

Digital
Image
Process
Report #5



Histogram Equalization

로봇기계공학과
21721160
최병희

Lab 3-3 MATLAB: Histogram Equalization

- 문제 - histeq 커스텀 함수 구현

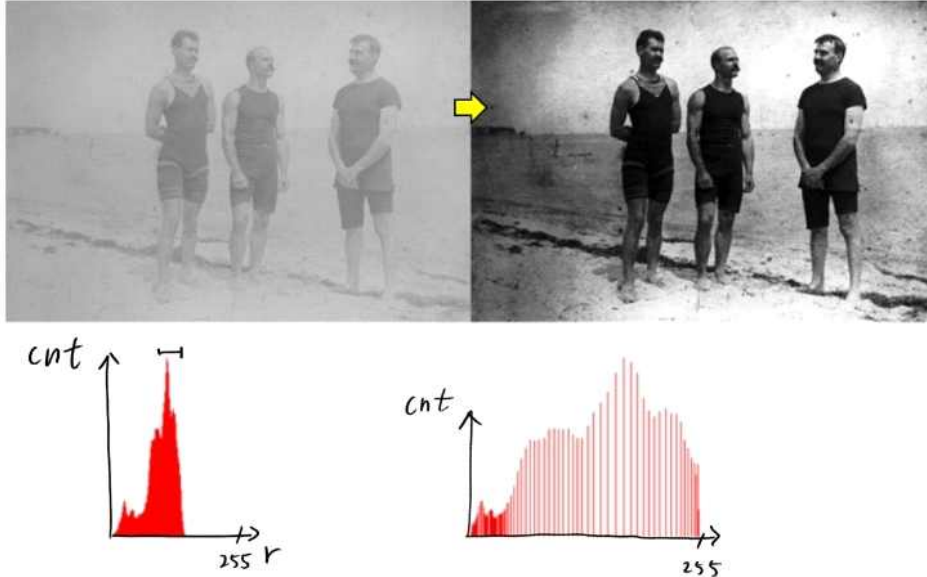


그림 2 Histogram Equalization의 목적

Histogram Equalization은 이미지 밝기의 Histogram의 형태를 변화시켜 이미지의 전반적인 밝기를 조정하는 방법입니다. 그림 2에서의 왼쪽 Histogram은 왼쪽 이미지의 이미지 픽셀 intensity의 개수를 나타낸 그래프입니다. 이미지 픽셀 intensity가 낮은 쪽으로 몰려 있음을 확인할 수 있습니다. 사람의 눈은 contrast가 높은 이미지를 보기 편하다고 느끼기 때문에 이 Histogram을 contrast가 높은 이미지의 Histogram과 같은 분포를 갖도록 바꾸어 줄 필요가 있습니다. 따라서 contrast가 높은 이미지로 바꾸었을 때의 Histogram은 오른쪽 그래프와 같습니다.

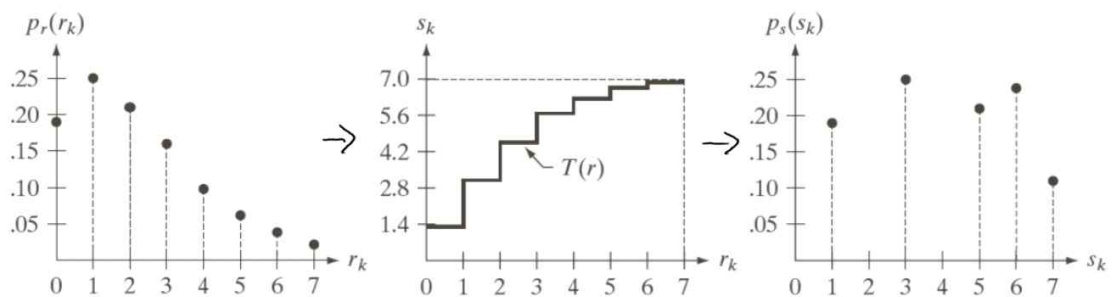


그림 3 Histogram Equalization의 과정

대략적인 방법은 원래 intensity r 값을 이미지 Transform 함수 $T(r)$ 에 대입하여 나온 S 값으로 다시 이미지를 구성하는 것입니다. 이때 intensity를 Histogram 영역에서 계산하여, normalize된 값을 바탕으로 Transform 한 뒤 다시 이 값을 바탕으로 Equalized Histogram을 그립니다. 새로운 Histogram을 따르는 이미지의 각 픽셀값은 가운데의 $s_k - r_k$ 그래프를 참고하여 저장합니다.

- Source code

myhisteq.m

```
function [img, S, hist, equalized_hist, pr_rk, ps_sk] = myhisteq(image,
intensity_level)
[row column]= size(image);
hist = zeros(1,intensity_level);
for i = 1:row
    for j = 1:column
        hist(image(i,j)) = hist(image(i,j)) + 1;
    end
end
pr_rk = hist/(row*column);
for i = 1:intensity_level
    S(i) = round(((intensity_level-1)/(row*column))*sum(hist(1:i)));
end
equalized_hist = zeros(1,length(S));
for i = 1:length(S)
    if S(i) ~= 0
        equalized_hist(S(i)) = equalized_hist(S(i)) + pr_rk(i);
    end
end
ps_sk = equalized_hist;
equalized_hist = ps_sk*(row*column);
img = uint8(S(image));
end
```

DIP_5.m

```
clc; clear all; close all;
img = imread('data/dark.tif');
[eq_img, S, hist, eq_hist, pr_rk, ps_sk] = myhisteq(img, 256);
builtin_eq_img = histeq(img, 256);
figure(1);
subplot(1,3,1);
imshow(img);
title('Original Image');
subplot(1,3,2);
imshow(eq_img);
title('Equalized Image');
subplot(1,3,3);
imshow(builtin_eq_img);
title('Built-in Equalized Image');
figure(2); subplot(1,2,1);
bar(hist);
xlabel('r_k'); ylabel('n_k');
title('Original Histogram');
subplot(1,2,2);
bar(eq_hist);
xlabel('r_k'); ylabel('n_k');
title('Equalized Histogram');
figure(3);
bar(S);
title('s=T(r)');
xlabel('r_k'); ylabel('s_k');
figure(4); subplot(1,2,1);
bar(pr_rk);
xlabel('r_k'); ylabel('p_{r}(r_k)');
title('Normalized Original Histogram');
subplot(1,2,2);
bar(ps_sk);
xlabel('r_k'); ylabel('p_{s}(s_k)');
title('Normalized Equalized Histogram');
```

- 결과 비교 및 배운 점

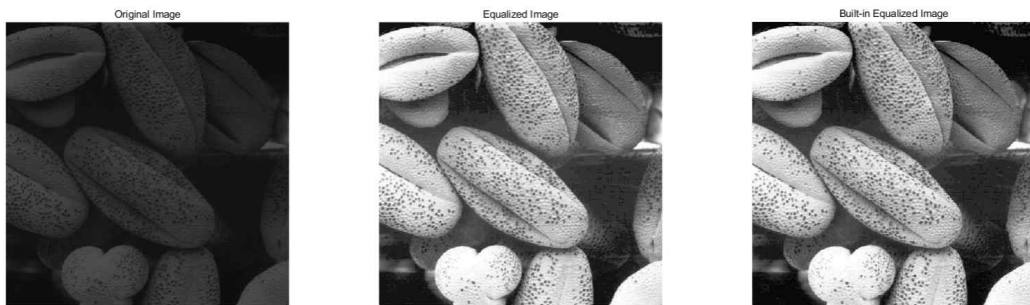


그림 4 원본, myhisteq, histeq 내장함수의 결과

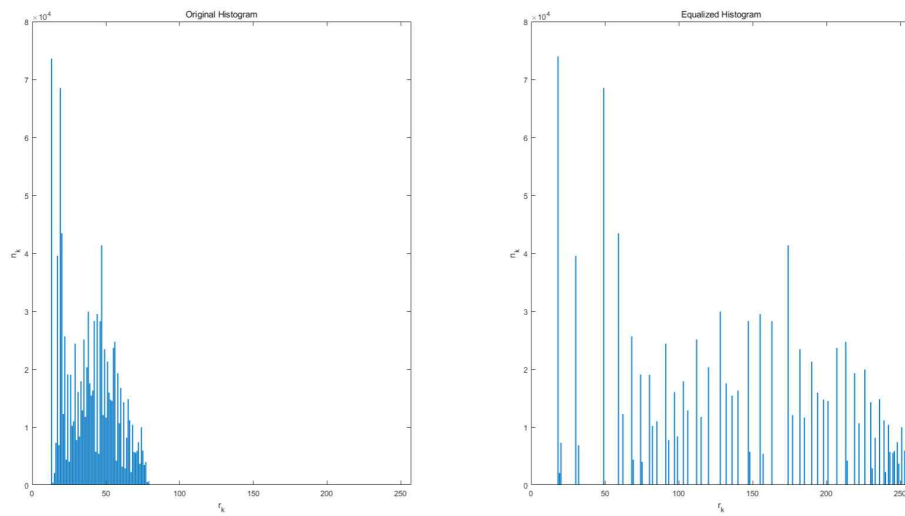


그림 5 원본 Histogram, Equalized Histogram

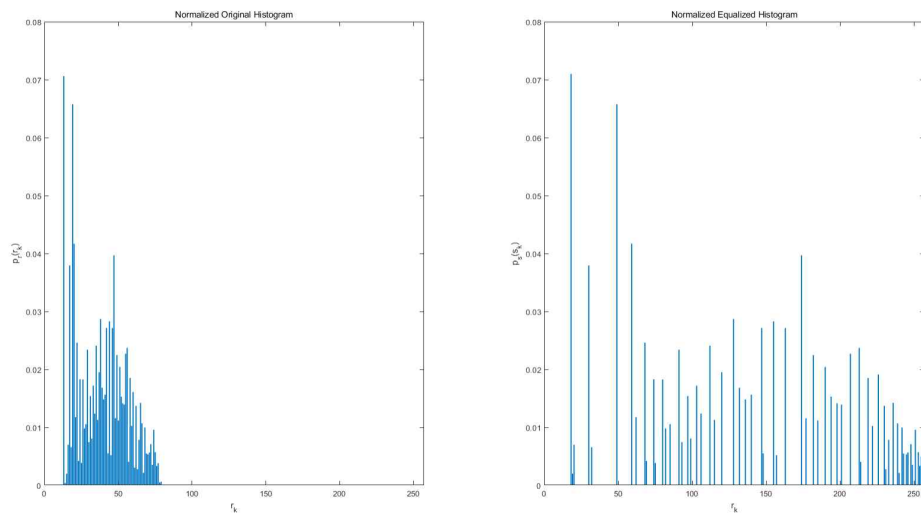


그림 6 원본 Normalized Histogram, Normalized Equalized Histogram

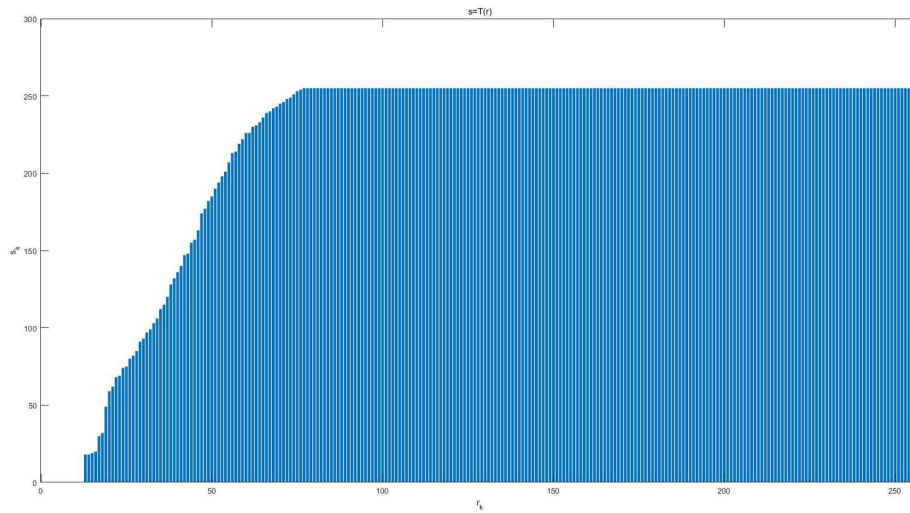


그림 7 $s = T(r)$

그림 4를 통해 Matlab 내장함수와 myhisteq를 비교했을 때 거의 차이가 없는 것을 볼 수 있습니다. 또한 Histogram상에서도 원본 Histogram은 약 20~80 사이에 픽셀 밝기가 분포해 있는 반면, Equalized Histogram은 20~255 사이에 픽셀 밝기가 분포하여 원본 Histogram보다 훨씬 큰 contrast값을 보입니다. 그림 6은 원본 이미지와 Equalized 이미지의 정규화된 Histogram으로 각각 $r_k - p_r(r_k)$ 그래프, $s_k - p_s(s_k)$ 그래프입니다.

그림 7은 원본 intensity r 을 equalized intensity s 로 표현할 수 있는 Transform 함수를 나타낸 것입니다. 계단 형태로 잘 나타난 것을 볼 수 있습니다.