

4 Point Processing

소프트웨어융합 김태민

- 1) 연습문제 2, 3 (91~92쪽, blocks.tif 영상 사용)

연습문제 2

blocks 영상 **b**에 대해서 다음의 명령들을 수행하라.

```
>> b2=imdivide(b,64);
```

```
>> bb2=immultiply(b2,64);
```

```
>> imshow(bb2)
```

그 결과에 대해 간략하게 설명해라. 왜 이 결과는 원 영상과 같지 않은가?

영상의 크기는 같지만 품질이 원본 영상과 같지 않다.
영상의 픽셀들을 64로 나눌때 나머지 값들이 소실되면서 영상의 세밀한 명암 정보가 감소하고, 결과적으로 이미지의 품질이 저하된다. (양자화 소실)

특히 이러한 현상은 세밀한 그라데이션이 필요한 부분에서 두드러지게 나타나며 명암 대비가 줄어들어 이미지가 흐릿해지거나 블록 현상이 나타난다.

255를 64로 나누어 0,1,2,3,4의 영역으로 나뉘고, 기본 데이터에서 나머지 값들이 소실된 데이터를 다시 64로 곱하기 때문에 원본과 달라진다.



연습문제 3

문제 2에서의 값 64를 32와 16으로 치환하여 수행하라. 문제 2에 있는 결과와는 어떻게 다른가?

```
>> b = imread('blocks.tif');
>> b1 = imdivide(b, 64);
>> bb1 = immultiply(b1, 64);
>> b2 = imdivide(b, 32);
>> bb2 = immultiply(b2, 32);
>>
```

```
>> b3 = imdivide(b, 18);
>> bb3 = immultiply(b3, 18);
>> subplot(2,2,1), imshow(b), title('원본')
```

경고: 도킹된 Figure에서 이미지의 초기 배율이 'fit'으로 설정되었습니다.

> imshow (322번 라인)

```
>> subplot(2,2,2), imshow(bb1), title('64')
```

경고: 도킹된 Figure에서 이미지의 초기 배율이 'fit'으로 설정되었습니다.

> imshow (322번 라인)

```
>> subplot(2,2,3), imshow(bb2), title('32')
```

경고: 도킹된 Figure에서 이미지의 초기 배율이 'fit'으로 설정되었습니다.

> imshow (322번 라인)

```
>> subplot(2,2,4), imshow(bb3), title('18')
```

경고: 도킹된 Figure에서 이미지의 초기 배율이 'fit'으로 설정되었습니다.

> imshow (322번 라인)

fx>>

나눗셈 단위가 작을수록 더 적은 분할을 시행해
그레이스케일 레벨이 유지가 되어 원본에
가까운 품질이 유지가 된다.

각 단계에서 소실되는 정보의 양을 최소화
하려면 적절한 양자화 수준을 선택하는것이
중요하다.

원본



64



32



18



- **2) 카메라맨 영상을 인덱스 영상으로 변환하라.(연습문제 10 참조)**

- 원영상, 인덱스영상의 인덱스를 영상으로 디스플레이
- 위 두 영상의 히스토그램 디스플레이
- 히스토그램 스트레칭 알고리즘을 적용하여 영상을 개선하시오.
 - 인덱스 영상의 히스토그램을 보고 2개의 적절한 스트레칭 함수를 만들고 디스플레이 하세요.
 - 이 스트레칭 함수를 사용하여 히스토그램 스트레칭을 구현하라.
 - 결과 영상과 히스토그램을 디스플레이하고 비교 설명하라.


```
>> original=imread('cameraman.tif');
>> [index, map] = gray2ind(original, 256);
>> subplot(2, 2, 1);
>> imshow(original);
경고: 도킹된 Figure에서 이미지의 초기
설정되었습니다.
> imshow (322번 라인)
>> title('원본');
>> subplot(2, 2, 2);
>> imshow(index, map);
경고: 도킹된 Figure에서 이미지의 초기
설정되었습니다.
> imshow (322번 라인)
>> title('인덱스영상');
>>
>> subplot(2, 2, 3);
>> imhist(original);
>> title('원본 히스토그램');
>> subplot(2, 2, 4);
>> imhist(index, 256); %256색
>> title('인덱스 영상 히스토그램');
```

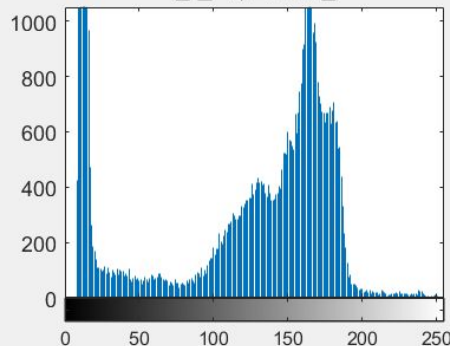
원본



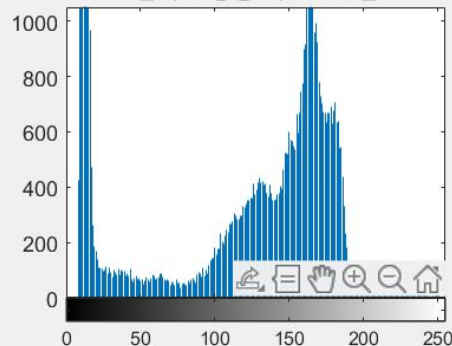
인덱스영상



원본 히스토그램



인덱스 영상 히스토그램



원본



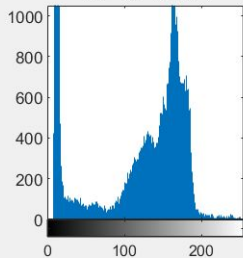
스트레칭 이미지



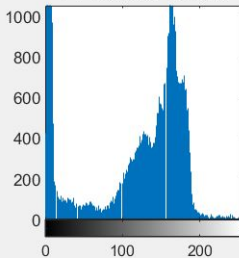
적절히 조정된 스트레칭 이미지



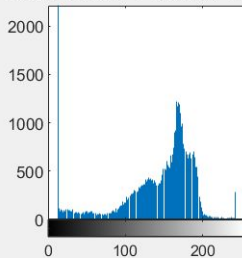
원본 히스토그램



스트레칭 히스토그램



적절히 조정된 스트레칭 히스토그램



원본 이미지 : 초기 그레이스케일 이미지
 스트레칭된 이미지 : 이미지 밝기 범위가 0에서 255로 확장돼 이미지의 대비가 크게 향상,
 시각적 품질 개선
 조정된 스트레칭 이미지 : 밝기를 더 보수적으로
 조정, 노이즈 영향을 줄이고 자연스러운 대비
 제공

```
>> b = imread('cameraman.tif');
```

```
>> subplot(2,3,1);
```

```
>> imshow(b);
```

경고: 도킹된 Figure에서 이미지의 초기 배율이 'fit'으로 설정되었습니다

```
> imshow (322번 라인)
```

```
>> title('원본');
```

```
>> subplot(2,3,4);
```

```
>> imhist(b);
```

```
>> title('원본 히스토그램');
```

```
>> b_min = double(min(b(:)));
```

```
>> b_max = double(max(b(:)));
```

```
>> b_stretched = imadjust(b, [b_min/255 b_max/255], [0 1]);
```

```
>> subplot(2,3,2);
```

```
>> imshow(b_stretched);
```

경고: 도킹된 Figure에서 이미지의 초기 배율이 'fit'으로 설정되었습니다

```
> imshow (322번 라인)
```

```
>> title('스트레칭 이미지');
```

```
>> subplot(2,3,5);
```

```
>> imhist(b_stretched);
```

```
>> title('스트레칭 히스토그램');
```

```
>> b_adjusted_stretched = imadjust(b, [0.1 0.9], [0.05 0.95]);
```

```
>> subplot(2,3,3);
```

```
>> imshow(b_adjusted_stretched);
```

경고: 도킹된 Figure에서 이미지의 초기 배율이 'fit'으로 설정되었습니다

```
> imshow (322번 라인)
```

```
>> title('적절히 조정된 스트레칭 히스토그램');
```

```
>> subplot(2,3,6);
```

```
>> imhist(b_adjusted_stretched);
```

```
>> title('적절히 조정된 스트레칭 히스토그램');
```

- 3) 연습문제7

- 히스토그램 평활화 과정을 손으로 풀어라.
- 위 내용을 프로그래밍 하여 결과를 비교하라

Gray Level	n_i	Σn_i	$\frac{1}{64}\Sigma n_i(L-1)$	Round Value
0	0	0	0.00	0
1	0	0	0.00	0
2	1	1	0.30	0
3	2	3	0.89	1
4	2	5	1.48	1
5	2	7	2.08	2
6	2	9	2.67	3
7	3	12	3.56	4
8	5	17	5.05	5
9	3	20	5.94	6
10	5	25	7.42	7
11	4	29	8.61	9
12	7	36	10.69	11
13	5	41	12.17	12
14	8	49	14.55	15
15	3	52	15.44	15
16	4	56	16.63	17
17	2	58	17.22	17
18	3	61	18.11	18
19	3	64	19.00	19

원본 8x8 그레이 격자

```

12 6 5 13 14 14 16 15
11 10 8 5 8 11 14 14
9 8 3 4 7 12 18 19
10 7 4 2 10 12 13 17
16 9 13 13 16 19 19 17
12 10 14 15 18 18 16 14
11 8 10 12 14 13 14 15
8 6 3 7 9 11 12 12

```

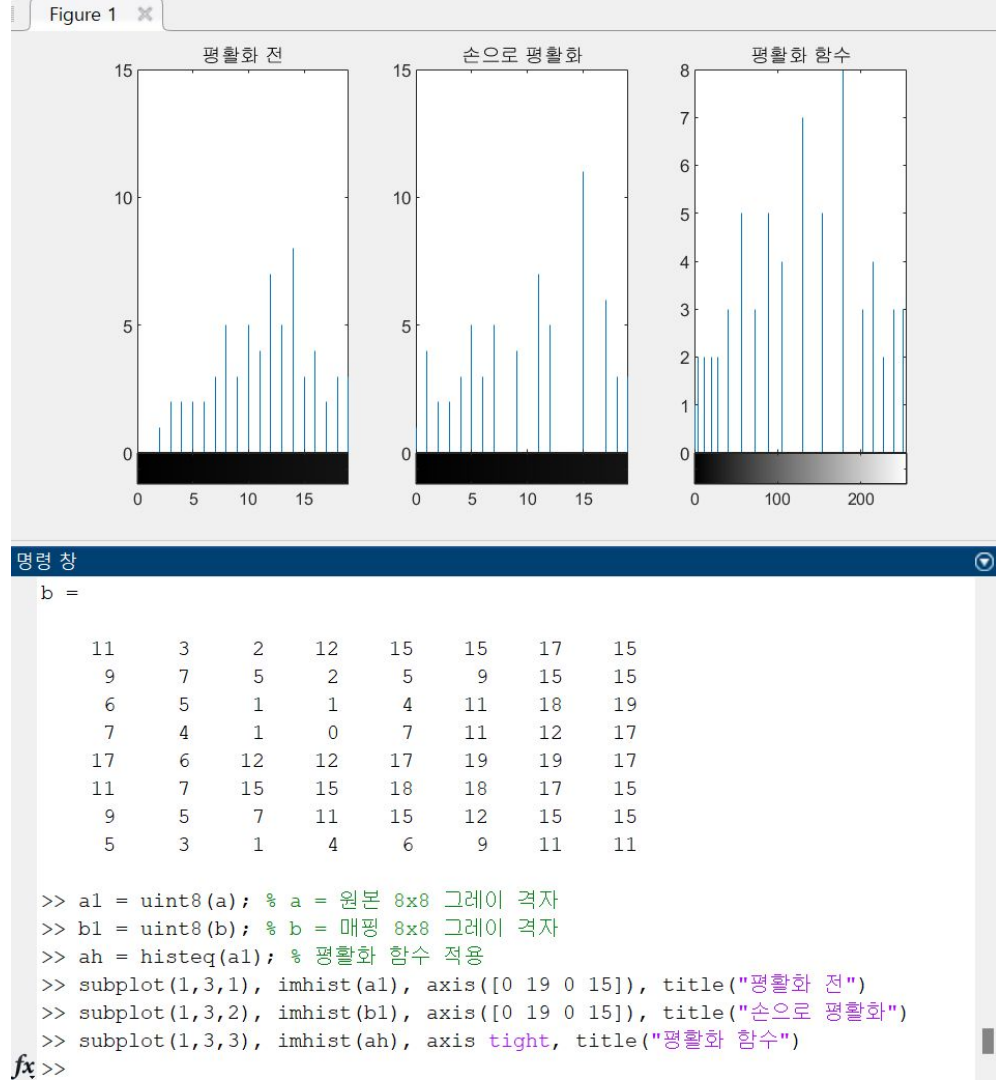
매핑 8x8 그레이 격자

```

11 3 2 12 15 15 17 15
9 7 5 2 5 9 15 15
6 5 1 1 4 11 18 19
7 4 1 0 7 11 12 17
17 6 12 12 17 19 19 17
11 7 15 15 18 18 17 15
9 5 7 11 15 12 15 15
5 3 1 4 6 9 11 11

```

함수로 평활화시 손으로 평활화 한것과 비슷한
히스토그램 빈도수가 나옴.
함수 사용시 ~19 에서 ~255로 변환됨

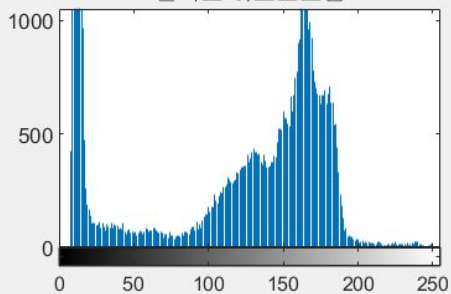


- 4) P4-1에서 변환한 카메라맨 인덱스 영상을 입력으로 사용하여 히스토그램 평활화 알고리즘을 실행하는 프로그램을 작성하라.
 - P4-1에서의 히스토그램 스트레칭 결과와 비교 평가하라.

인덱스



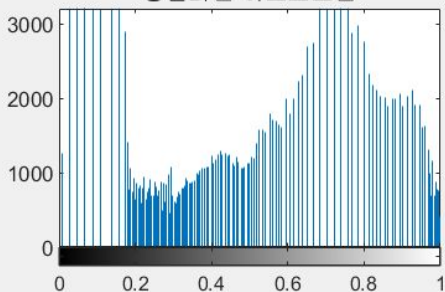
인덱스 히스토그램



평활화된 이미지



평활화된 히스토그램



```
>> original = imread('cameraman.tif');
>> [indexed, map] = gray2ind(original, 256);
>> subplot(2, 2, 1);
>> imshow(indexed, map);
경고: 도킹된 Figure에서 이미지의 초기 배열이 'fit'으로 설정되었습니다
> imshow (322번 라인)
>> title('인덱스');
>> subplot(2, 2, 2);
>> imhist(indexed, 256);
>> title('인덱스 히스토그램');
>>
>> equalized_image = histogram_equalization(indexed, map);
>> subplot(2, 2, 3);
>> imshow(equalized_image);
경고: 도킹된 Figure에서 이미지의 초기 배열이 'fit'으로 설정되었습니다
> imshow (322번 라인)
>> title('평활화된 이미지');
>> subplot(2, 2, 4);
>> imhist(equalized_image, 256);
>> title('평활화된 히스토그램');
```

```
histogram_equalization.m * +
1 function equalized_image = histogram_equalization(indexed, map)
2 % 인덱스 영상과 컬러맵을 받아서 히스토그램 평활화를 수행함
3
4 % 원본 히스토그램 계산
5 original_hist = imhist(indexed, size(map, 1));
6
7 % 누적 분포 함수 (CDF) 계산
8 cdf = cumsum(original_hist) / numel(indexed);
9
10 % CDF를 [0, 1] 범위로 스케일링
11 cdf_scaled = uint8(255 * cdf);
12
13 % 새로운 인덱스 영상 생성
14 equalized_indexed = cdf_scaled(indexed(:) + 1);
15
16 % 매핑된 색상을 원래 형태로 되돌림
17 equalized_image = reshape(equalized_indexed, size(indexed));
18 equalized_image = ind2rgb(equalized_image, map);
19 end
```