

MFC 영상처리

목차

01



이미지 처리 - 필터

02

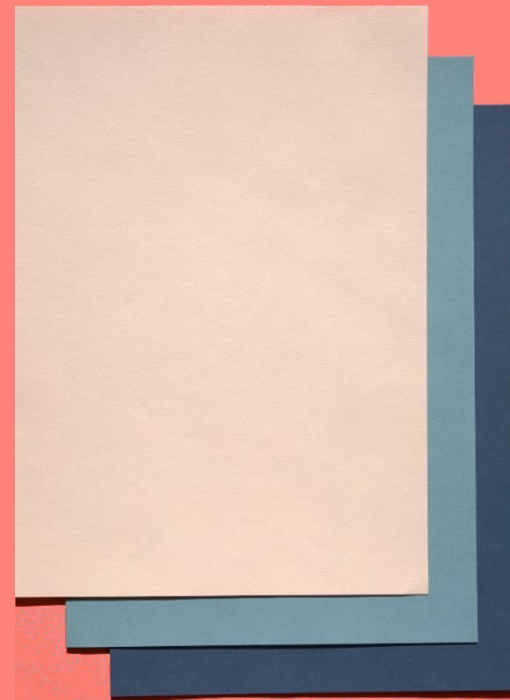


설정

03



기타 기능



이미지 처리 필터 - Gray Scale

그레이 필터 = 공식을 이용해서 영상을 회색화 하는 영상처리

그레이 알고리즘

- 1) 이미지에서 픽셀 추출
- 2) 픽셀에서 R,G,B 값 추출 -EX) GetRValue
- 3) 영상의 R,G,B에서 밝기 값을 추출하는 행렬공식 사용해 픽셀들을 그레이화.

$$\text{Gray} = (\text{Red} * 0.2126 + \text{Green} * 0.7152 + \text{Blue} * 0.0722)$$

원본 이미지



필터(그레이
이)



이미지 처리 필터 - binary

바이너리 필터 = 흑 또는 백으로 처리하는 영상처리

바이너리 알고리즘

- 1) 영상을 그레이화 한 후 임계값(Threshold) 설정
- 2) 임계값(0~255)을 기준으로 낮은 값을 가진 픽셀은 검은색(0)으로 바꾸어 저장, 높은 값은 흰색(255) 으로 바꾸어 저장

원본 이미지



필터(바이너리)



이미지 처리 필터 - Sobel

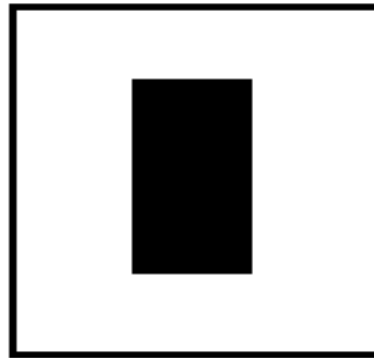
소벨 필터 = 영상의 경계선을 보이게 처리하는 영상처리

-1	0	1
-2	0	2
-1	0	1

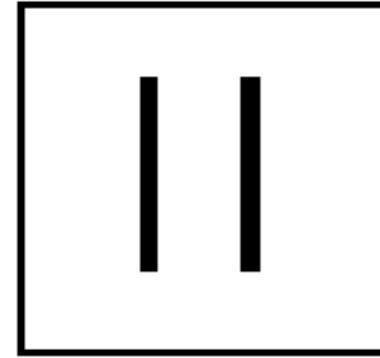
▲ 수직 마스크

1	2	1
0	0	0
-1	-2	-1

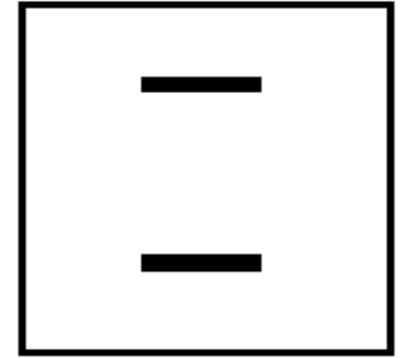
▲ 수평 마스크



원본 이미지



수직 마스크만 적용



수평 마스크만 적용

이미지 처리 필터 - Sobel

소벨 필터 = 영상의 경계선을 보이게 처리하는 영상처리

-1	0	1
-2	0	2
-1	0	1

▲ 수직 마스크

1	2	1
0	0	0
-1	-2	-1

▲ 수평 마스크

그레이 이미지 pixel = gray[h][w] 배열에 저장
h = 이미지 높이 w = 이미지 너비

계산법 (for문을 통해서 반복 진행 => 전체 이미지 다 돌때까지)

수직 마스크 = $(-1 * \text{gray}[h+0][w+0]) + (0 * \text{gray}[h+0][w+1]) + (1 * \text{gray}[h+0][w+2]) + \dots$

수평 마스크 = $(1 * \text{gray}[h+0][w+0]) + (2 * \text{gray}[h+0][w+1]) + (1 * \text{gray}[h+0][w+2]) + \dots$

이미지 처리 필터 - Sobel

소벨 필터 = 영상의 경계선을 보이게 처리하는 영상처리

소벨 알고리즘

- 1) 이미지를 그레이화 후 수직 마스크와 수평 마스크를 연산하여 모든 방향의 윤곽선을 추출.
- 2) `abs()` 함수를 이용하여 절대값을 각각 구해서 더해줍니다.
- 3) 임계값(0~255)을 기준으로 기준보다 높은 값을 가진 픽셀은 흑(0), 낮은 값은 백(255)으로 설정

원본 이미지



필터(소벨)



설정 - 임계값

바이너리 필터

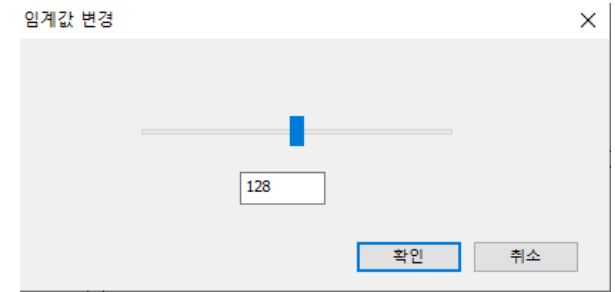
변경



초기



변경



다이얼로그를 통해서 사용자가
임의로 임계값을 설정

설정 - 임계값

소벨 필터

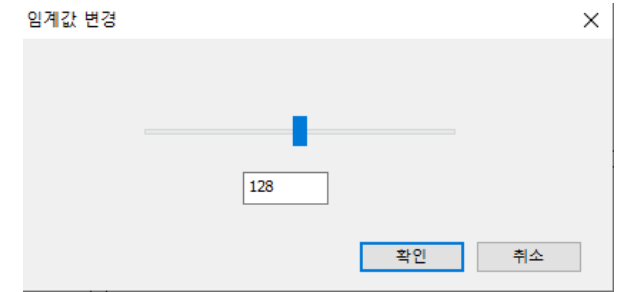
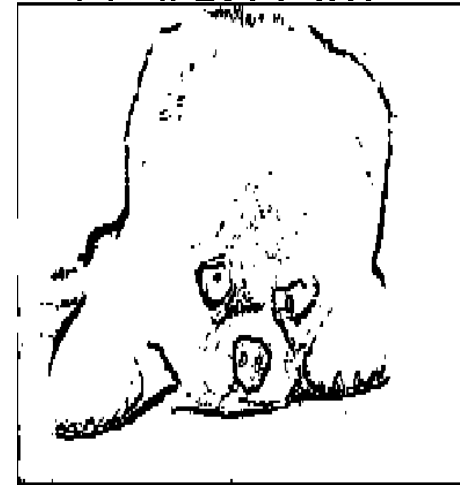
변경
임계값(50)



초기
임계값(128)

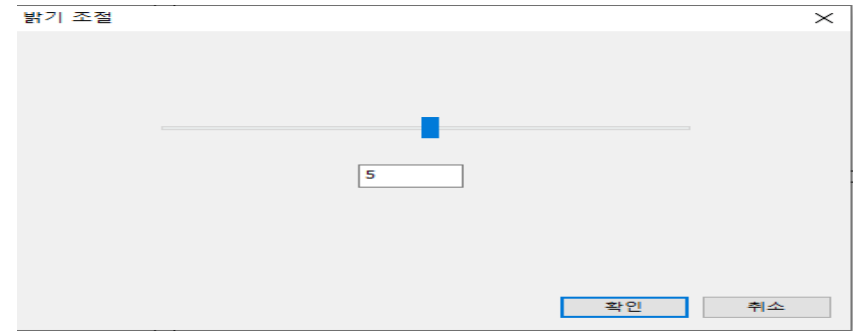


변경
임계값(150)



임계값은 바이너리 필터와 소벨 필터에서만 영향을 준다.

설정 - 밝기 조절



해당하는 이미지의 RGB값을 받고 R,G,B에 밝기 변경값 만큼을 더해서 출력



변경 상태(-70)



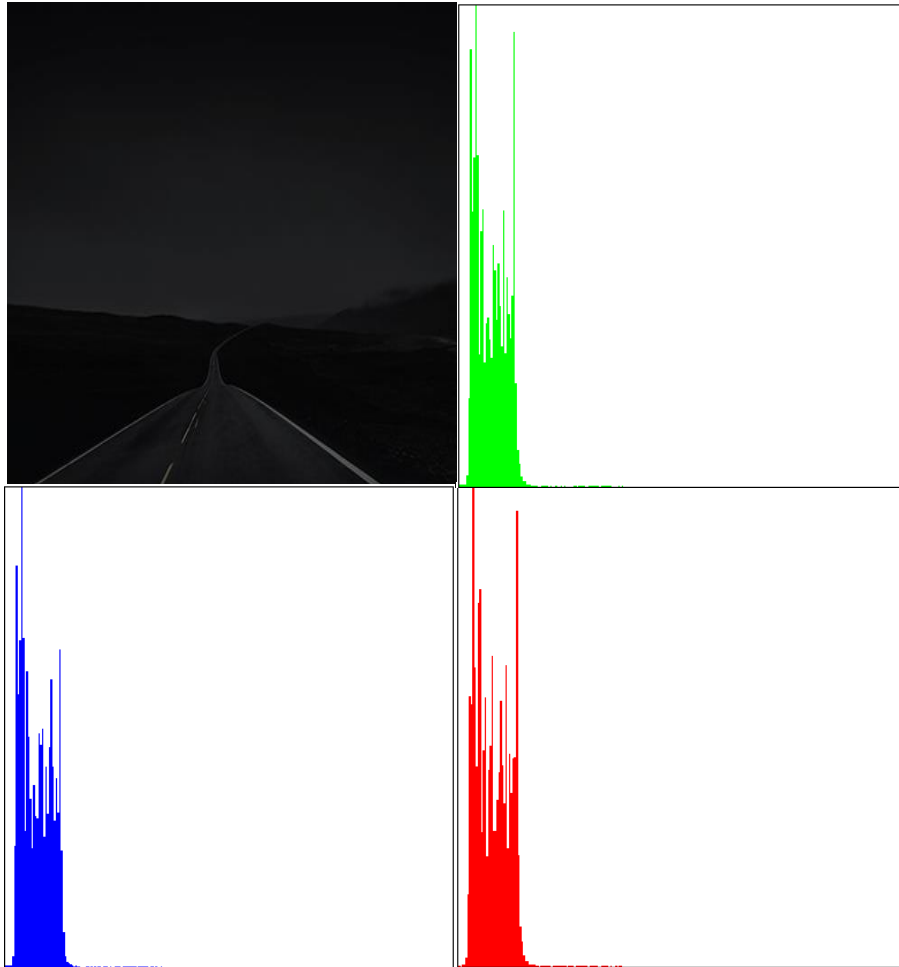
처음 상태(5)



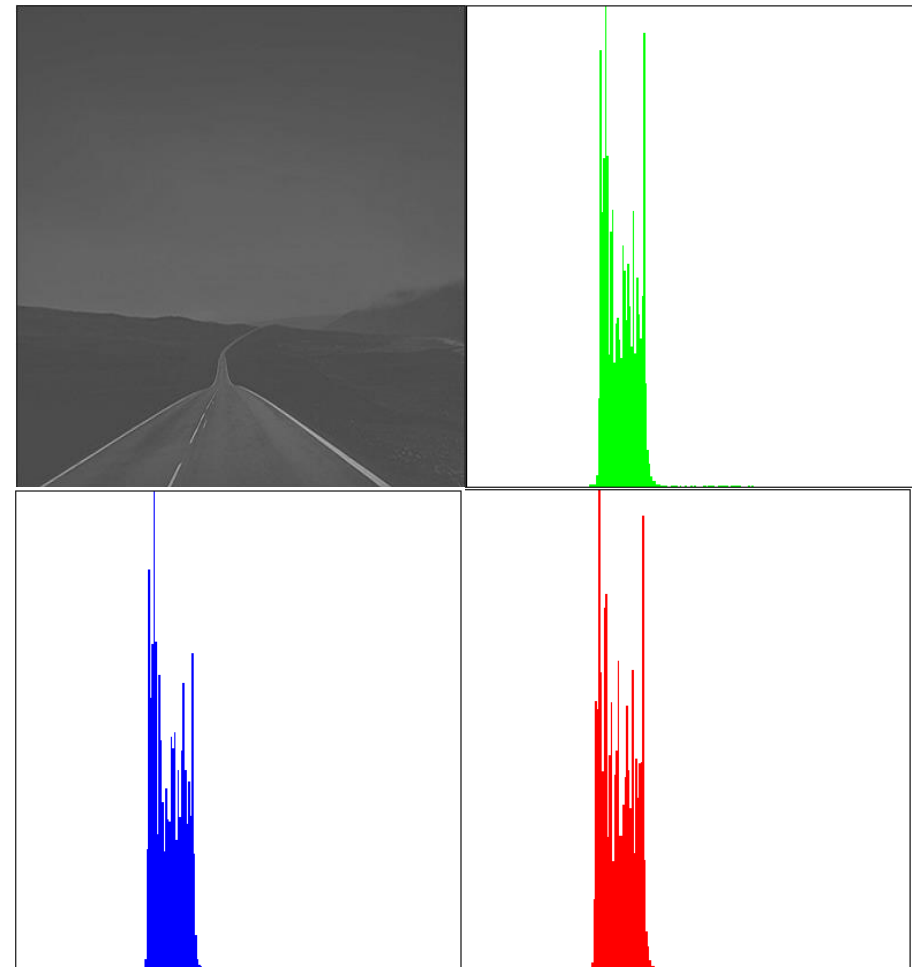
변경 상태(+70)

설정 - 밝기 조절

처음 상태(5)



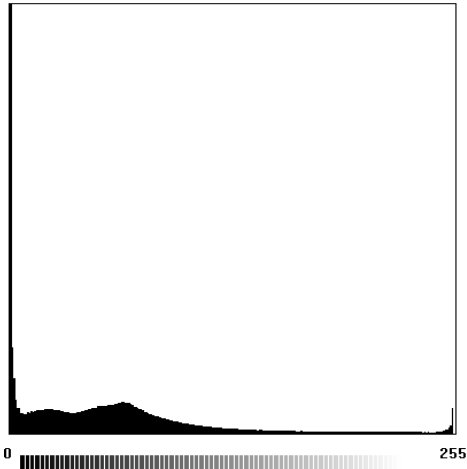
변경 상태(+70)



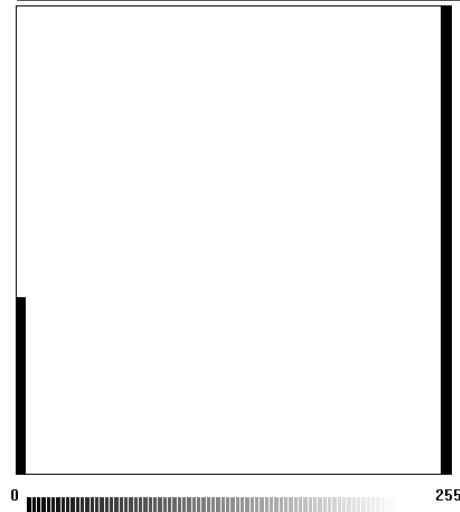
기타 기능 - 히스토그램

히스토그램 - 영상의 밝기 값의 분포를 보여주는 그래프

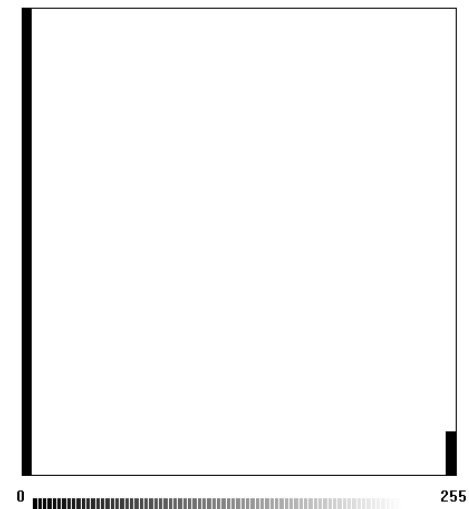
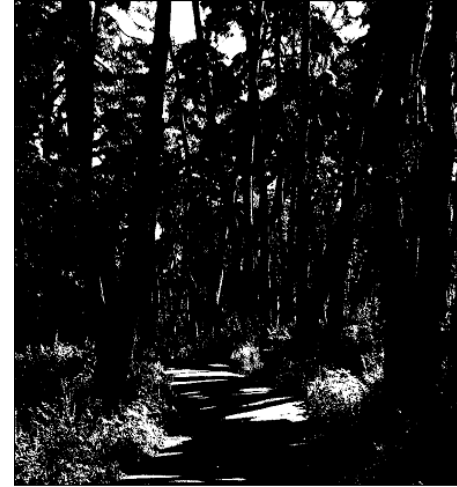
필터(그레이



필터(소벨)



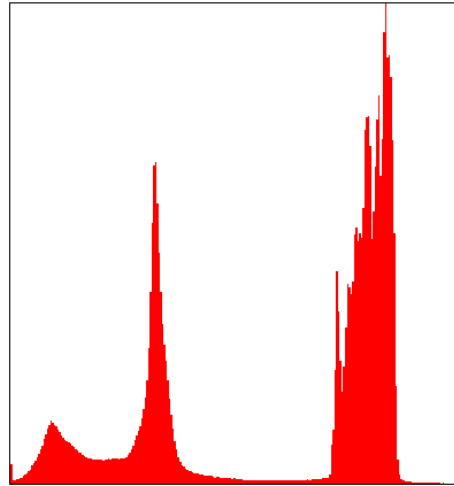
필터(바이



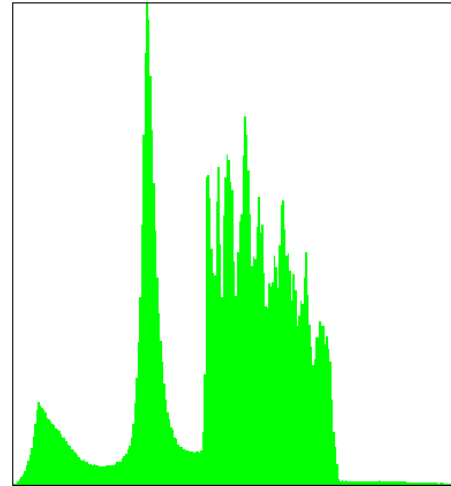
기타기능 - 히스토그램(RGB)



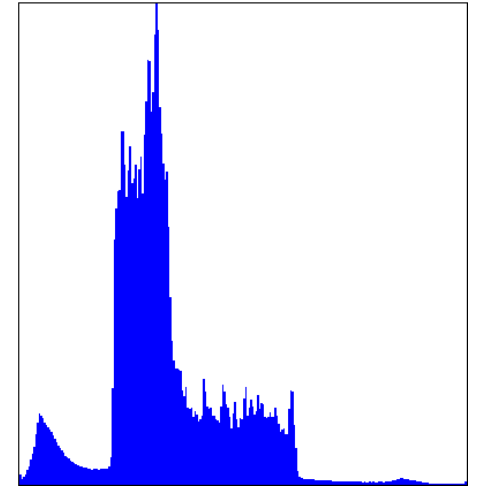
이미지의 픽셀에서 GetR(G,B)Value()함수를 통해서 RGB값을 받고 이를 그래프로 표현



R-히스토그램



G-히스토그램

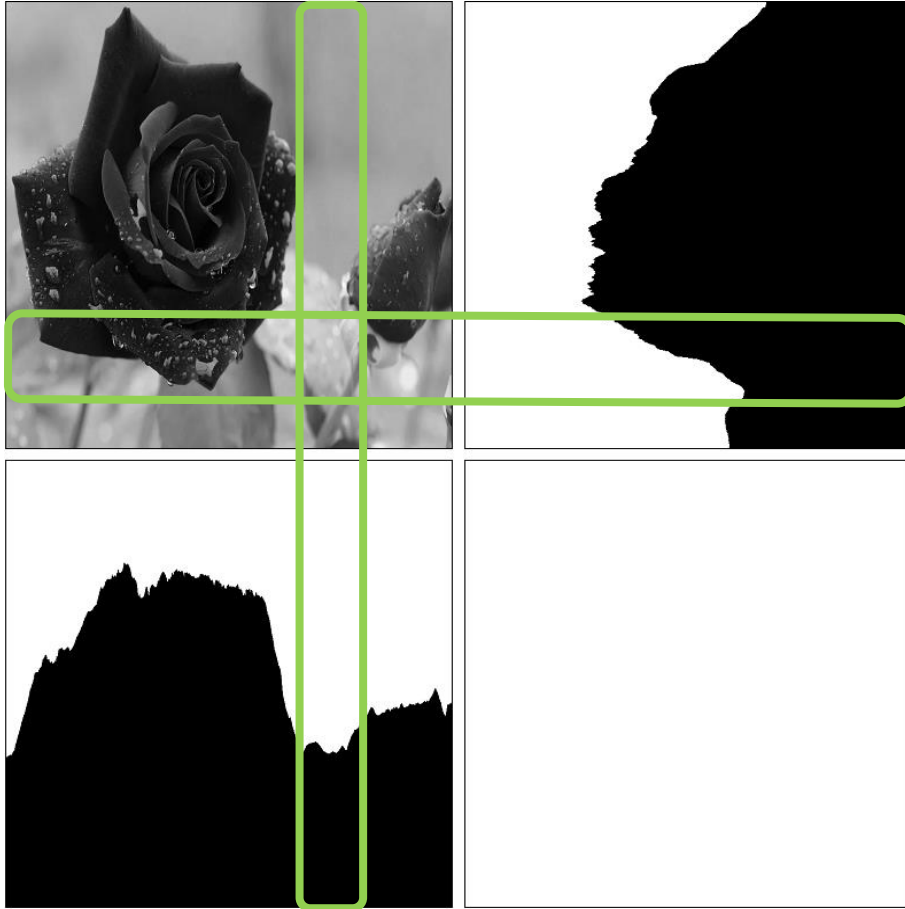


B-히스토그램

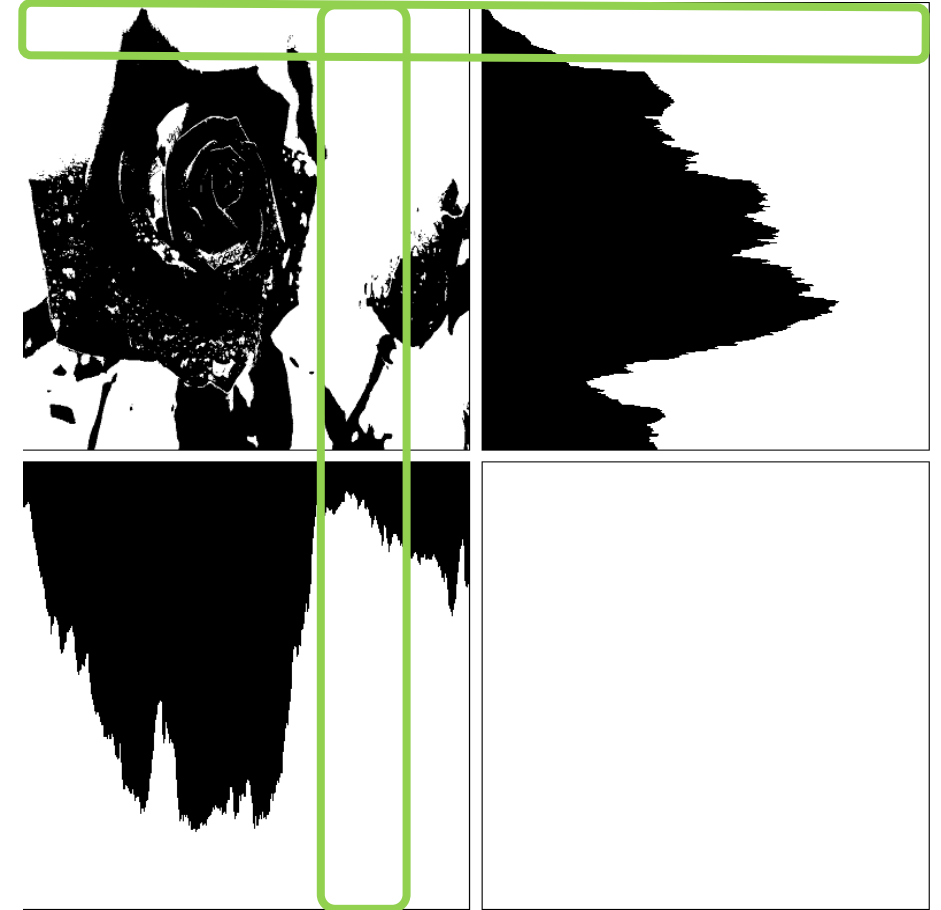
기타기능 - 프로젝션

프로젝션 - 해당 이미지의 픽셀 개수 or 밝기 값을 더해서 그래프로 표현

필터(그레이) - 밝기 값



필터(바이너리) - 픽셀



기타 기능 - 스트레칭

스트레칭 - 특정 밝기 영역에 영상 픽셀의 밝기 값이 집중되어 있는 것을 퍼트리하는 방법

$$new\ pixel = \frac{old\ pixel - low}{high - low} \times 255$$

high = 히스토그램 끝 위치

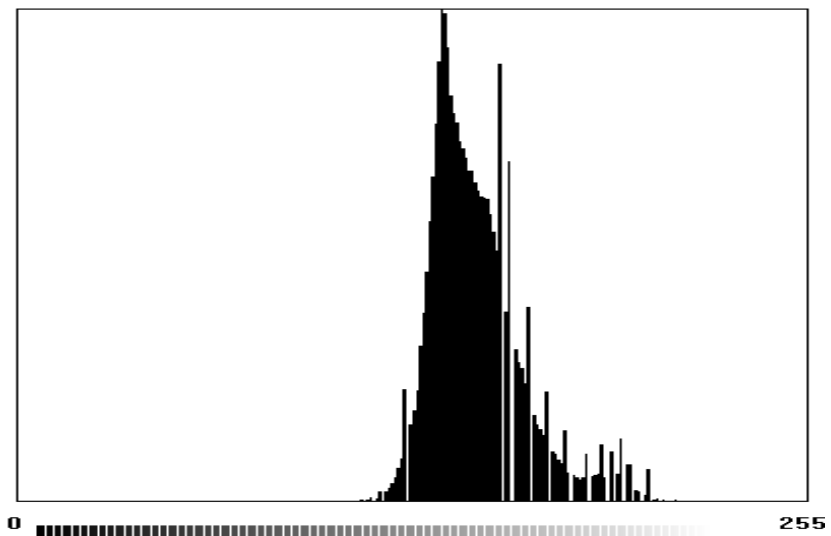
low = 히스토그램 시작 위치

old pixel = 기존 gray 이미지 pixel

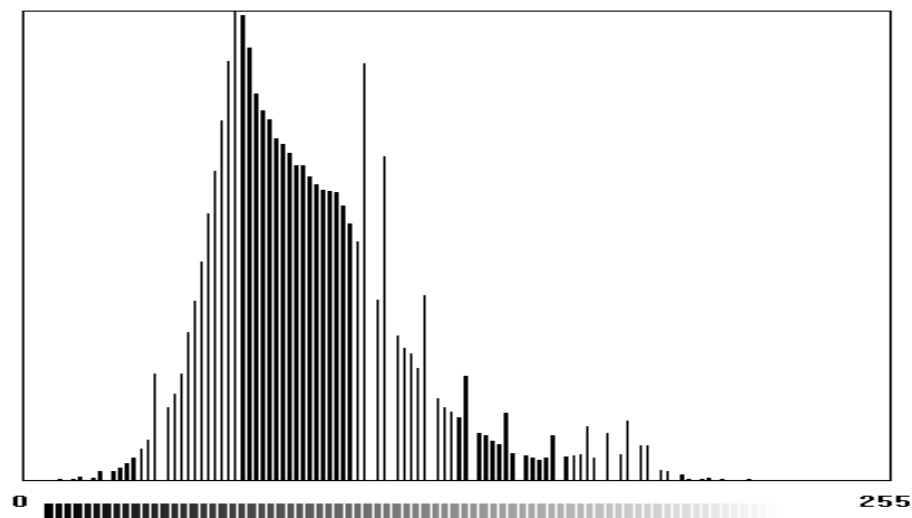
해당 공식을 통해서 gray 이미지를 다시 그리고,
히스토그램 또한 재설정을 한다.

기타 기능 - 스트레칭

스트레칭 전



스트레칭 후



기타 기능 - 평활화

평활화 - 영상의 밝기 분포를 재분해하여서 명암 대비 ~~최대~~ ^{적당}정규화 된 누적합은 반올림을 진행 해주어야 한다.

평활화 과정

- 1) 히스토그램 그리고 배열에 저장
- 2) 히스토그램 누적 빈도수 계산
- 3) 누적 빈도수를 통해 정규화된 누적합 계산
- 4) 정규화 된 누적합으로 이미지, 히스토그램 그리기

$$sum[i] = \sum_{j=0}^i hist[j]$$

누적 빈도수 계산

```
sum[0] = hist[0]
sum[1] = hist[0]+hist[1]
sum[2] = hist[0]+hist[1]+hist[2]
```

$$n[i] = sum[i] \times \frac{1}{N} \times I_{\max}$$

누적 빈도수 정규화

(반올림 필요)

```
n[0] = sum[0]/이미지크기*255
n[1] = sum[1]/이미지 크기*255
n[2] = sum[2]/이미지 크기*255
```

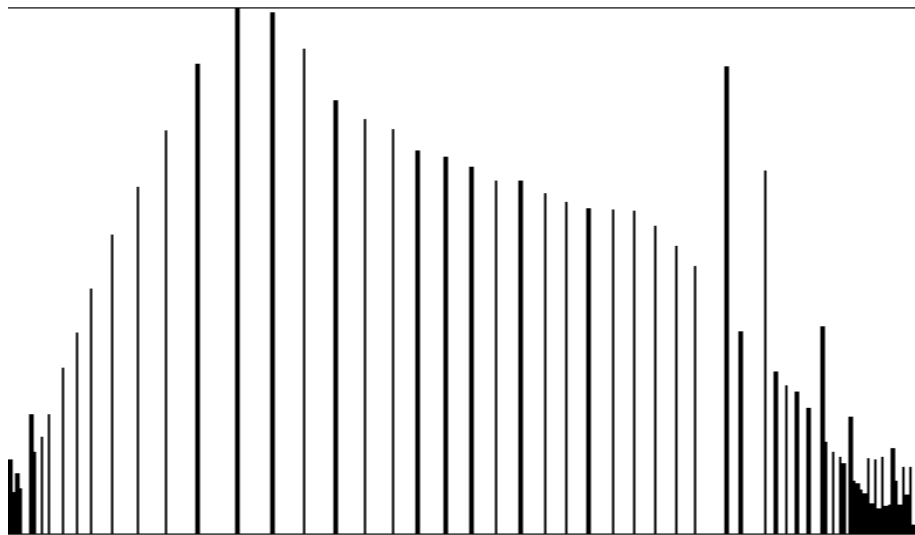
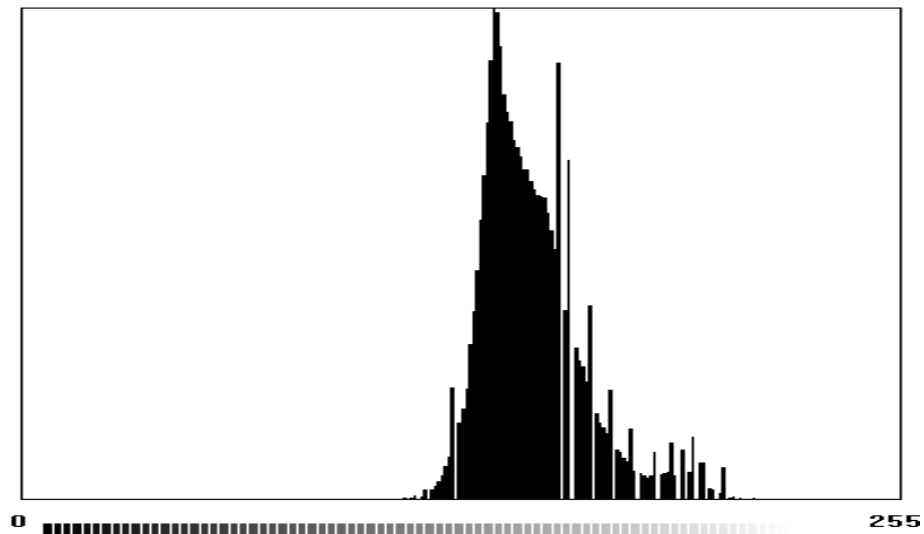
N = 이미지 크기
I = 255(히스토그램)

기타 기능 - 평활화

평활화 전



평활화 후



Thank You

