

Discrete values:

$$S_k = T(r_k) = (L-1) \sum_{j=0}^k P_r(r_j)$$

$$= (L-1) \sum_{j=0}^k \frac{n_j}{MN} = \frac{L-1}{MN} \sum_{j=0}^k n_j \rightarrow k = 0, 1, \dots, L-1$$

19페이지 히스토그램 값을 내장 함수 없이 구현하기 위한 알고리즘이다.

```
function [out] = myhisteq(input,n) %input = image, n = intensity level range
% discrete domain

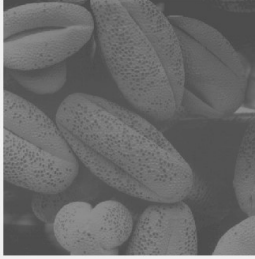
[nj r] = imhist(input); %nj = amount of each value, r = intensity value
L=n;
[M N] = size(input);
MN = M*N;
Pr = nj/MN;
Sk = [];
% p = [];
for i = 1:256
    sum = 0;
    for j = 1:i
        sum = sum + Pr(j);
    end
    % p(i) = sum;
    Sk(i) = (L-1)*sum;
end

Ieq1D = Sk(input(:)+1); %r=0->index 1
out = reshape(Ieq1D, [M, N]); %Ieq2D

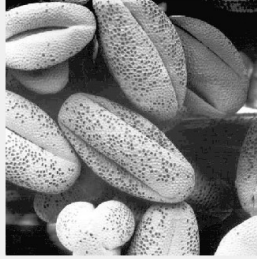
out = uint8(out);
```

input 값으로 이미지를 받고 n으로 밝기의 범위를 받고, imhist를 사용하여 각 밝기 값마다 빈도를 측정해준다. 그 후 input의 행렬크기를 구한 후 Pr을 구해준다. for문 내는 실습때 했던 cumsum함수를 구현한 코드이다. 해당 값을 이용하여 Sk를 구해주고 out을 출력해준다.

Original



original Histogram equalized



my Histogram equalized

