my main job is to find out which A performs best in unsharp masking.

2 Discussion

- I - Enhancement Using the Laplacian

  - When implementing the spatialfilter function, a tricky step is to map the mask's scale to matlab style(1-m, 1-n).
  - In terms of the boundary handling, I use the mirroring strategy, that is, when the code visits an out-of-bound address, It will do a reflection and take the boundary row/column as the center line.
  - As shown below, I summarize how scale affect the output image:
    - no scale: I only get several figures, lines, flash points in the image, because the value of each pixel is limited to [0 L-1], some negative value generated by laplacian method is abolished and replaced by 0, that's why the image is almost in black.
    - positive scale: Positive scale will blur the image, because I'm using masks with positive mid-value, a positive scale violate the basic method of laplacian:

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f, & mid < 0 \\ f(x,y) + \nabla^2 f, & mid > 0 \end{cases}$$

    - negative scale: Negative scales do clarify the image, most features become brighter and clearer, and in my research, a value of -2 or -2.5 appears to be a acceptable choice.
  - After all, both of two different masks performs well.

- II - Unsharp Masking

  - I decided to do high-boosting filter with Laplacian method, and it performed rather good.
  - With those 'A's I had tried, a value of 2 may be the best choice among them. In detail, the image not only went brighter, but many features concealed before(mainly caused by the darkness) took on a clearer look.

3 Results

- I - Enhancement Using the Laplacian
  - Mask = M1

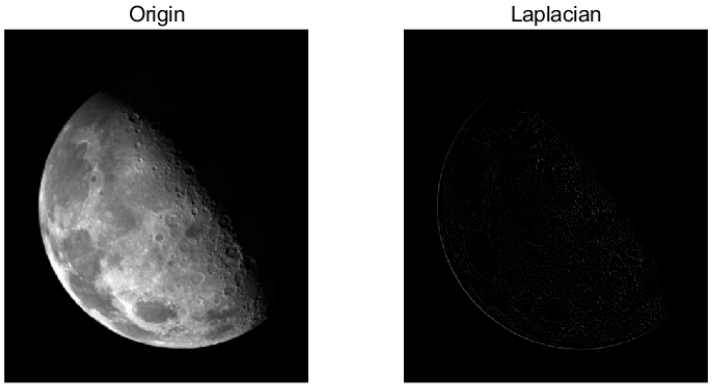

Figure 1: [M1]Origin-brief-laplacian

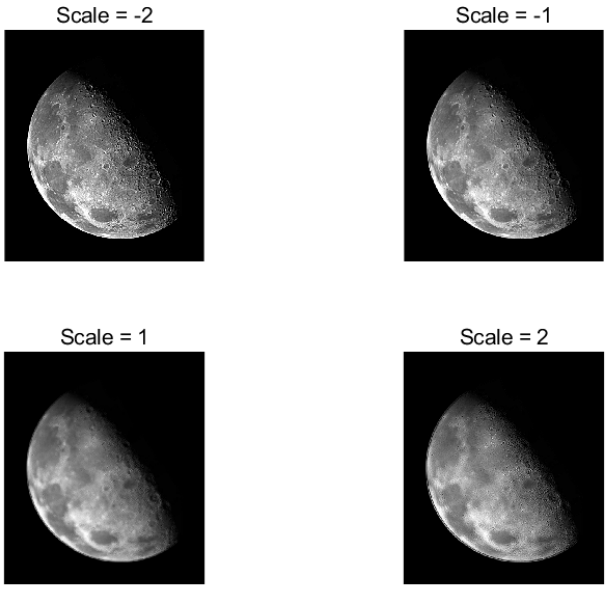

Figure 2: [M1]neg-pos-scale-test

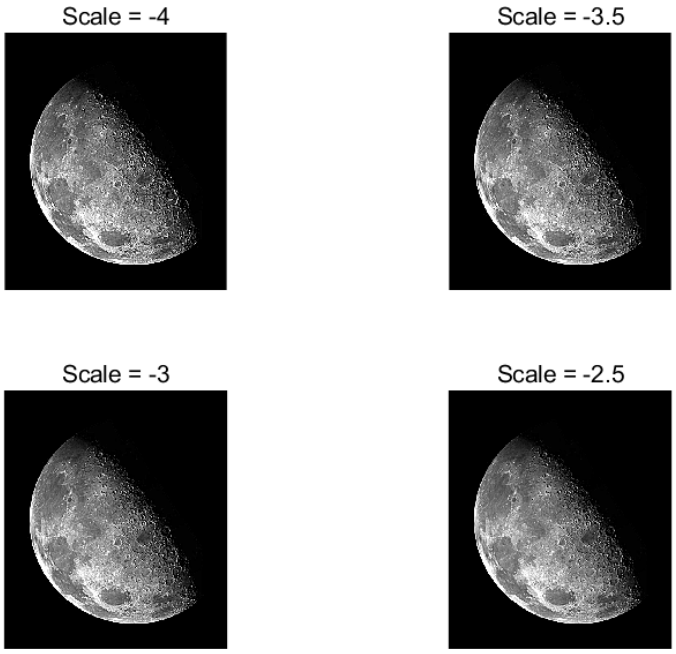

Figure 3: [M1]neg-gradient-scale1

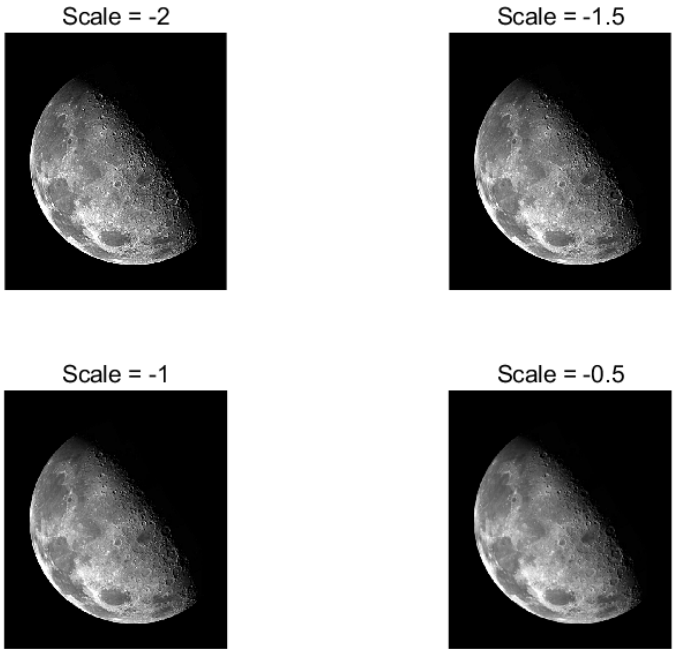

Figure 4: [M1]neg-gradient-scale2

– Mask = M2

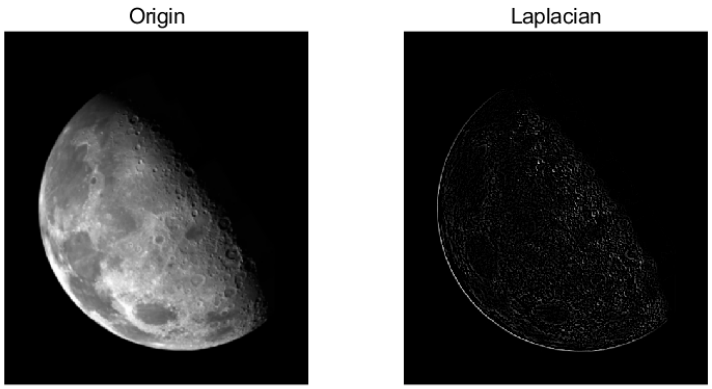

Figure 5: [M2]Origin-brief-laplacian

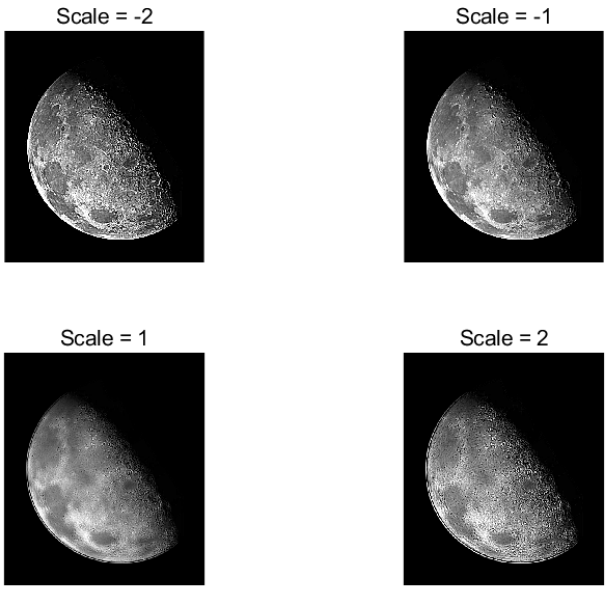

Figure 6: [M2]neg-pos-scale-test

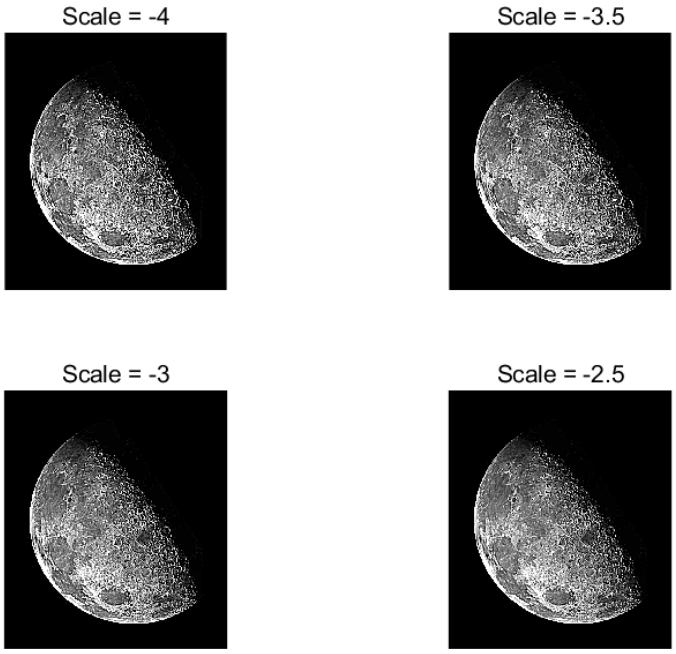

Figure 7: [M2]neg-gradient-scale1

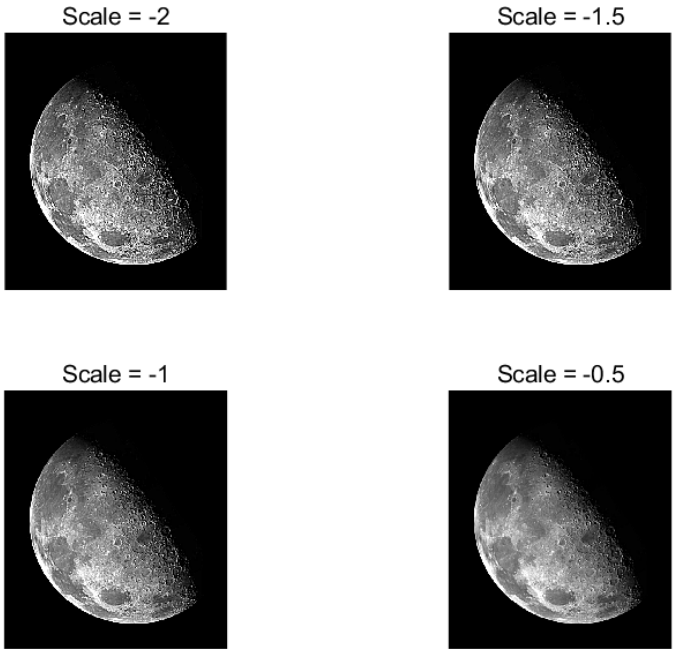

Figure 8: [M2]neg-gradient-scale2

- II - Unsharp Masking
  - Mask = M1

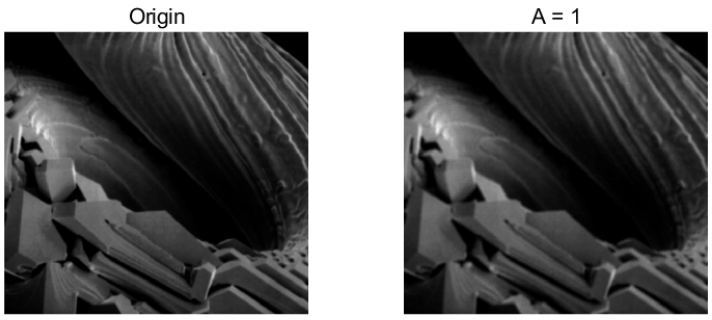

Figure 9: [M1] A = 1

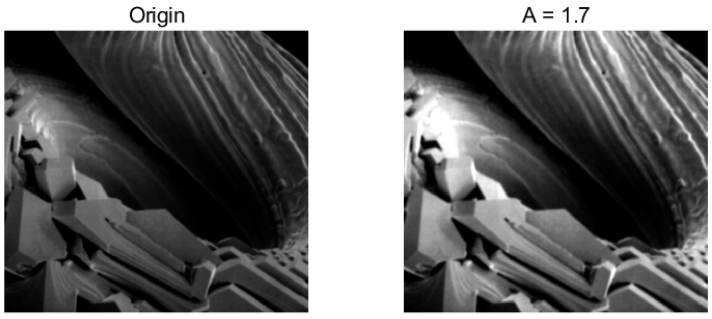

Figure 10: [M1] A = 1.7

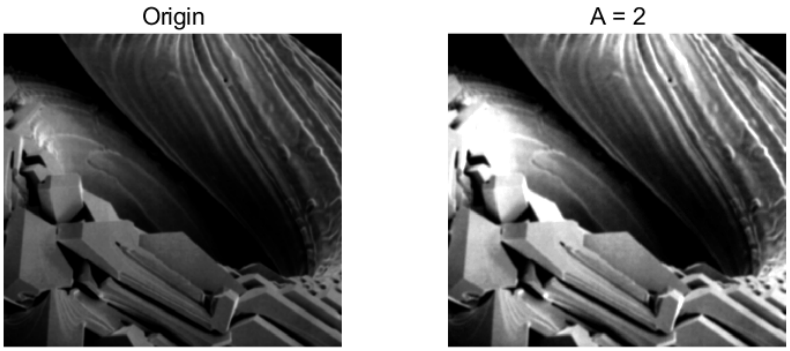

Figure 11: [M1] A = 2

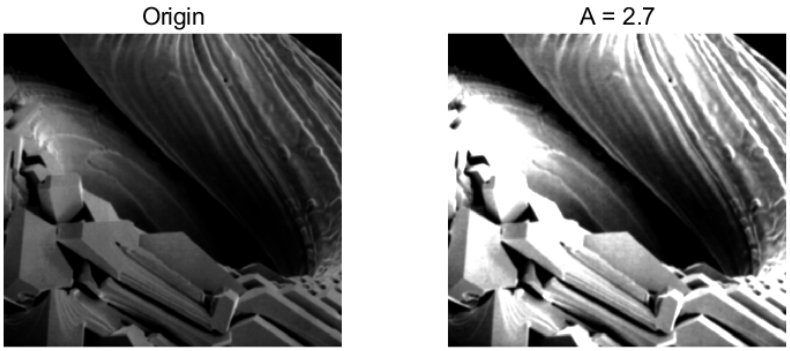

Figure 12: [M1] A = 2.7

– Mask = M2

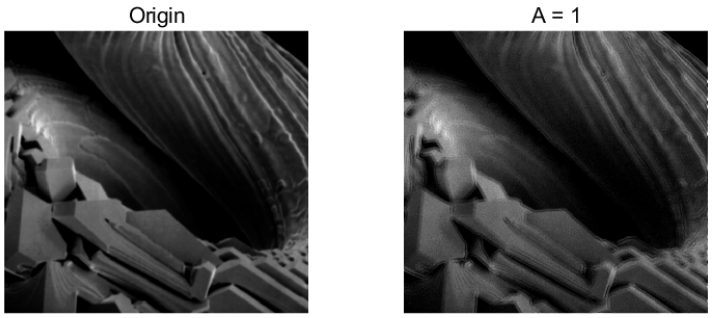

Figure 13: [M2] A = 1

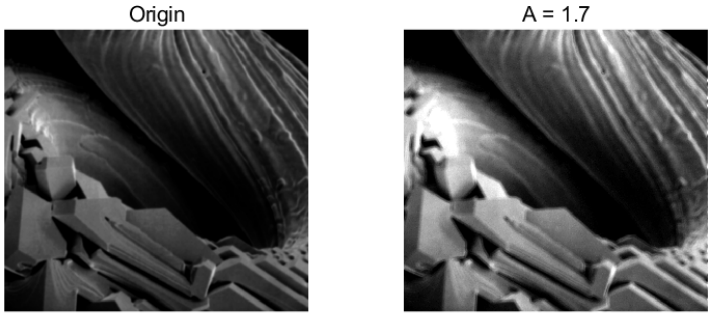

Figure 14: [M2] A = 1.7

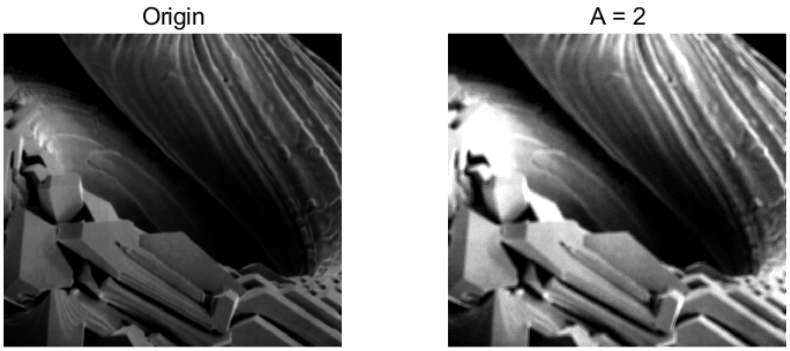

Figure 15: [M2] A = 2

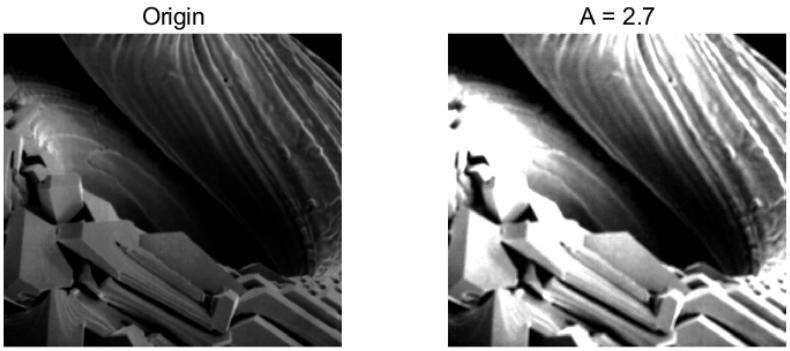

Figure 16: [M2] A = 2.7

4 Appendix

- Func: spatialfilter(I, mask)

```
function [ res ] = spatialfilter(I, mask)

    [M,N] = size(I);
    [m,n] = size(mask); % Generally as [3, 3]

    res = zeros(M,N);
    for x = 1:M
        for y = 1:N
            for i = 1:m
                for j = 1:n
                    % [1-m/2 ~ m-1/2 ] >>> [1 ~ m]
                    target_x = x-i+(m+1)/2;
                    target_y = y-j+(n+1)/2;
                    % Boundary handling [mirroring]
                    if(target_x <= 0)
                        target_x = 1-target_x;
                    else
                        if(target_x > M)
                            target_x = M-(target_x-M);
                        end
                    end
                    if(target_y <= 0)
                        target_y = 1-target_y;
                    else
                        if(target_y > N)
                            target_y = N-(target_y-N);
                        end
                    end
                    % Calculation
                    res(x,y) = res(x,y) + I(target_x,target_y) * mask(i,j);
                end
            end
        end
    end
end
```

- I - Enhancement Using the Laplacian

```
% PJ0305
clc, clear;

I = im2double(imread("../assets/Fig3.40(a).jpg"));

% mask = [0 1 0; 1 -4 1; 0 1 0];
mask = [1 1 1; 1 -8 1; 1  1 1];
scale = [-2 -1 1 2];

% Laplacian without scaling
laplacian_I = spatialfilter(I, mask);
figure(1);
subplot(121); imshow(I), title("Origin");
subplot(122); imshow(laplacian_I), title("Laplacian");
saveas(gcf, "Origin-brief-laplacian.png")

% Laplacian with scaling
figure(2);
for i = 1:4
    scaled_laplacian_I = I + scale(i) .* laplacian_I;
    subplot(2, 2, i); imshow(scaled_laplacian_I), title("Scale = " + scale(i))
end
saveas(gcf, "neg-pos-scaletest.png")

% negative gradient scales
neg_scale = -4:0.5:-0.5;
figure(3);
for i = 1:4
    scaled_laplacian_I = I + neg_scale(i) .* laplacian_I;
    subplot(2, 2, i); imshow(scaled_laplacian_I), title("Scale = " + neg_scale
end
saveas(gcf, "neg-gd-scale1.png")
figure(4);
for i = 5:8
    scaled_laplacian_I = I + neg_scale(i) .* laplacian_I;
    subplot(2, 2, i-4); imshow(scaled_laplacian_I), title("Scale = " + neg_sca
end
saveas(gcf, "neg-gd-scale2.png")
```

- II - Unsharp Masking

```
% PJ0306
clear, clc;

I = im2double(imread("../assets/Fig3.43(a).jpg"));

% mask = [0 1 0; 1 -4 1; 0 1 0];
mask = [1 1 1; 1 -8 1; 1 1 1];

A = [1 1.7 2 2.7];
laplacian_I = spatialfilter(I, mask);
for i = 1:4
    hb_I = A(i) .* I + laplacian_I;
    figure(i);
    subplot(121); imshow(I), title("Origin");
    subplot(122); imshow(hb_I), title("A = " + A(i));
    saveas(gcf, "Unsharp masking(A = " + A(i) + ").png")
end
```