Digital Image Processing

Lab #3. Image enhancement

학과 : 전자공학과

학번 : 21611648

이름 : 유준상

담당교수 : 김성호

Table of Contents

Introduction

Histogram Equalization은 사진을 선명하게 해줄 수 있다. 어둡게 또는 밝게 아니면 한 밝기 레벨에만 픽셀들이 집중되어 있다면 이미지가 갖고 있는 내용을 사람이 이해하기 힘들다. 이럴 때에는 사진을 선명하게 해줄 필요가 있다.

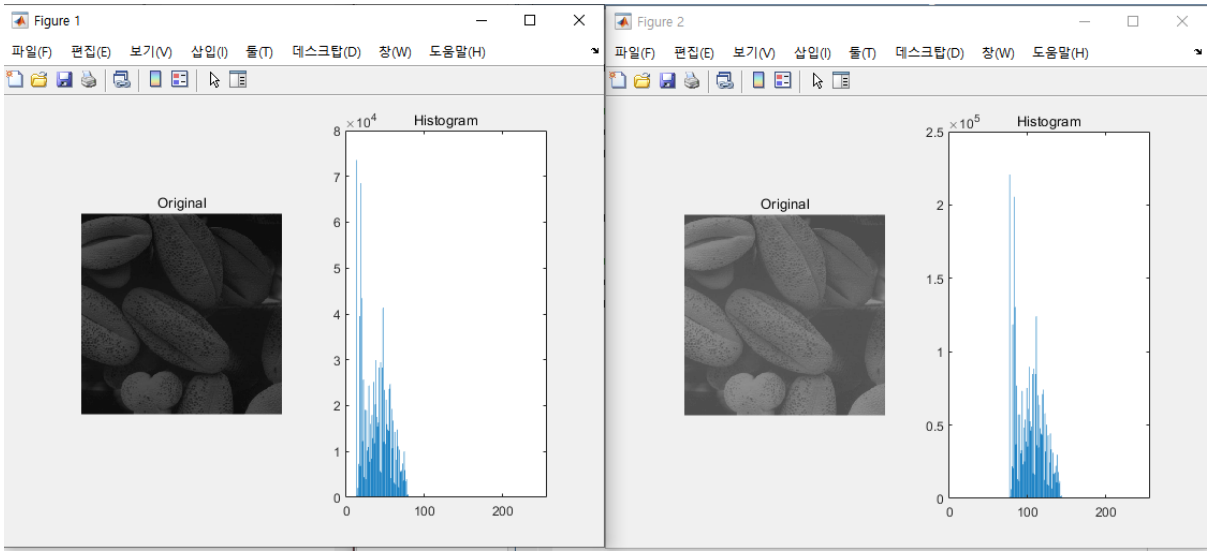
사진을 선명하게 하는 방법은 간단하다. 일정 영역에 모여있는 픽셀들을 0~255 범위의 밝기 레벨로 퍼뜨려주면된다. 이번 Lab #3에서는 그 방법에 대해 실험해본다.

1. Lab 3-1

1) source code

```
편집기 - C:₩Users₩beatr₩Documents₩MATLAB₩DIP₩week3₩lab3_1.m
lab3_1.m
1 -     I=imread('dark.tif');
2 -     figure(1); subplot(1,2,1); imshow(I); title('Original');
3
4       % histogram
5 -     [count center]=hist(double(I(:)),0:255)
6 -     figure(1); subplot(1,2,2); bar(center, count); title('Histogram');
7
8 -     I=imread('light.tif'); %gray image
9 -     figure(2); subplot(1,2,1); imshow(I); title('Original');
10
11      % histogram
12 -    [count center]=hist(double(I(:)),0:255)
13 -    figure(2); subplot(1,2,2); bar(center, count); title('Histogram');
```

2) result figure



3) discussions

imread로 작업하는 폴더에서 어두운 이미지와 밝은 이미지를 읽어오고 figure를 만든 후, imshow로 읽어온 이미지를 디스플레이 했습니다. 그리고 해당 figure에 밝기 레벨 당 픽셀의 개수를 알아보도록 Histogram을 만들어서 디스플레이 했습니다. 어두운 이미지에서는 비교적 낮은 밝기 레벨에 많이 분포 되어 있었고, 밝은 이미지에서는 어두운 이미지 보다는 비교적 높은 밝기 레벨에

분포 되어 있는 것을 확인했습니다

2. Color to gray transform

1) source code

편집기 - C:₩Users₩beatr₩Documents₩MATLAB₩DIP₩week3₩lab3_2.m

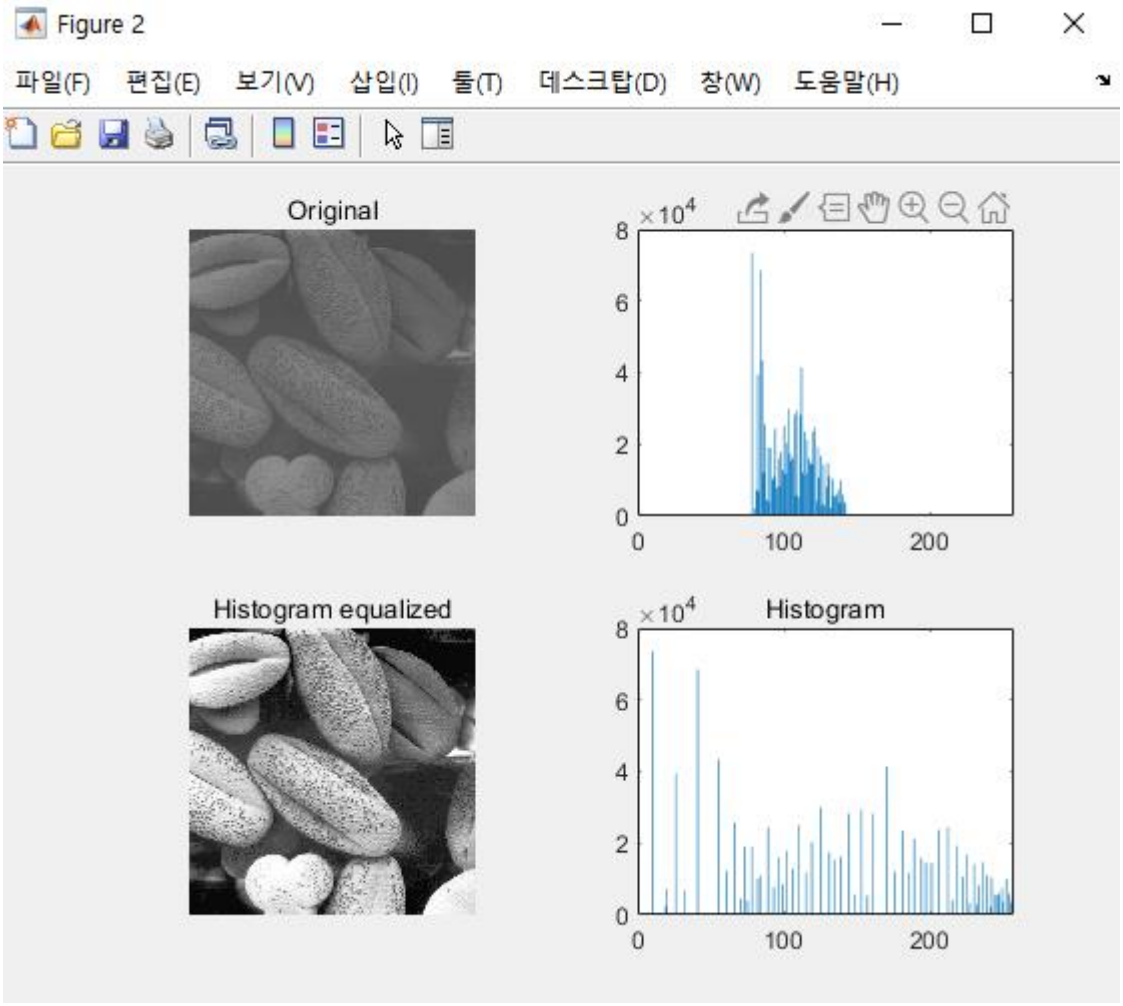
lab3_1.m | lab3_2.m

```
% Equalization
I=imread('light.tif');
I=rgb2gray(I);
h1=imhist(I,256);

Ieq=histeq(I,256);
h2=imhist(Ieq,256);

figure(2); subplot(2,2,1); imshow(I), title('Original');
subplot(2,2,2); bar(h1), title('Histogram');
subplot(2,2,3); imshow(Ieq), title('Histogram equalized');
subplot(2,2,4); bar(h2), title('Histogram');
```

2) result figure



3) discussions

원본 이미지를 먼저 불러오고, 해당 이미지의 히스토그램을 확인했습니다. 그 후, Histogram Equalizing(built-in func)를 사용해서 밝기 레벨 100 부근에 모여있는 픽셀들의 밝기 값을 떨어 뜨려 밝기 레벨 전체적으로 평평하게 만들게 떨어뜨려 놓았습니다. 그 결과, 이미지의 대조비가 높아져 선명하게 이미지를 볼 수 있습니다.

3. Lab 3-3

1) source code

편집기 - C:₩Users₩beatr₩Documents₩MATLAB₩DIP₩week3₩myhisteq.m

lab3_1.m | lab3_3.m | myhisteq.m

```
function s=myhisteq(image,l)
img=double(image);
[M,N]=size(img);
L=0:l-1;
% number of pixels of each intensity
n=zeros(1,l);
for i = 1: M
    for j=1:N
        for k=1:l
            if img(i,j)==L(k)
                n(k)=n(k)+1;
            end
        end
    end
end
% Convert cumulative intensities into integers
out_int=zeros(1,l);

for k=1:l
    %probability of each intensity
    p_r=n(1:k);
    % Cumulative sum of probabilities
    pp_r=sum(p_r);
    out_int(k)=((l-1)/(M*N))*pp_r;
end
```

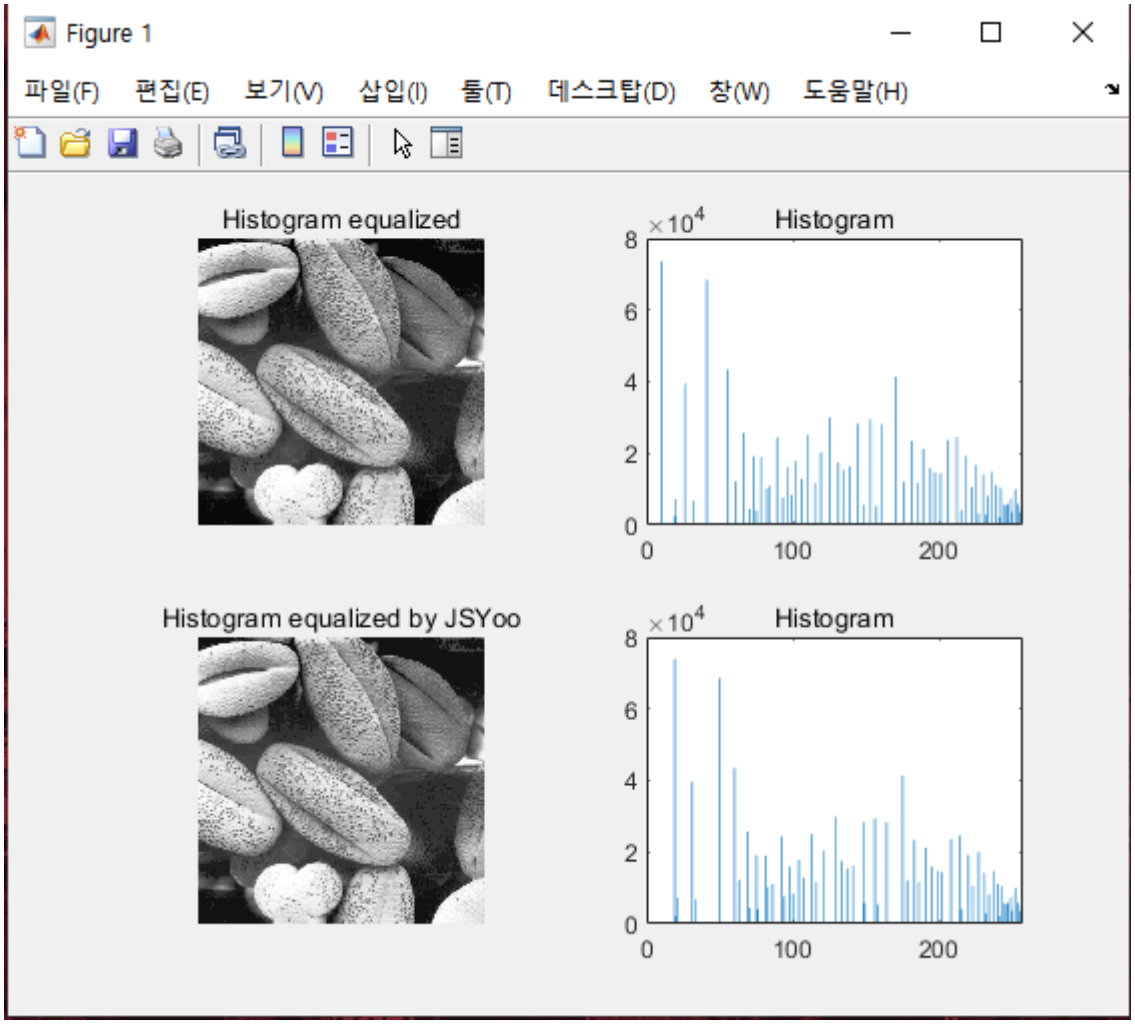
```
img_out=zeros(M,N);
for i=1:M
    for j=1:N
        img_out(i,j)=out_int(img(i,j)+1);
    end
end
s=uint8(img_out);
end
```

편집기 - C:₩Users₩beatr₩Documents₩MATLAB₩DIP₩week3₩lab3_3.m

lab3_1.m | lab3_3.m | myhisteq.m

```
1 -    clc; close all;
2      % Equalization
3 -    I=imread('light.tif');
4 -    I=rgb2gray(I);
5 -    h1=imhist(I,256);
6
7 -    Ieq=histeq(I,256);
8 -    h2=imhist(Ieq,256);
9
10 -   subplot(2,2,1); imshow(Ieq), title('Histogram equalized');
11 -   subplot(2,2,2); bar(h2), title('Histogram');
12
13     % my code
14 -   myeq=myhisteq(I,256);
15 -   h3=imhist(myeq,256);
16
17 -   subplot(2,2,3); imshow(uint8(myeq)), title('Histogram equalized by JSYoo');
18 -   subplot(2,2,4); bar (h3), title('Histogram');
```

2) result figure



3) discussions

직접 만든 Histogram Equalization 함수를 사용해서 처리한 이미지가 아래이고, 위는 built-in function인 histeq를 사용한 결과이다. 적절히 Equalization된 결과를 확인할 수 있었다. 비교를 해봐도 육안으로 어느 것이 조금 더 월등하다는 것은 확인이 어렵고, 비슷한 것 같다.

Conclusion

대부분의 사진 편집 어플리케이션에 탑재되어 있는 기본 기능 중에 하나가 바로, 사진을 선명하게 바꿔주는 기능이다. 사용할 때는 어떻게 한 것인지에 대한 생각이 딱히 없었는 것이 사실이다. 그러나, 강의를 듣다 보니 그 원리가 궁금해졌고, 직접해보니 더욱 재밌었다. Histogram을 좀더 flat하게 바꾸려면 어떤 수식을 사용해야할지 궁금하다.