

AI비전응용 PBL (2주차)

이미지 처리

Prof. Seung Ho Lee

이미지 디지털화

- 이미지 신호의 디지털화 과정 요약



Sampling ⇨ Quantizing ⇨ Coding

샘플링

양자화

코딩(압축)

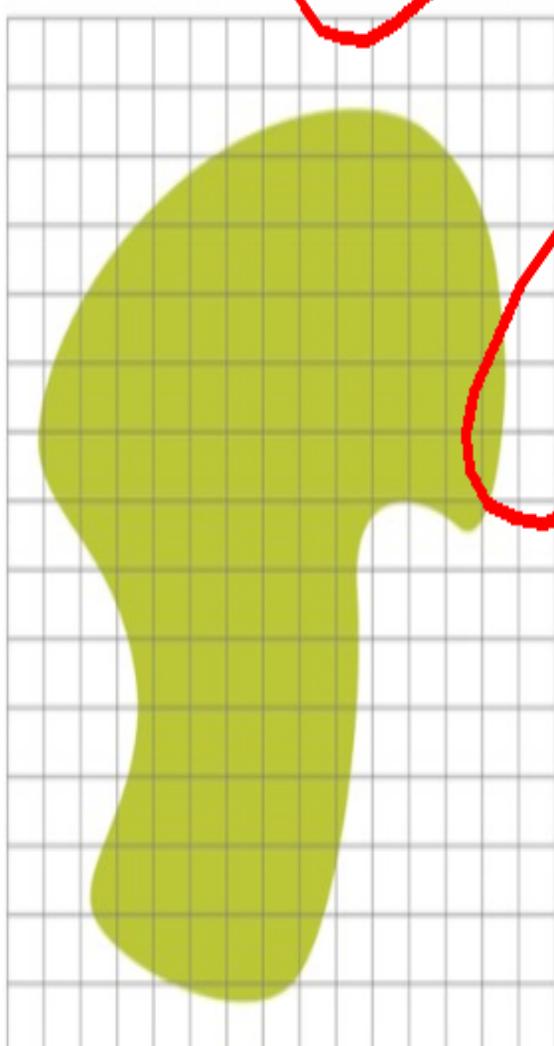
그림 출처 :

<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

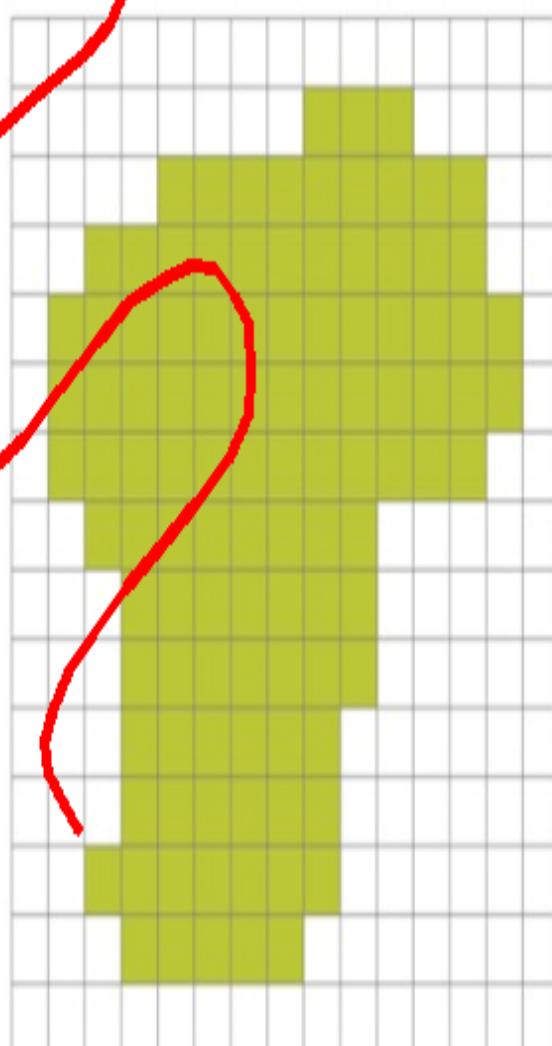
이미지 디지털화 (샘플링)

■ 샘플링

- 자연적으로 존재하는 장면을 격자에 맞추어 표현하는 과정



실제 객체의 모습



샘플링 결과

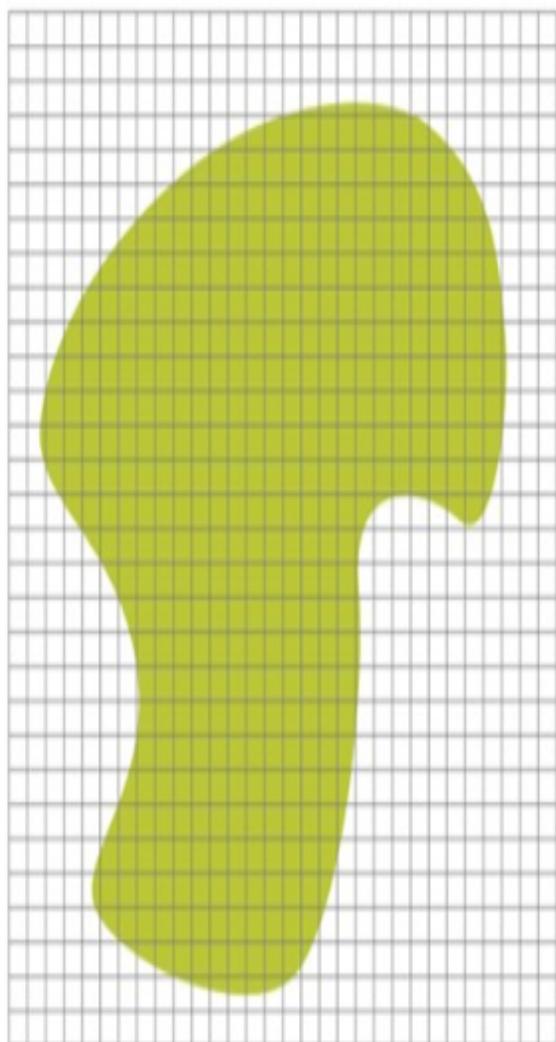
그림 출처 :

<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

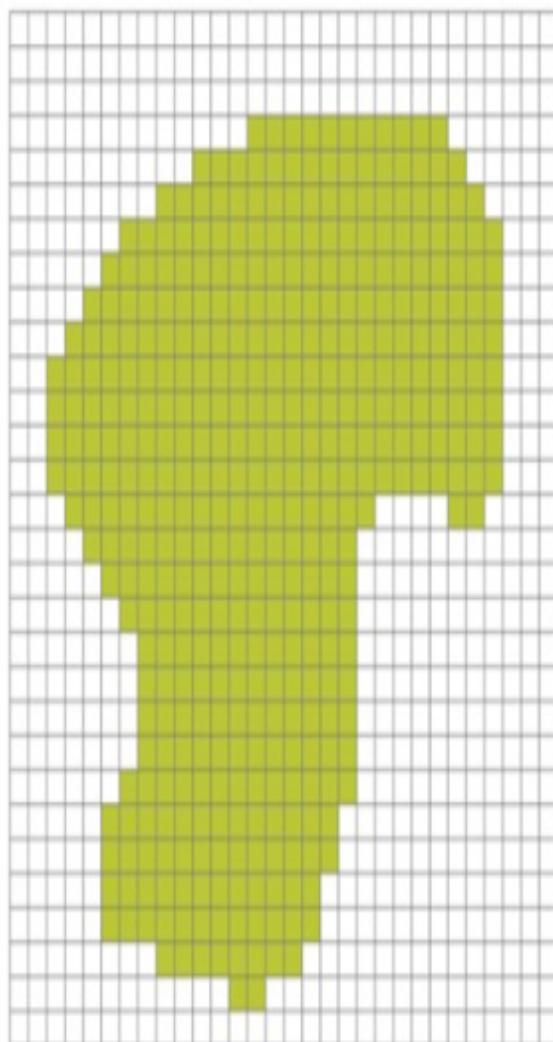
이미지 디지털화 (샘플링)

■ 샘플링

- 격자 크기를 더 줄일 경우 실물과 좀 더 가깝게 표현 가능



실제 객체의 모습



샘플링 결과

그림 출처 :

<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

이미지 디지털화 (샘플링)

■ 픽셀 수가 많을수록

- 고해상도
- 격자가 작음(=더 촘촘함)
- 더 세밀한 표현 가능

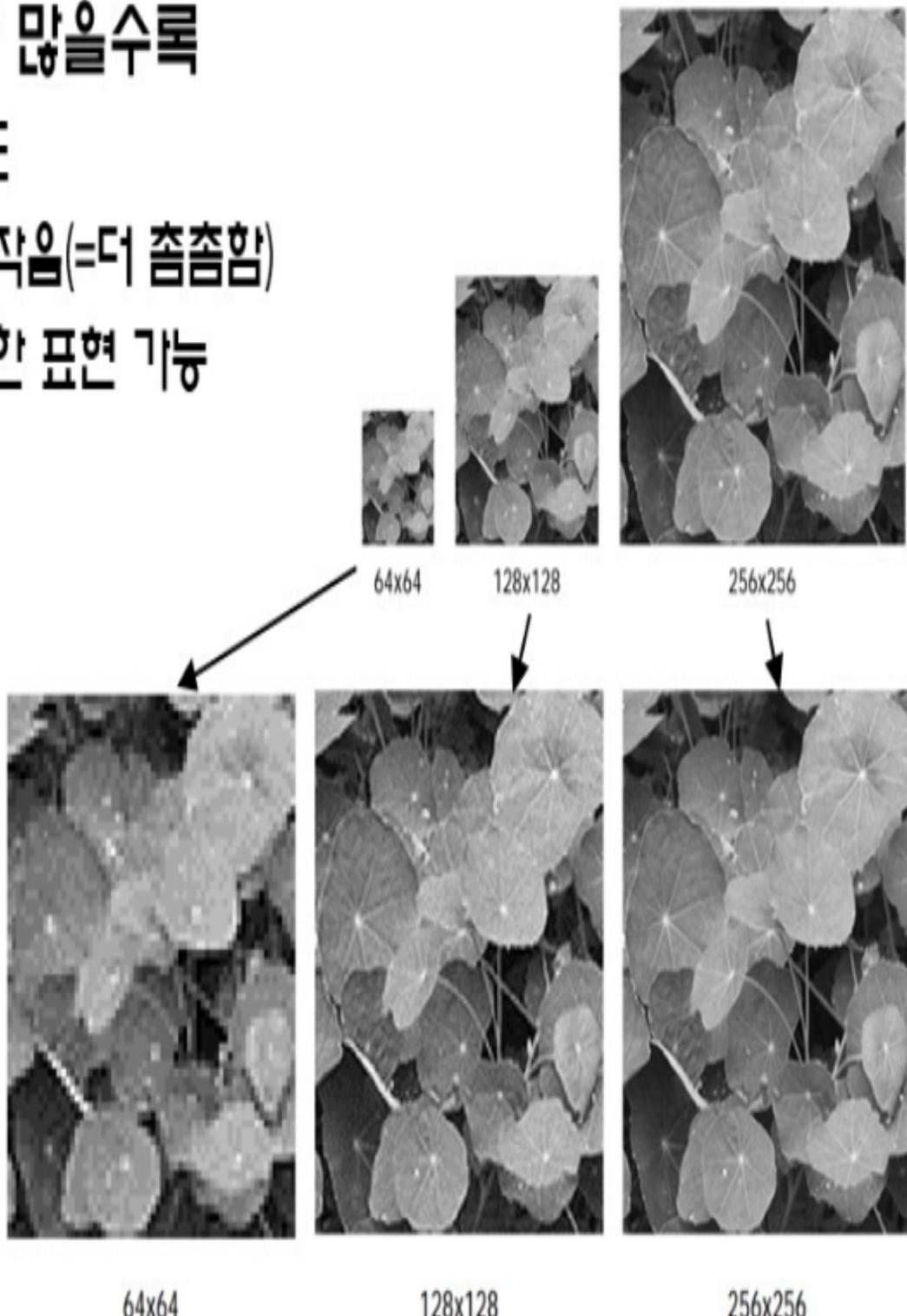


그림 출처 :

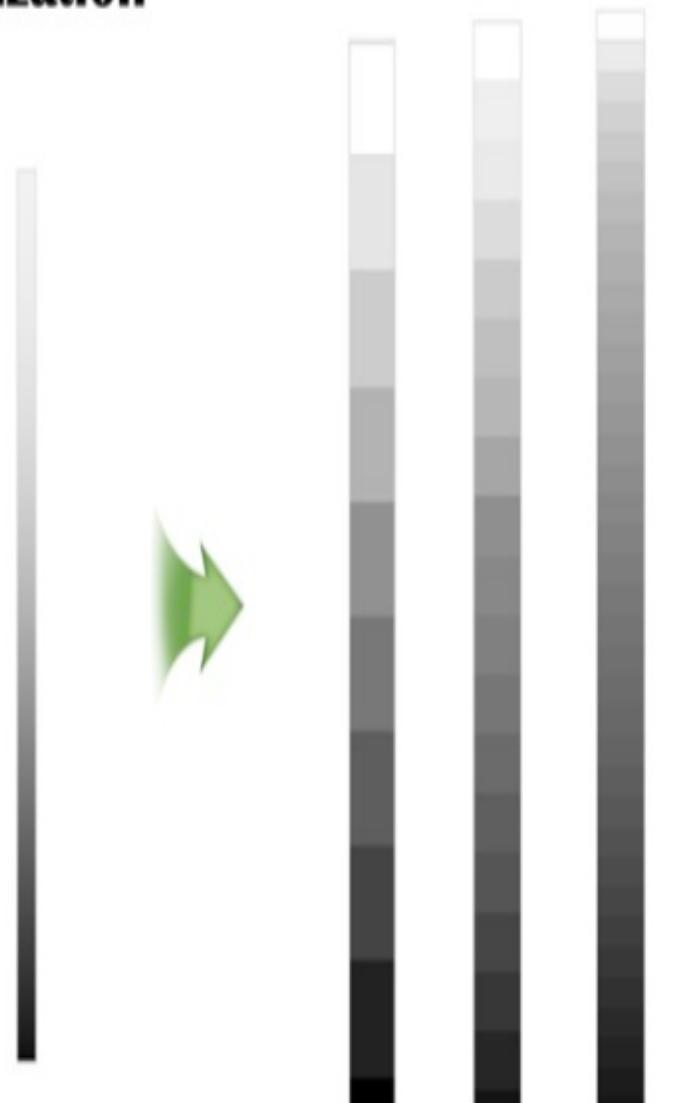
<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

이미지 디지털화 (양자화)

■ 양자화

- 유사한 값을 둉쳐서 하나의 대표값으로 표현하는 과정
- 비트 수에 따라 밝기의 단계수가 결정

Quantization



실제 밝기값

10 levels 20 levels 40 levels

양자화 된 값

그림 출처 :

<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

이미지 디지털화 (양자화)

- 양자화를 적용한 경우

- 비트 수가 적다면 블랙값의 세밀함이 떨어져 이질감이 존재

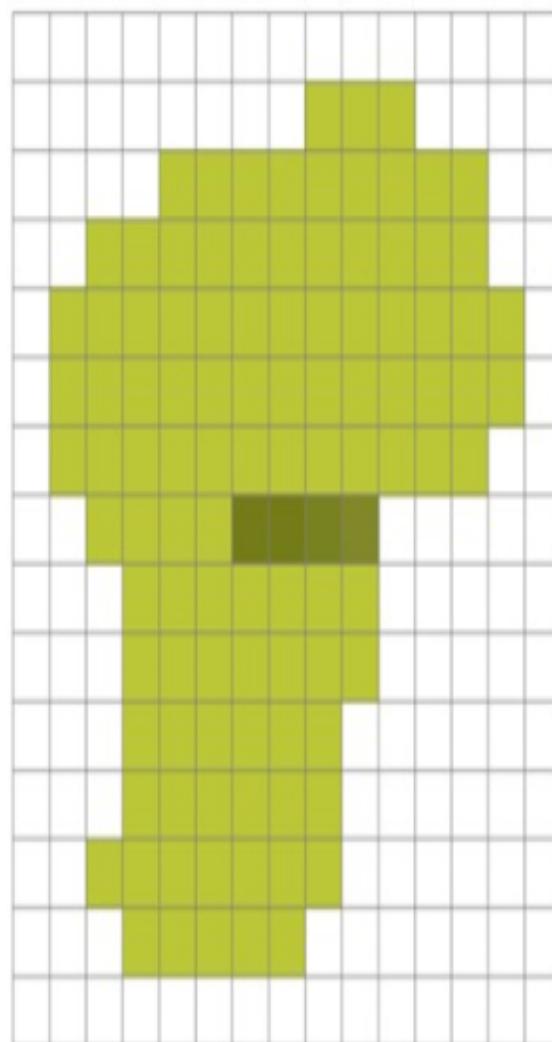
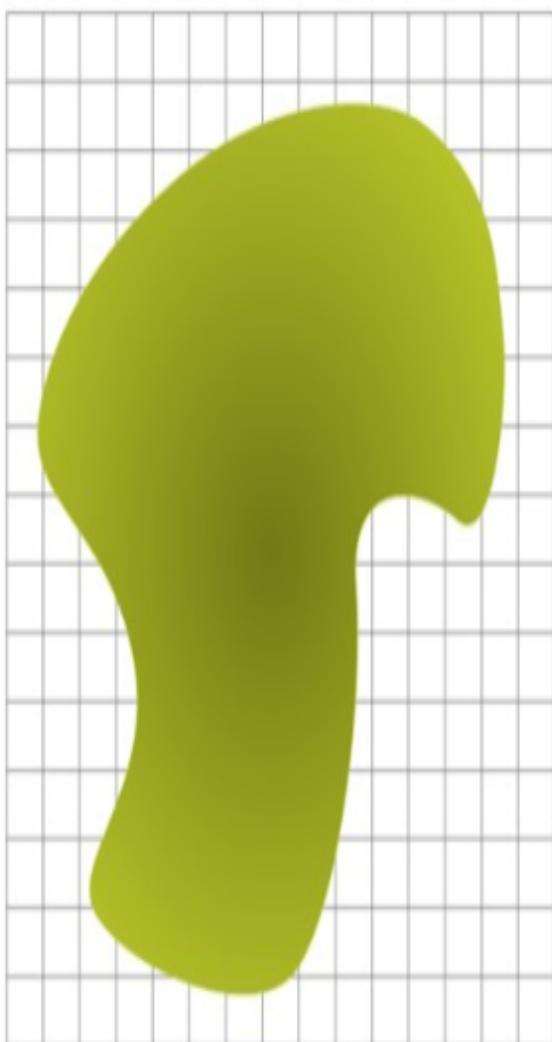


그림 출처 :

<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

이미지 디지털화 (양자화)

■ 양자화를 적용한 경우

- 1비트 \rightarrow 2단계 ($=2^1$)
- 2비트 \rightarrow 4단계 ($=2^2$)
- 4비트 \rightarrow 16단계 ($=2^4$)
- 8비트 \rightarrow 256단계 ($=2^8$)



(a) 2 levels



(b) 4 levels



(c) 16 levels



(d) 256 levels

그림 출처 :

<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

이미지 디지털화 (코딩)

■ 코딩

- 데이터를 파일로 저장하는 과정
- 보통 압축을 수행하여 파일 용량을 감소시킴

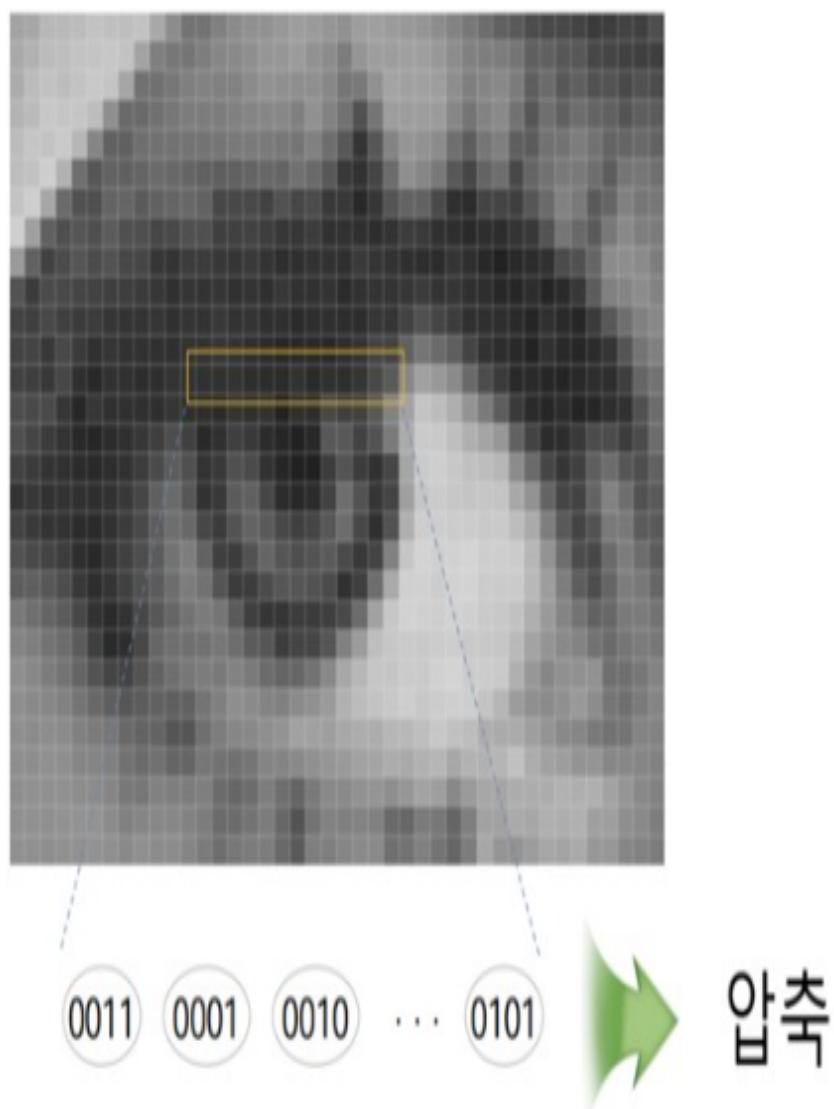
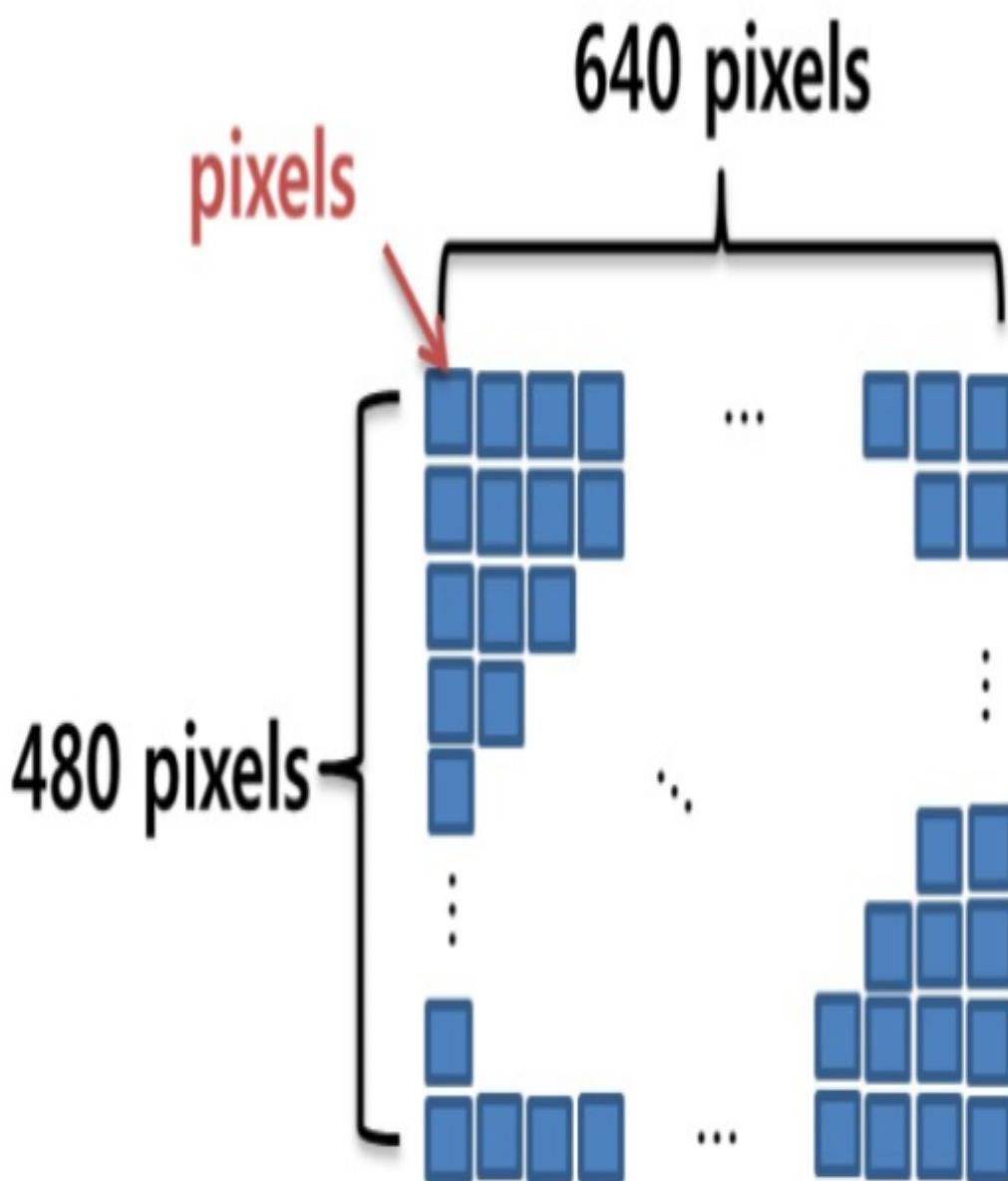


그림 출처 :

<http://www.kocw.net/home/search/kemView.do?kemId=1127905&ar=relateCourse.pdf>
(금오공대 김성영 교수님 공개 강의자료)

픽셀의 이해

- 이미지의 픽셀 크기



픽셀의 이해

■ 픽셀의 좌표와 밝기값

$$I(x, y)$$

where

x, y : spatial coordinates xy 좌표 = 픽셀 위치

I : intensity (gray level) 픽셀값

N: 가로 방향 픽셀 개수

M: 세로 방향 픽셀 개수

$$I(x, y) = \begin{bmatrix} I(0,0) & \dots & I(N-1,0) \\ \vdots & \ddots & \vdots \\ I(0, M-1) & \dots & I(N-1, M-1) \end{bmatrix}$$

그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/03.pdf>
(금오공대 김성영 교수님 공개 강의자료)

픽셀의 이해

■ 그레이스케일 이미지

- 보통 8비트(=256단계)를 사용
- 0: 검정색, 255: 흰색

147	146	148	150	153
145	149	151	154	156
149	152	153	156	157
150	153	155	157	158
149	151	152	156	159



그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/03.pdf>
(금오공대 김성영 교수님 공개 강의자료)

픽셀의 이해

- 그레이스케일 이미지
 - 이미지 일부를 확대한 모습

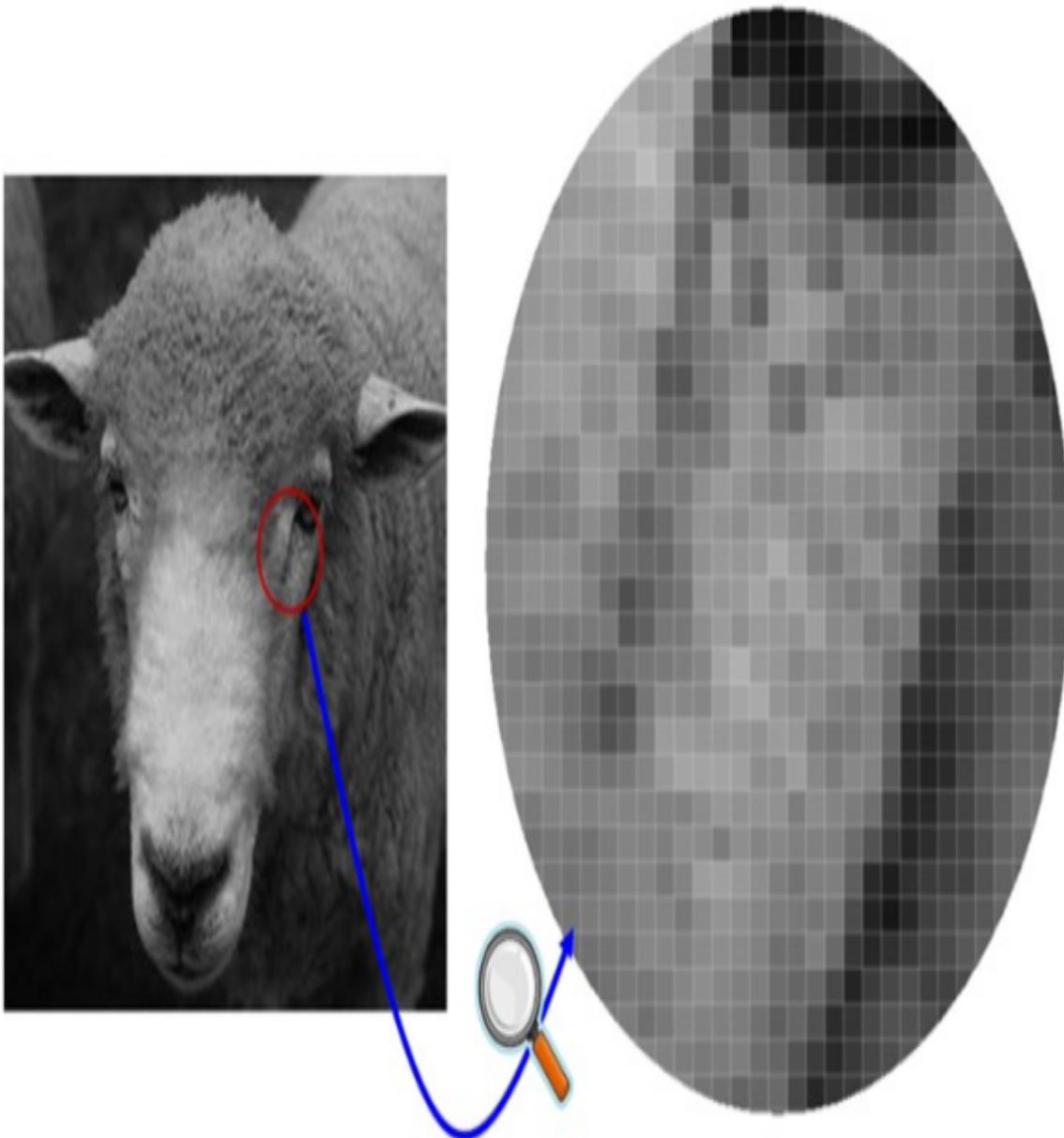
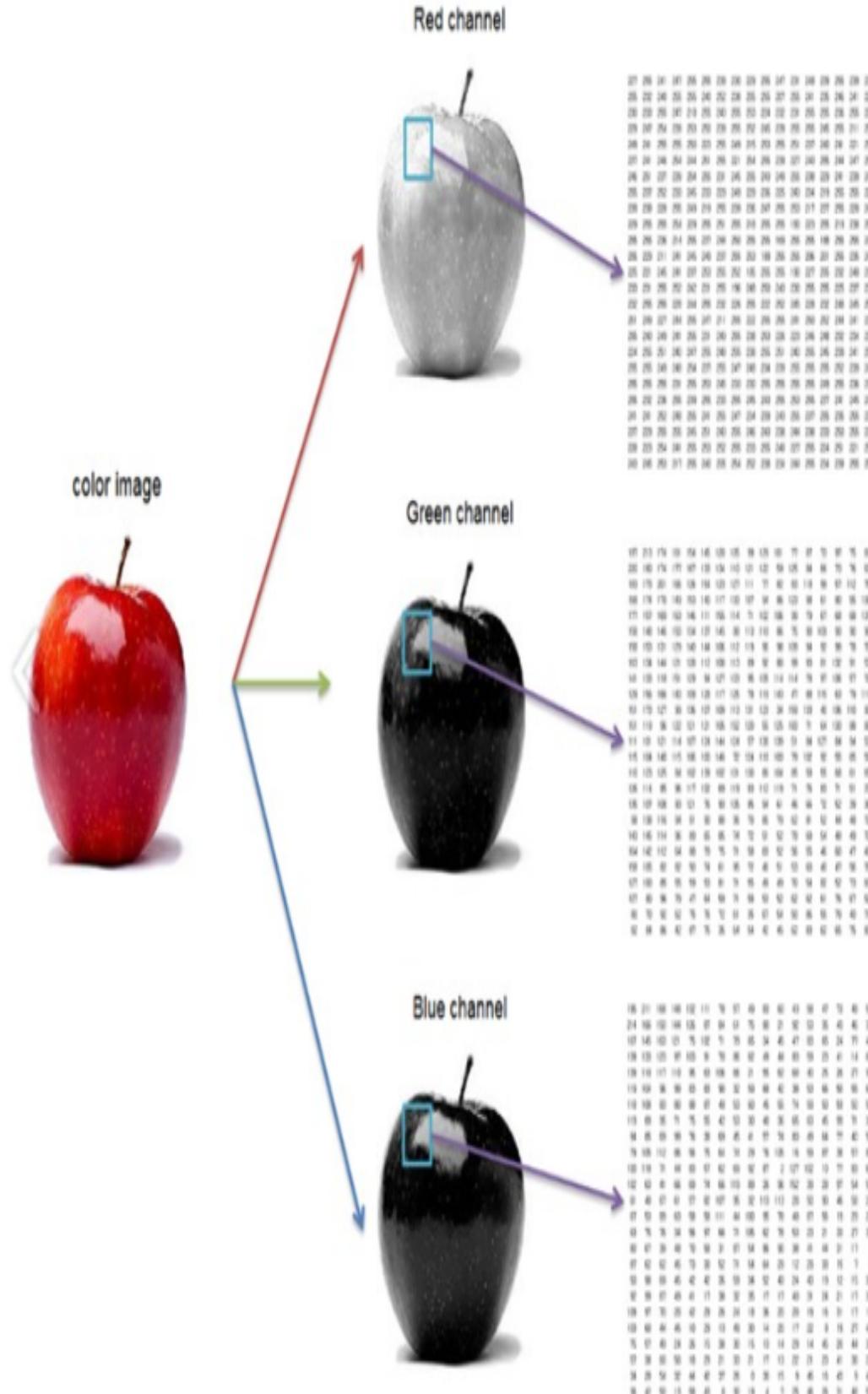


그림 출처 :

[http://contents.kocw.or.kr/KOCW/document
/2015/kumoh/kimseongyoung/03.pdf](http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/03.pdf)
(금오공대 김성영 교수님 공개 강의자료)

픽셀의 이해

■ 컬러 이미지(RGB)



픽셀의 이해

■ 컬러 이미지

- R, G, B값과 색상과의 관계

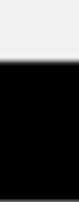
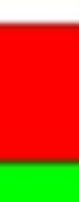
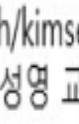
R	G	B	색상	
0	0	0	검은색	
255	255	255	흰색	
255	0	0	빨간색	
0	255	0	초록색	
0	0	255	파란색	
255	255	0	노란색	
128	0	0	어두운 빨간색	
69	69	69	어두운 회색	

그림 출처 :

픽셀의 이해

■ 이미지 용량 계산

흑백의 경우(픽셀 당 8비트 기준) :

이미지 용량(비트) = width x height x 8비트

이미지 용량(바이트) = (width x height x 8비트) x (1바이트/8비트)

(예) 세로 100픽셀, 가로 200픽셀인 경우

→ $100 \times 200 \times 8 = 160,000\text{비트} = 20,000\text{바이트} = 20\text{킬로바이트}$

컬러의 경우(픽셀 당 8비트 기준) :

이미지 용량(비트) = width x height x 8비트 x 3

이미지 용량(바이트) = (width x height x 8비트) x 3 x (1바이트/8비트)

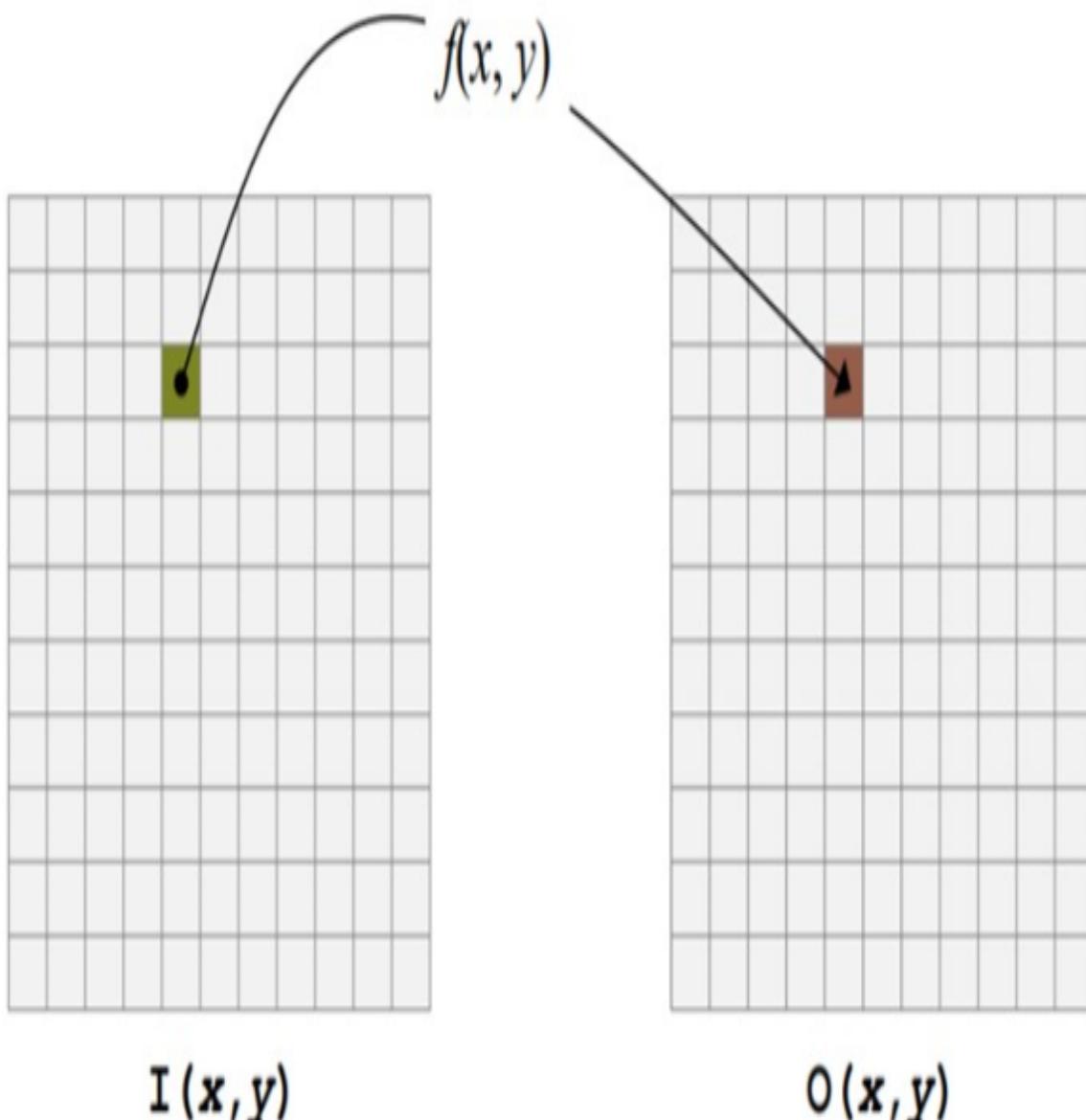
(예) 세로 100픽셀, 가로 200픽셀인 경우

→ $100 \times 200 \times 8 \times 3 = 480,000\text{비트} = 60,000\text{바이트} = 60\text{킬로바이트}$

픽셀 처리 이해

■ 픽셀 처리의 개념

- 입력 이미지의 픽셀값 $I(x,y)$ 이 $f(x,y)$ 를 통해 $O(x,y)$ 로 변경



$O(x,y)$

그림 출처 :

[http://contents.kocw.or.kr/KOCW/document
/2015/kumoh/kimseongyoung/03.pdf](http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/03.pdf)
(금오공대 김성영 교수님 공개 강의자료)

픽셀 처리 이해

■ 픽셀 처리의 종류

- 수학적 연산 Arithmetic Operation
- 히스토그램 수정 Histogram Modification

그림 출처 :

[http://contents.kocw.or.kr/KOCW/document
/2015/kumoh/kimseongyoung/05.pdf](http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/05.pdf)
(금오공대 김성영 교수님 공개 강의자료)

히스토그램 이해

■ 히스토그램

- 주어진 이미지에 픽셀값(8비트 : 0~255) 종류 별로 몇 개가 포함되어 있는지 그림으로 시각화 한 것

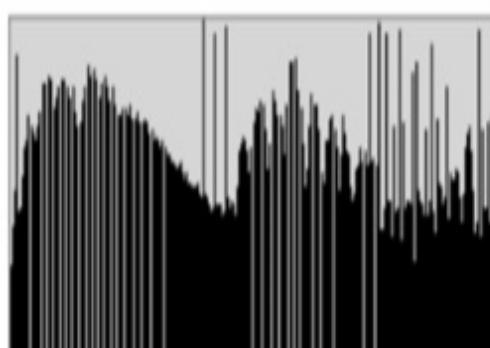


주어진 이미지

픽셀값
=명암값 빈도수

Bin	Counts
0	163
1	77
...	
255	1561

빈도수

number
of
pixels

픽셀값

gray level

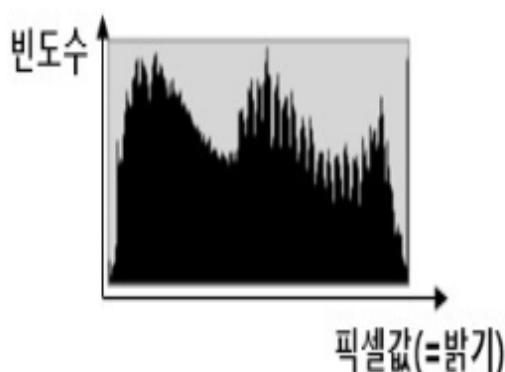
주어진 이미지에 대한 히스토그램

그림 출처 :

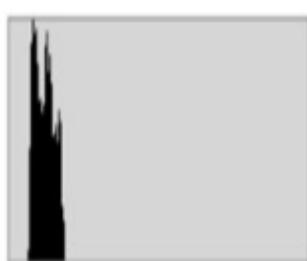
<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/06.pdf>
(금오공대 김성영 교수님 공개 강의자료)

밝기와 명암대비

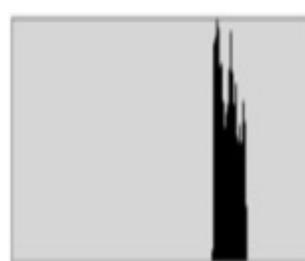
- 밝기 Brightness 와 명암대비 Contrast 이해



밝기 : 보통
명암대비 : 높음



밝기 : 낮음
명암대비 : 낮음



밝기 : 높음
명암대비 : 낮음

그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/05.pdf>
(금오공대 김성영 교수님 공개 강의자료)

수학적 연산

■ 수학적 연산

$$O(x, y) = k \times I(x, y) + l$$

출력 이미지 픽셀값
(처리 후)

입력 이미지 픽셀값
(처리 전)

l : level, k : gain

❖ 클리핑(clipping) 처리

if($O(x, y) > 255$) $O(x, y) = 255;$

if($O(x, y) < 0$) $O(x, y) = 0;$

그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/05.pdf>
(금오공대 김성영 교수님 공개 강의자료)

수학적 연산

- 픽셀값을 50 더하거나 빼는 경우 예시
 - 히스토그램이 왼쪽이나 오른쪽으로 시프트 됨

입력 이미지



출력 이미지(밝아짐)



$$l = 50, k = 1$$

출력 이미지(어두워짐)



$$l = -50, k = 1$$

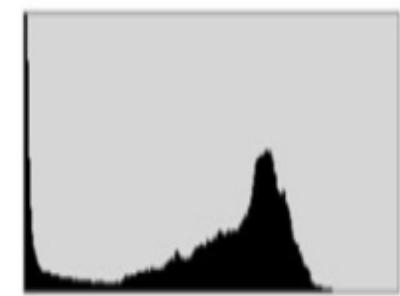
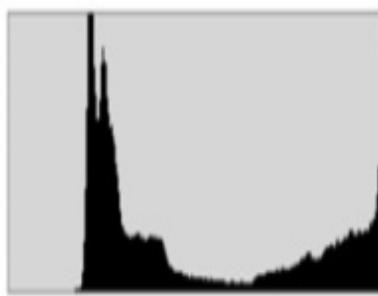
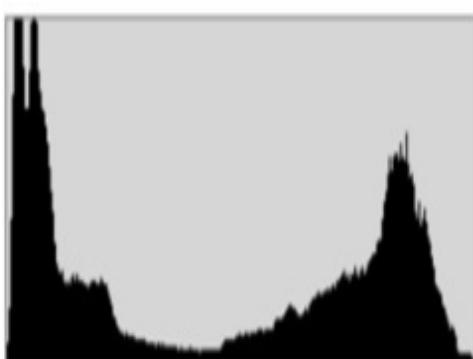


그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/05.pdf>
(금오공대 김성영 교수님 공개 강의자료)

수학적 연산

- 픽셀값을 1.2를 곱하거나 0.83을 곱하는 예시

입력 이미지



출력 이미지(밝아짐)



$$l = 0, k = 1.2$$

출력 이미지(어두워짐)



$$l = 0, k = 0.83$$

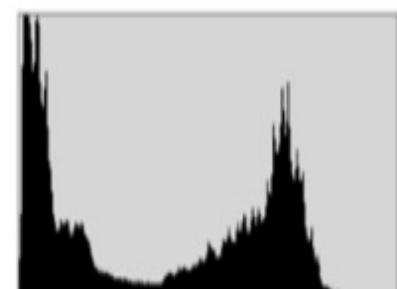
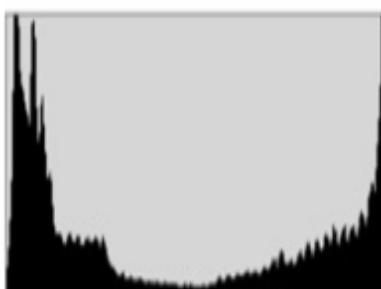
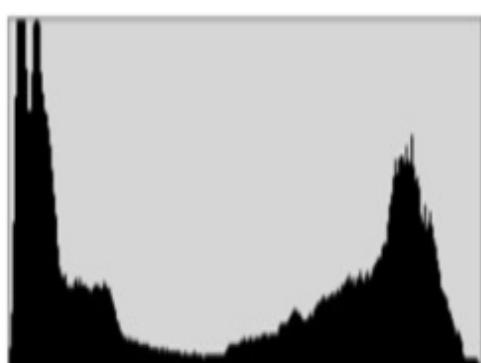


그림 출처 :

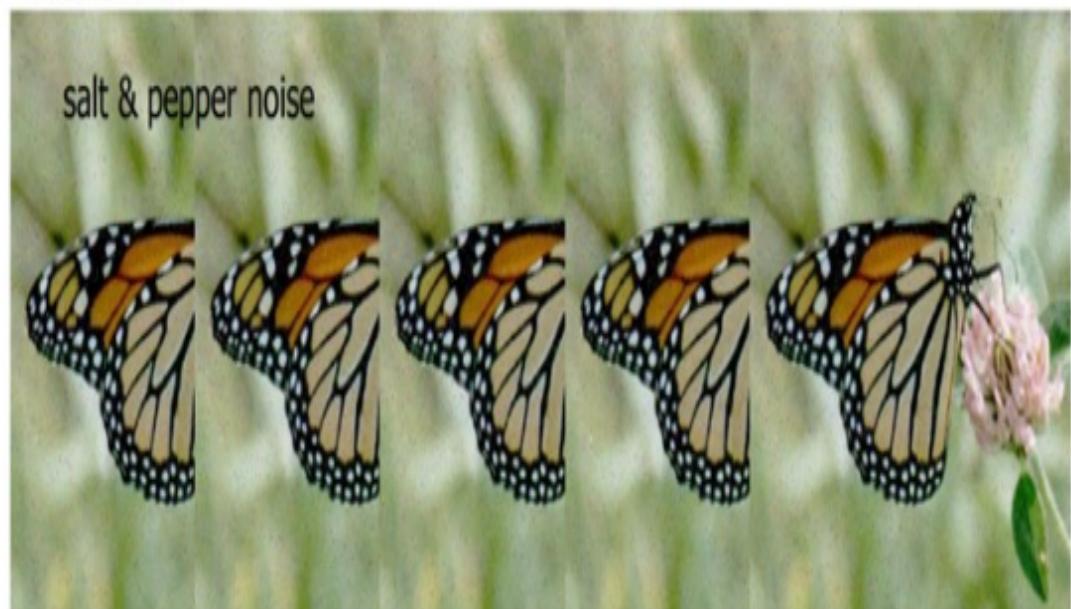
[http://contents.kocw.or.kr/KOCW/document
/2015/kumoh/kimseongyoung/05.pdf](http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/05.pdf)
(금오공대 김성영 교수님 공개 강의자료)

수학적 연산

■ 2개 이상의 이미지를 이용한 연산

- 이미지 평균 Averaging

2) 원본이미지에
랜덤 노이즈를
강제로 추가한
이미지 5개 준비



with 5 images

original image



3) 5장을 평균

1) 원본이미지
(노이즈 없음)

그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/05.pdf>
(금오공대 김성영 교수님 공개 강의자료)

수학적 연산

■ 2개 이상의 이미지를 이용한 연산

- 이미지 차 Image Difference



흰색 부분을 통해
트럭이 움직였음을
알 수 있다



그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/05.pdf>
(금오공대 김성영 교수님 공개 강의자료)

수학적 연산

- 2개 이상의 이미지를 이용한 연산
 - 이미지 더셈



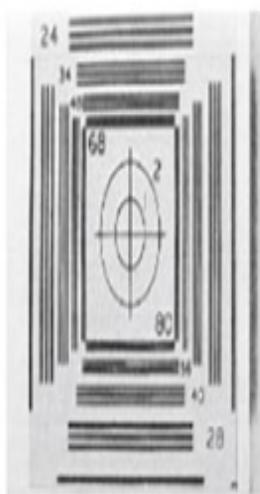
First Original



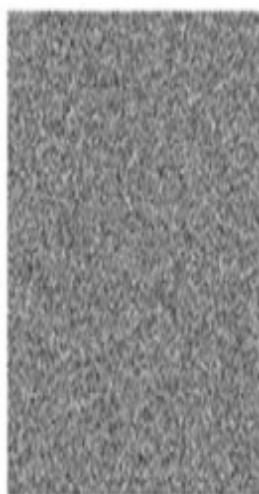
Second Original



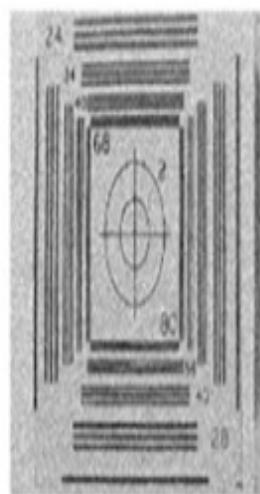
Addition of images



Original image



Gaussian noise

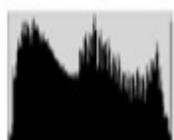


Addition of images

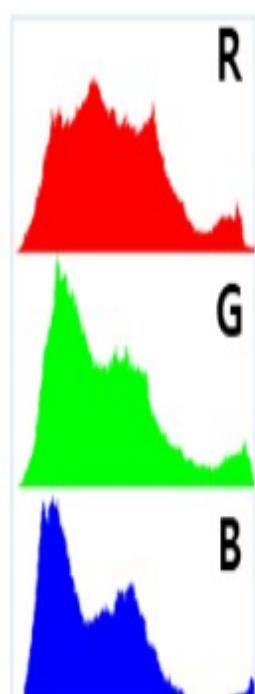
히스토그램 수정

■ 히스토그램

- 그레이스케일 이미지와 컬러 이미지에 대한 히스토그램 예시



그레이스케일 이미지의
히스토그램



컬러RGB 이미지의
히스토그램

그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/06.pdf>
(금오공대 김성영 교수님 공개 강의자료)

히스토그램 수정

- 히스토그램을 이용한 이미지 개선
 - 히스토그램 수정의 종류

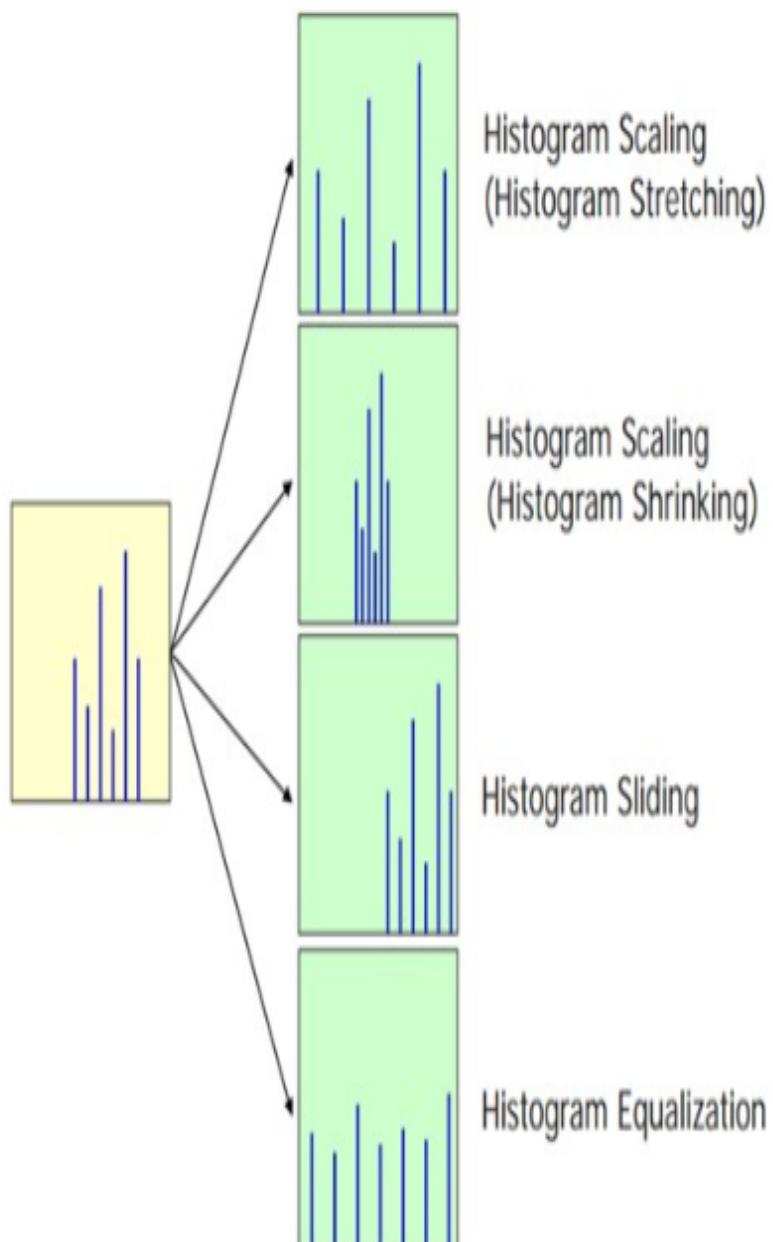
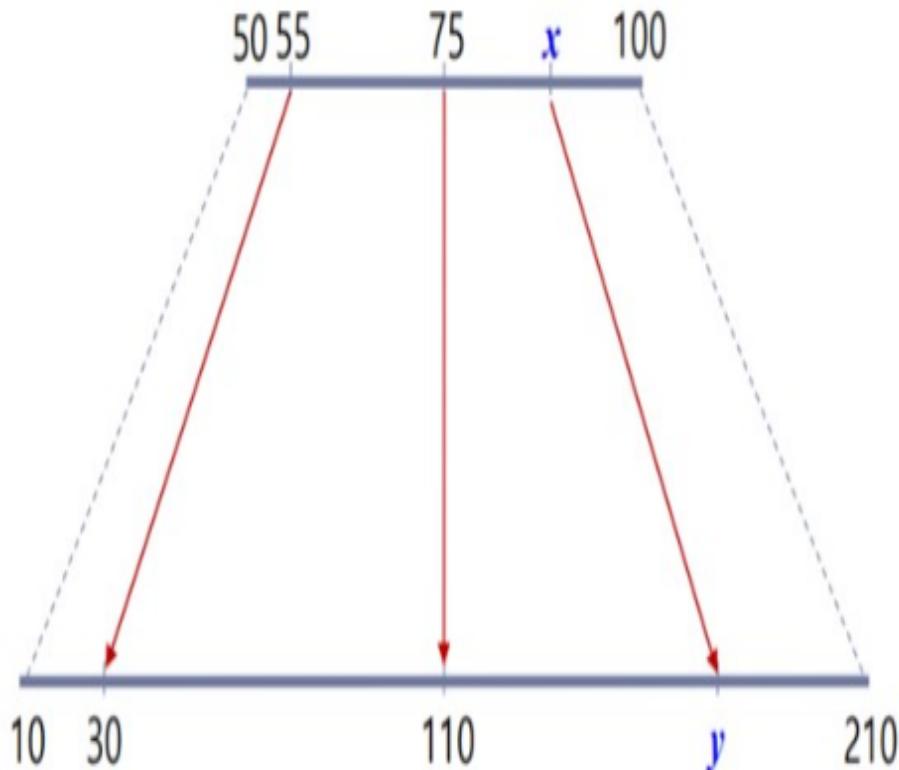


그림 출처 :

[http://contents.kocw.or.kr/KOCW/document
/2015/kumoh/kimseongyoung/06.pdf](http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/06.pdf)
(금오공대 김성영 교수님 공개 강의자료)

히스토그램 수정

■ 스케일링



$$(100-50):(x-50) = (210-10):(y-10)$$

$$(y-10)*(100-50) = (x-50)*(210-10)$$

$$y = \frac{(x-50)*(210-10)}{(100-50)} + 10 = \frac{(210-10)}{(100-50)}(x-50) + 10$$

$$I'(x,y) = \frac{(S_{\max} - S_{\min})}{(I_{\max} - I_{\min})}(I(x,y) - I_{\min}) + S_{\min}$$

I_{\max} : 이미지 내 최대 픽셀값 = 100

I_{\min} : 이미지 내 최소 픽셀값 = 50

S_{\max} : 수정 후 최대 픽셀값 = 210

S_{\min} : 수정 후 최소 픽셀값 = 10

그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/06.pdf>
(금오공대 김성영 교수님 공개 강의자료)

히스토그램 수정

■ 스케일링 결과 예시

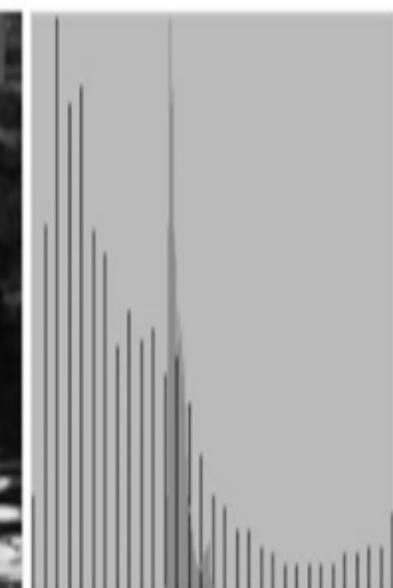
수정 전

Low-contrast
image



Histogram of
low-contrast
image

Image after
histogram
stretching



Histogram of
image after
stretching

수정 후

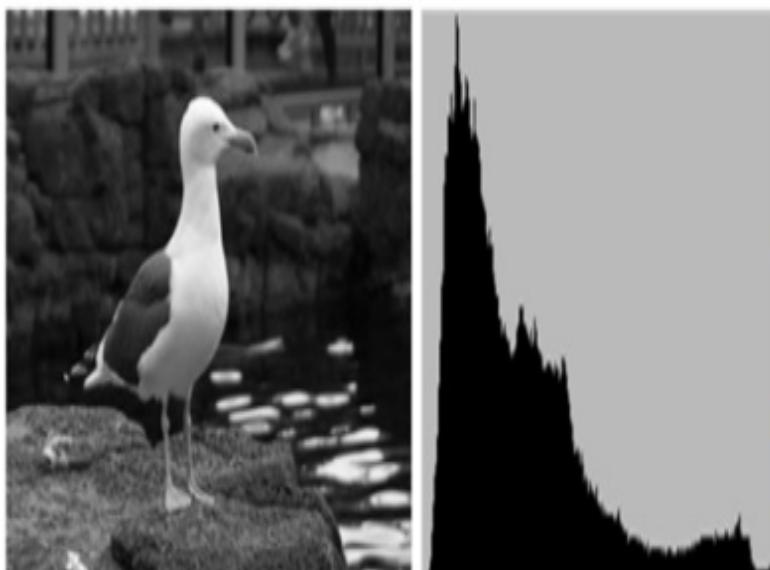
그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/06.pdf>
(금오공대 김성영 교수님 공개 강의자료)

히스토그램 수정

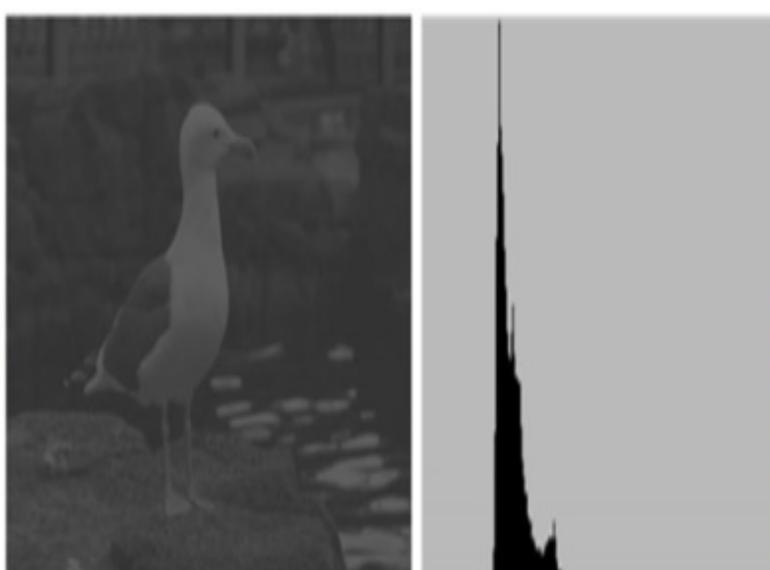
■ 스케일링 결과 예시

수정 전



Histogram
of original
image

Image after
shrinking



Histogram
of shrunk
image

수정 후

그림 출처 :

<http://contents.kocw.or.kr/KOCW/document/2015/kumoh/kimseongyoung/06.pdf>
(금오공대 김성영 교수님 공개 강의자료)

이미지 잡음의 종류

■ 가우시안 잡음

- 정규 분포를 갖는 잡음
- 이미지의 픽셀 값으로부터 불규칙적으로 벗어나지만 뚜렷하게 벗어나지 않는 잡음



원래 이미지



가우시안 잡음 이미지

이미지 잡음의 종류

■ 임펄스 잡음

- 이미지의 픽셀 값과는 뚜렷하게 다른 픽셀 값에 의한 잡음
- (8비트 기준) 0이나 255 같은 뚜렷한 밝기 값을 갖는 픽셀



원래 이미지

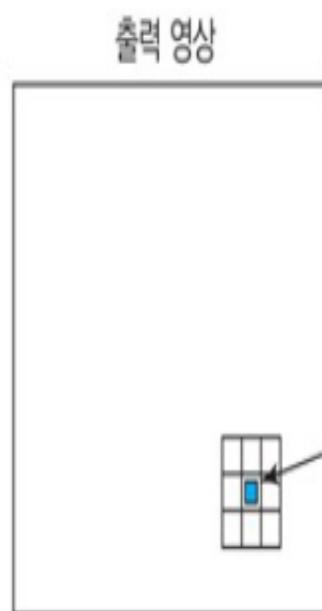
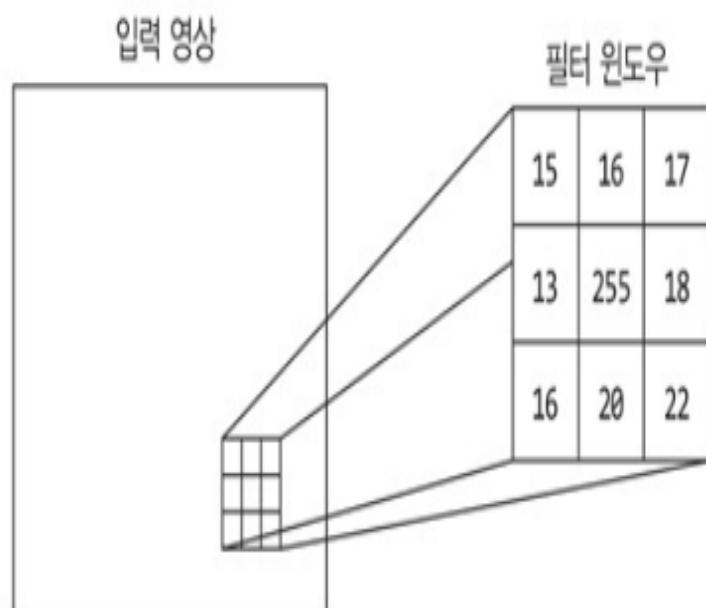


임펄스 잡음 이미지

평균 필터링

■ 평균 필터링

- 가우시안 잡음을 제거하기 위한 효과적인 방법
- 임펄스 잡음에는 비효과적
- 이미지 대비를 약화시킴



$$(15+16+17+ \\ 13+255+18+ \\ 16+20+22) / 9 \\ = 44$$

평균 필터링

▪ 평균 필터링 결과



가우시안 잡음 이미지



임펄스 잡음 이미지



가우시안 잡음 제거 결과

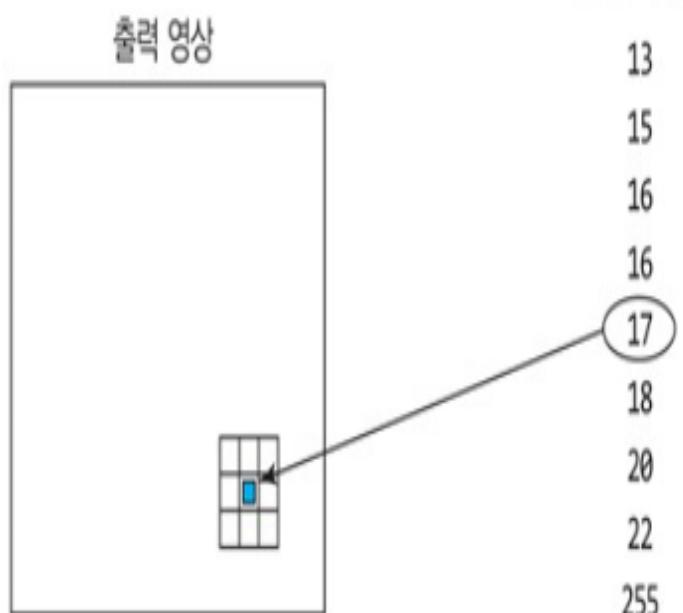
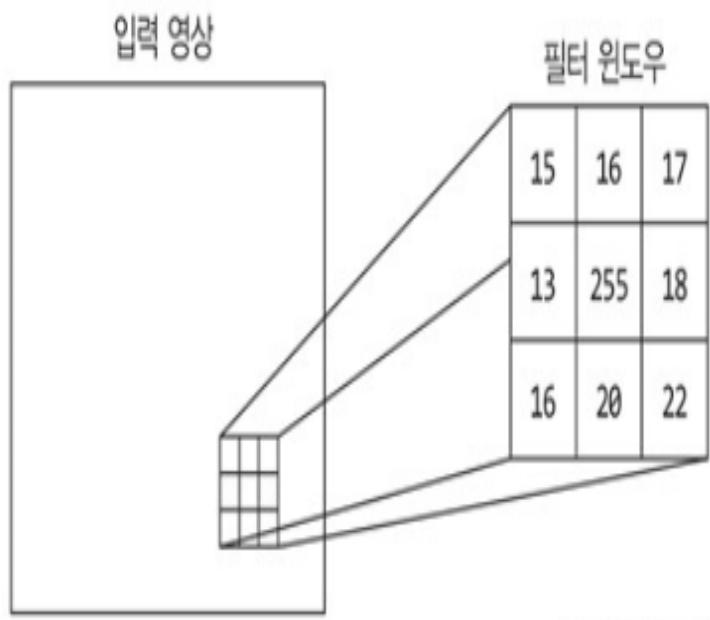


임펄스 잡음 제거 결과

중간값 필터링

■ 중간값 필터링

- 임펄스 잡음을 제거하기 위한 효과적인 방법
- 경계선을 보존



중간값 필터링

- 중간값 필터링 결과



가우시안 잡음 이미지



임펄스 잡음 이미지



가우시안 잡음 제거 결과



임펄스 잡음 제거 결과

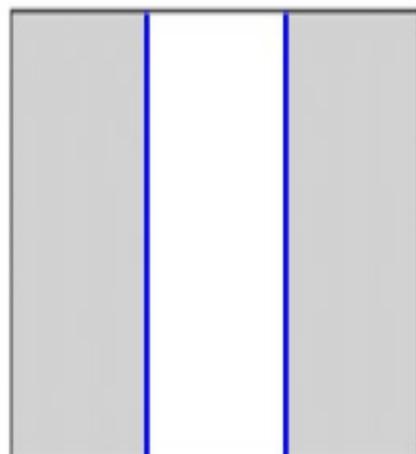
출처 : Matlab을 이용한 실용 이미지처리
(문성룡, 정성태, 조준호)

에지의 이해

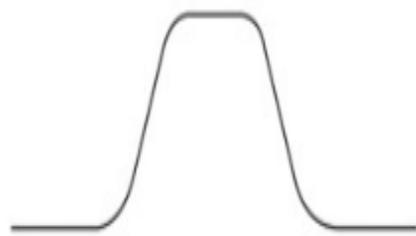
■ 에지 Edge

- 이미지의 밝기가 낮은 값에서 높은 값으로 또는 높은 값에서 낮은 값으로 변하는 지점에 존재
- 물체의 위치, 모양, 크기 등을 인지하는데 효과적

영상



명암도 변화



소벨 연산자를 이용한 에지 검출

■ 소벨 Sobel 연산자

- 에지 검출을 위한 3×3 필터 정의

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

 m_y

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

 m_x

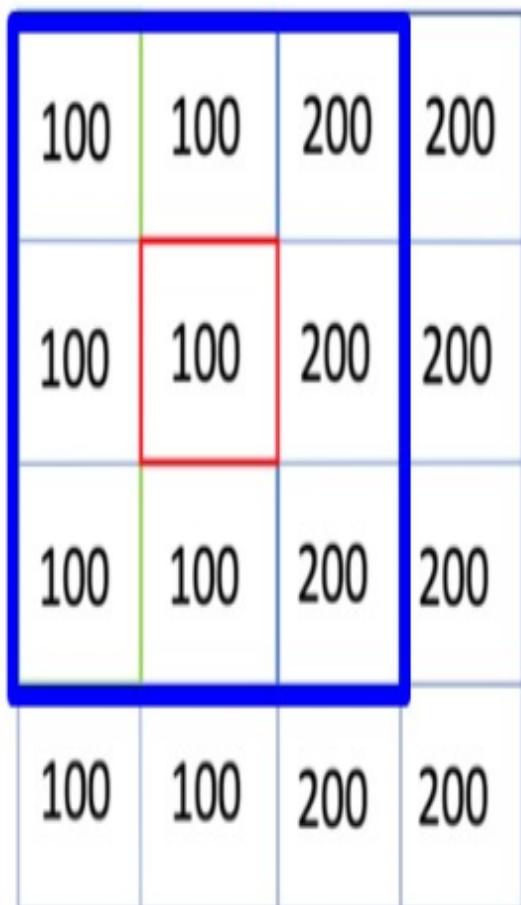
수평선 검출용
 3×3 필터

수직선 검출용
 3×3 필터

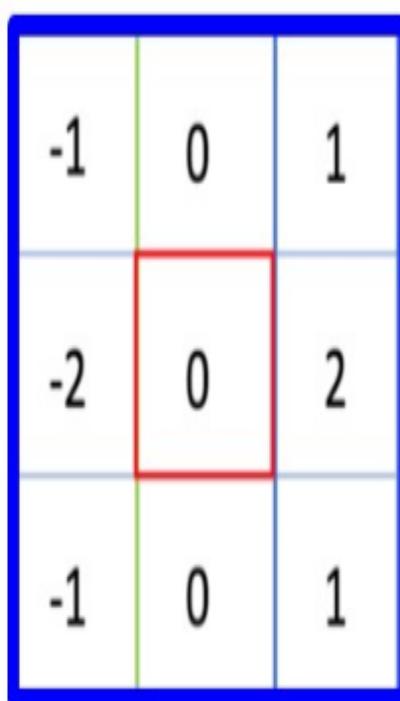
소벨 연산자를 이용한 에지 검출

■ 소벨 Sobel 연산자

- 수직선 검출용 필터(m_x) 적용 예시
- 수평선 검출용 필터(m_y)도 동일한 방법으로 적용



4x4 입력 이미지

수직선 검출용
3x3 필터
(m_x)

$$\begin{array}{r}
 -100 \\
 -200 \\
 -100 \\
 200 \\
 400 \\
 \hline
 +200 \\
 =400
 \end{array}$$

필터 적용 후
빨간색 좌표의
픽셀 값: 400

소벨 연산자를 이용한 에지 검출

■ 소벨 Sobel 연산자

- 적용 결과

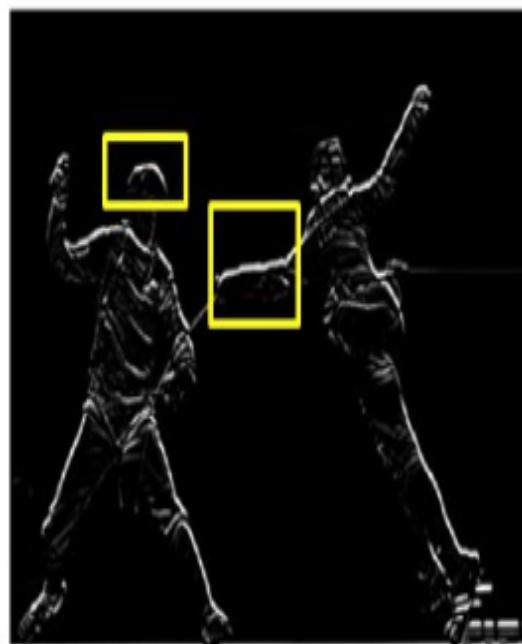
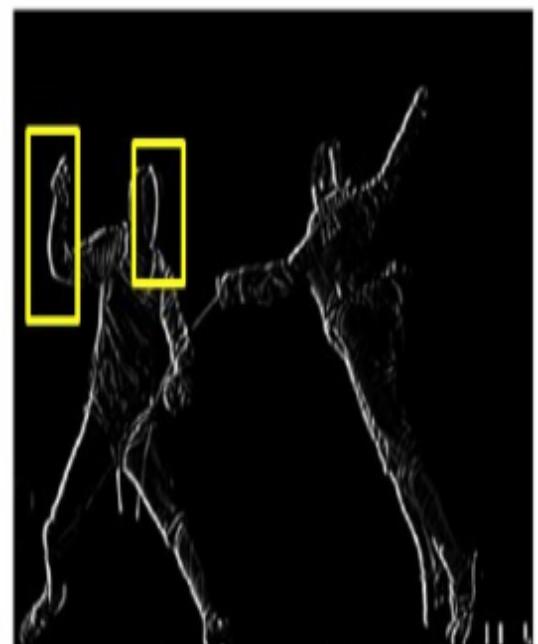
$$\text{에지 강도 : } S(y, x) = \text{magnitude}(\nabla f) = \sqrt{d_y^2 + d_x^2}$$



(a) 원래 영상



(b) 에지 강도

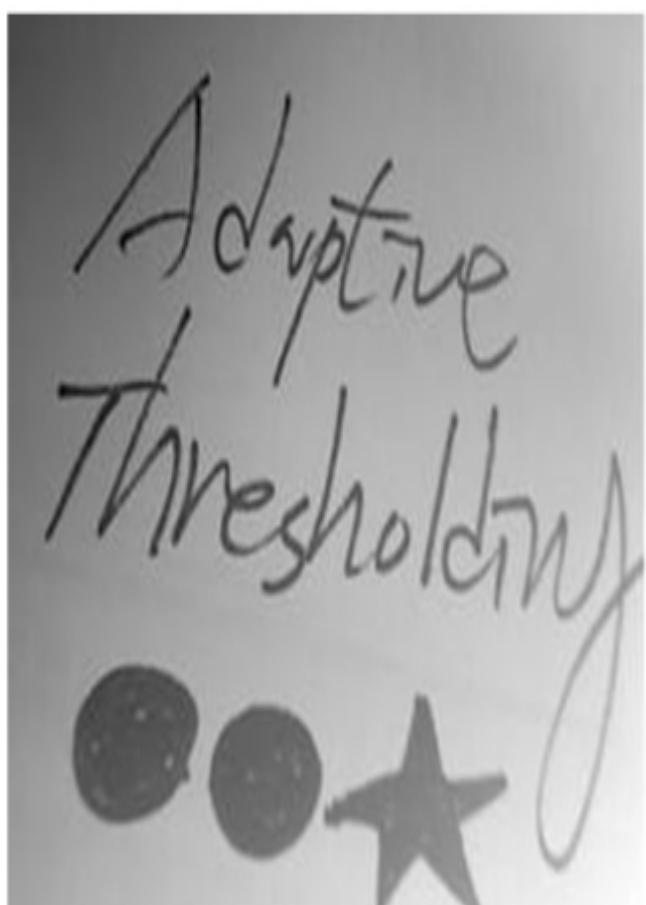
(c) d_y 수평선 검출(d) d_x 수직선 검출

d_x 나 d_y 보다는
에지 강도 결과를
컴퓨터 비전에
많이 활용함

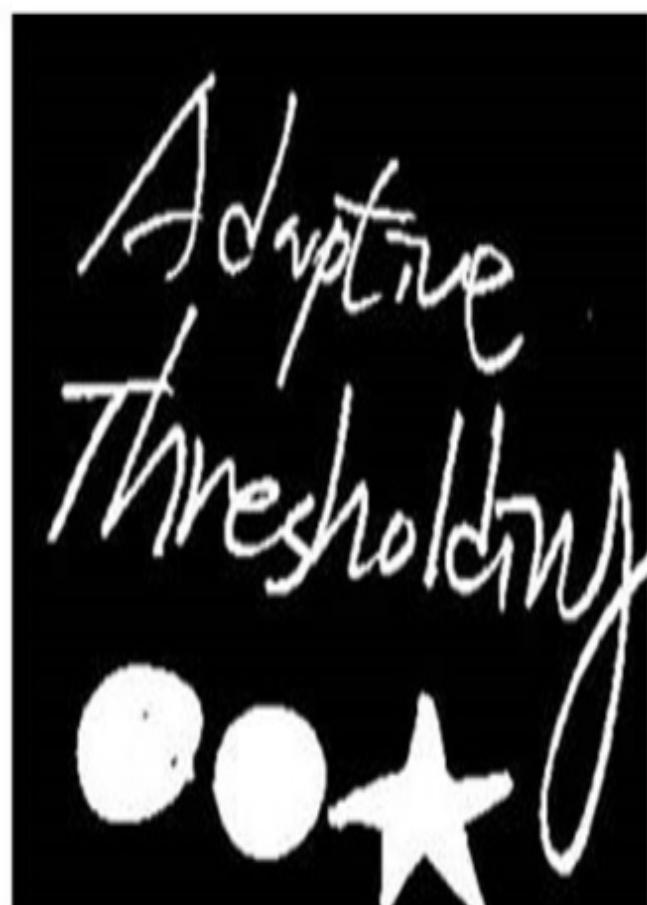
이진화란?

■ 이진화 Binarization 목적

- 2가지 픽셀값(검정색/흰색)을 갖는 이미지로 변환함으로써 분석을 용이하게 할 수 있음
- 픽셀값이 문턱값보다 크거나 같으면 흰색, 작으면 검정색으로 표시



입력 이미지

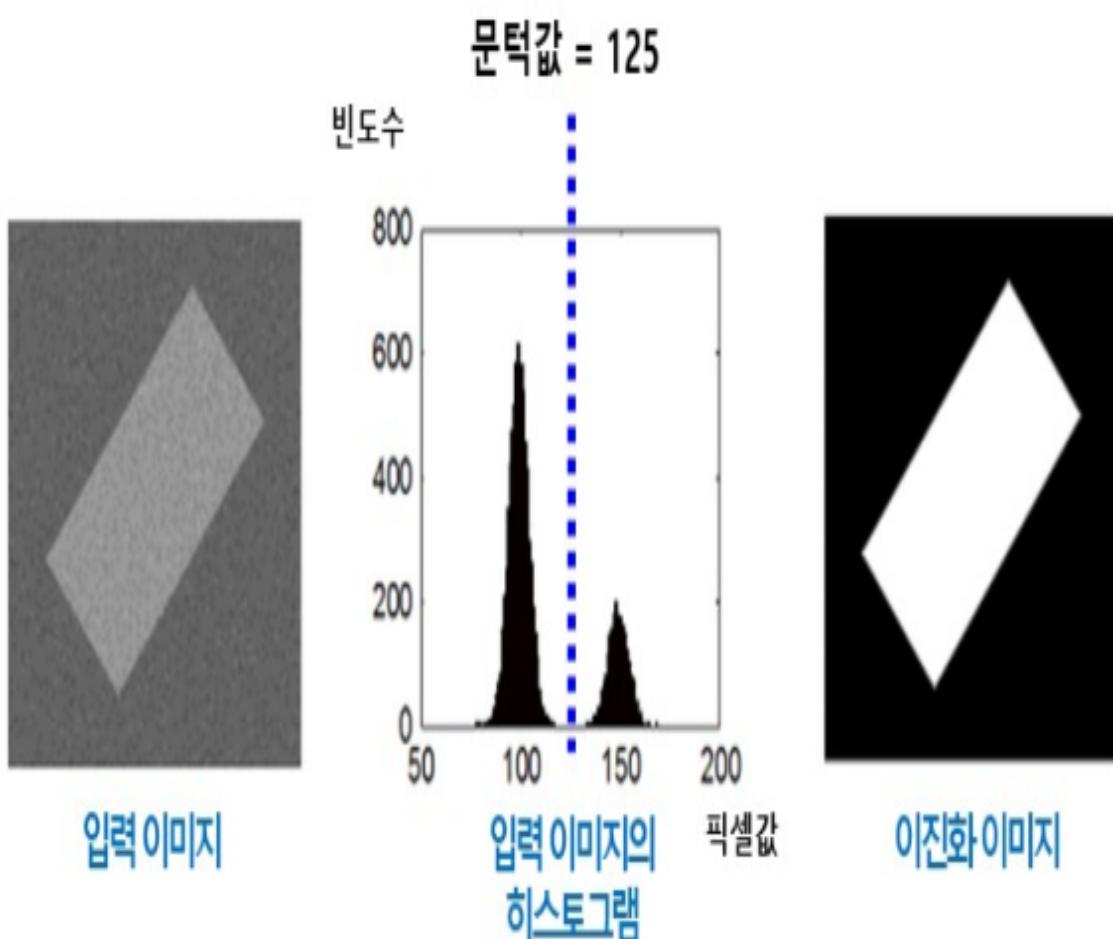


이진화 이미지

이진화란?

■ 간단한 예제

- 입력 이미지의 히스토그램을 관찰하여 문턱값을 125 정도로 정하면 깨끗한 이진화 이미지를 얻을 수 있음
- 이미지 전체에 동일한 문턱값을 적용하므로 전역적 문턱값 처리 방법이라고 함

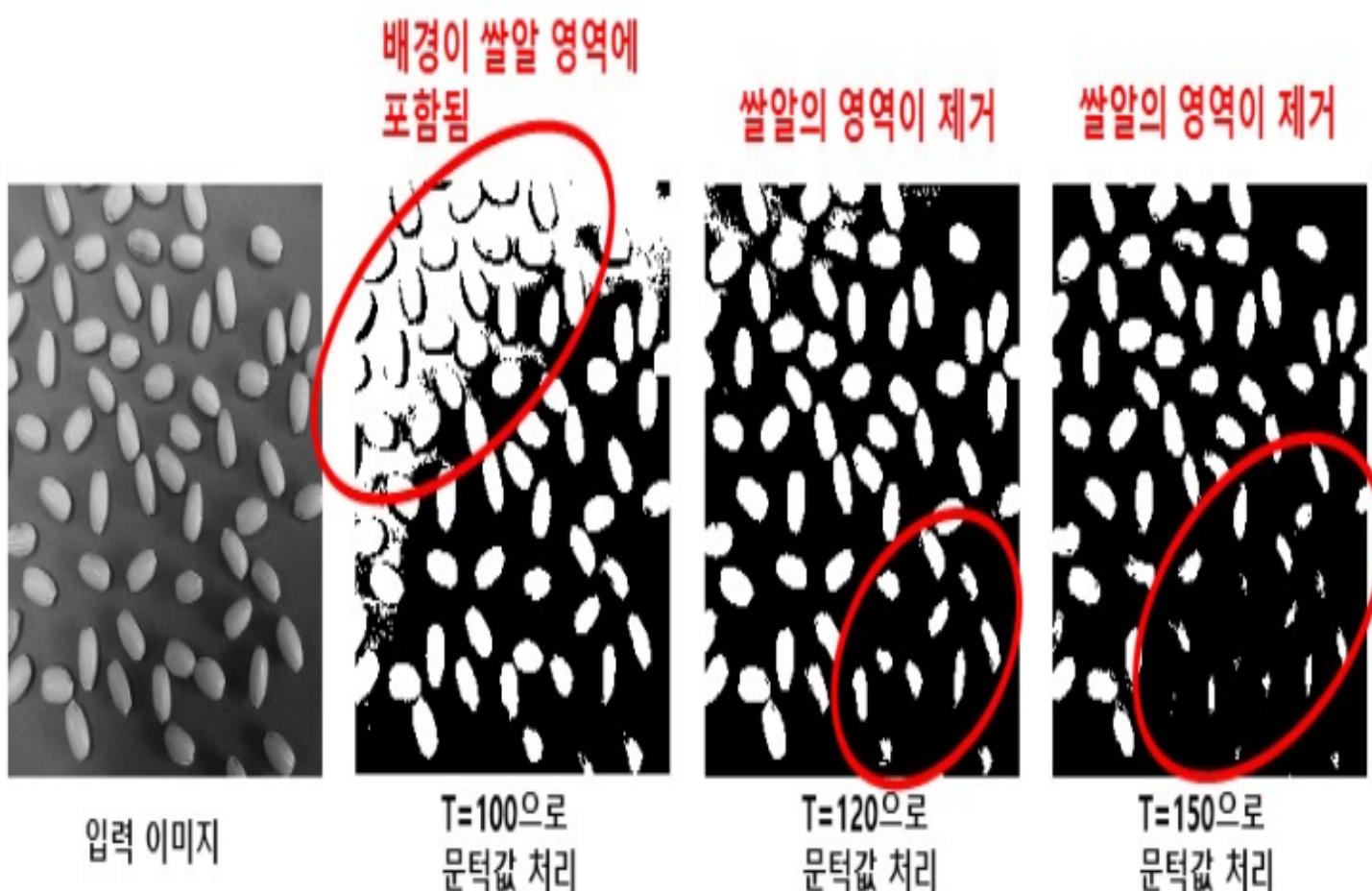


전역적 문턱값 처리

■ 전역적 문턱값 처리의 문제점

- 이미지의 밝기가 고르지 않은 경우 한 가지 문턱값으로는 완벽하게 쌀알과 배경 영역을 분리하기 어려움

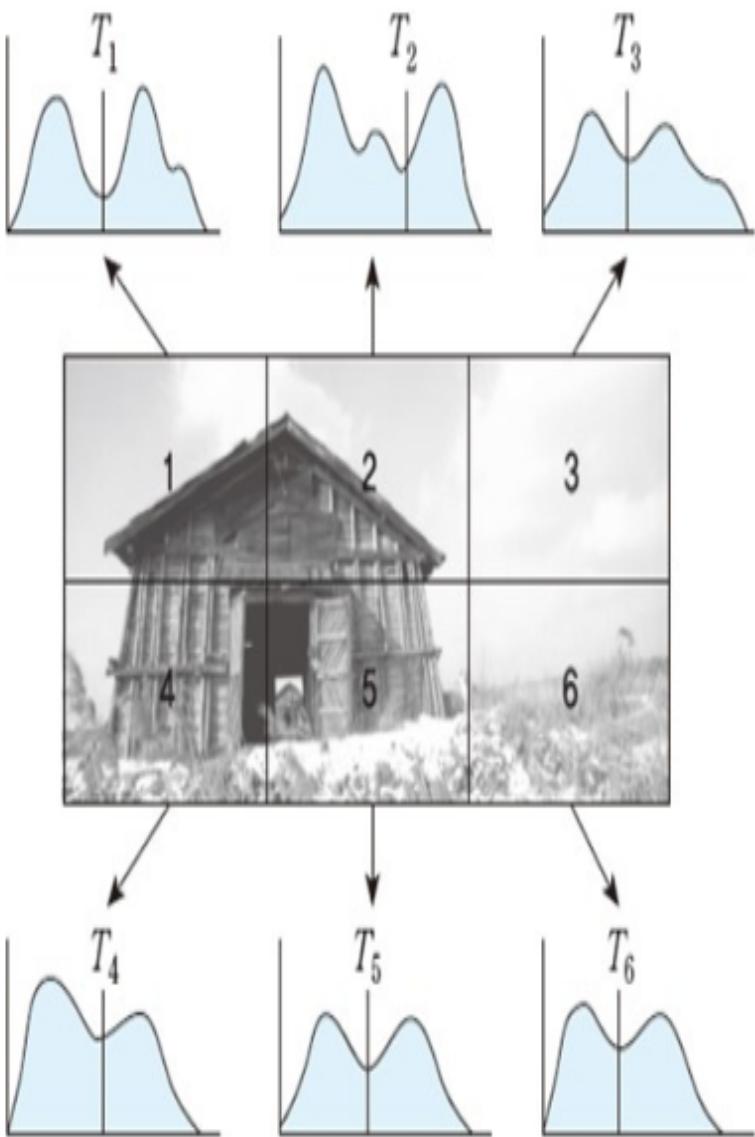
예 : 쌀알 픽셀범위 : 110~180
배경 픽셀범위 : 70~125



영역별 문턱값 처리

■ 영역별 문턱값 처리의 개념

- 입력 이미지를 여러 개의 부분 이미지로 나누고, 분할된 이미지의 히스토그램들로부터 각각의 알맞은 문턱값($T_1 \sim T_6$)을 선택하여 원하는 물체를 검출

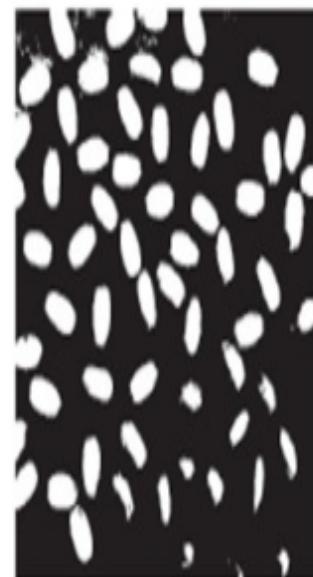


영역별 문턱값 처리

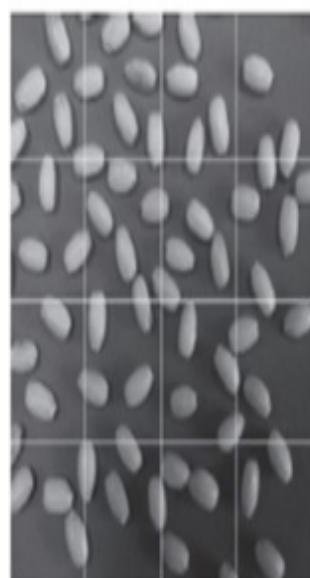
- 전역적, 영역별 문턱값 처리 결과 비교



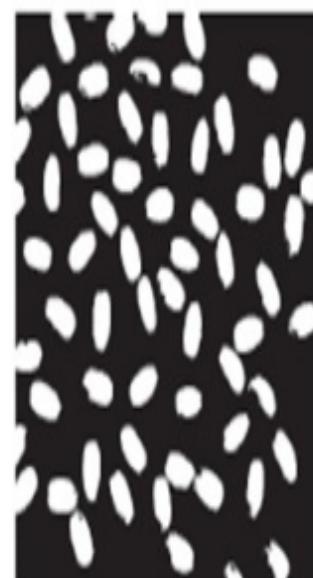
(a) 입력영상



(b) 전역 문턱값을 이용한 문턱값 처리 결과



(c) 영상분할

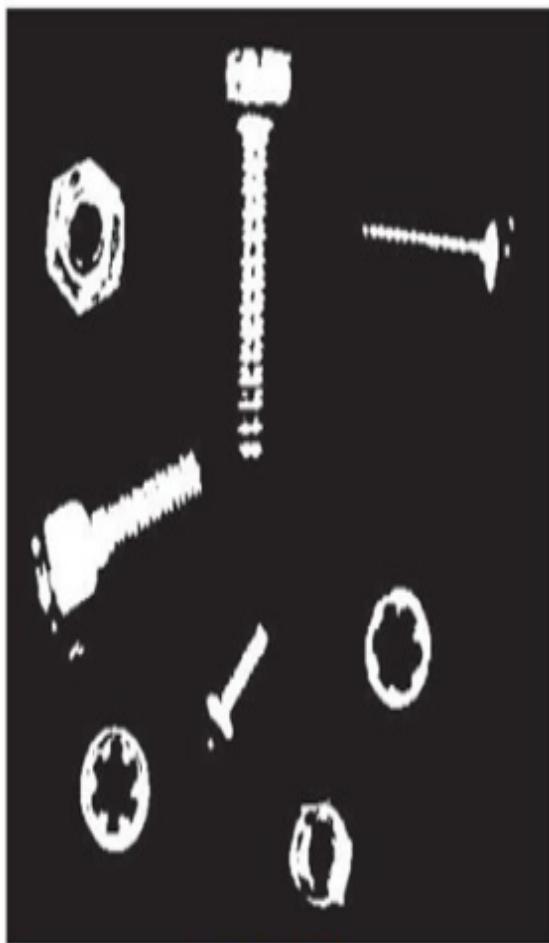


(d) 영역별 문턱값 처리 결과

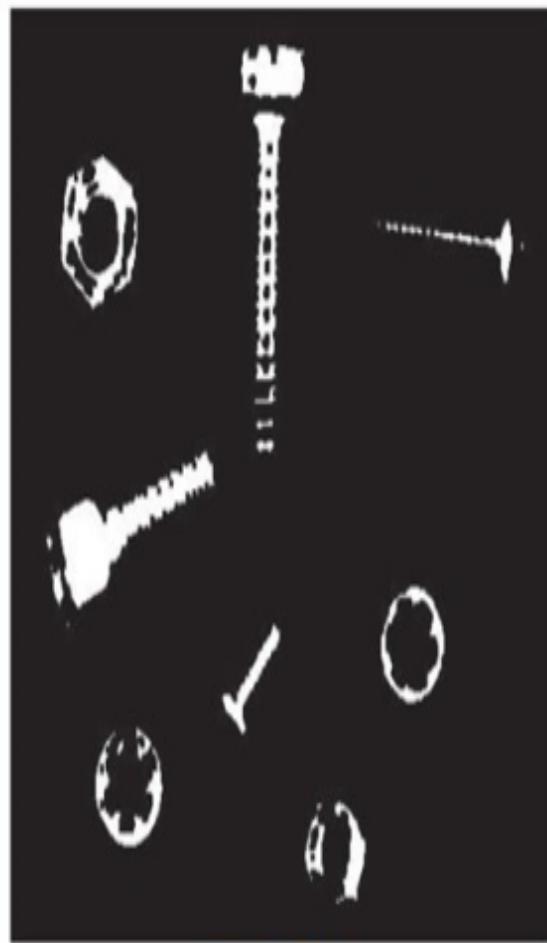
침식 연산

▪ 침식 연산

- 흰색 물체의 크기를 축소하고 검정색 배경을 확장



입력 이미지

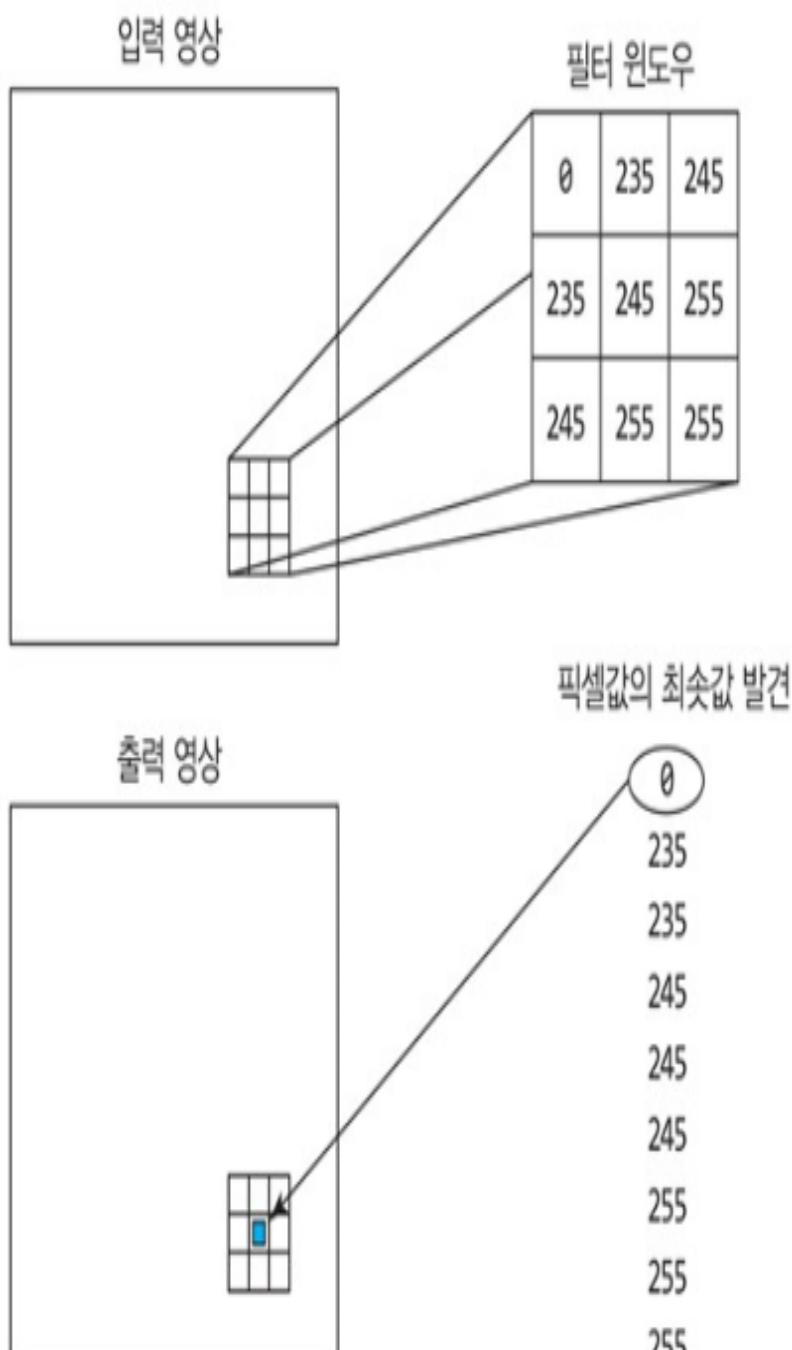


침식 이미지

침식 연산

■ 침식 연산의 아이디어

- 3x3 영역 내의 픽셀값의 최솟값을 그 영역의 중심 픽셀값으로 사용

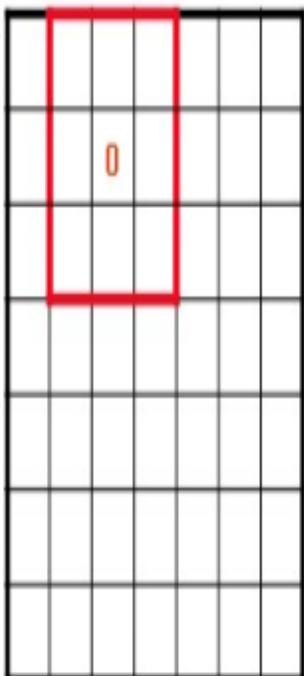


침식 연산

0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0

입력 이미지

(a)

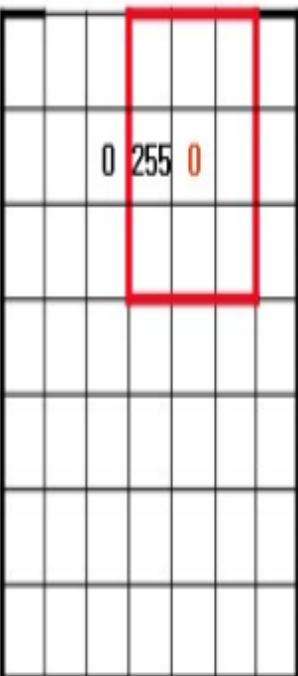


출력 이미지

0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0

입력 이미지

(c)

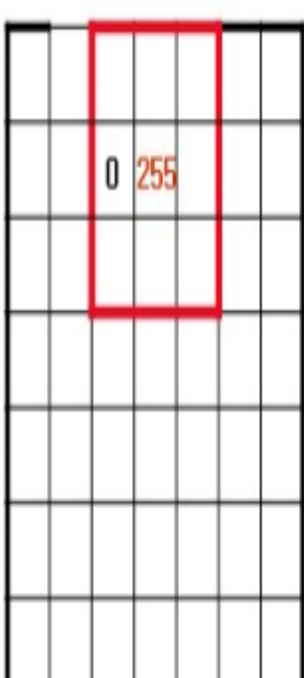


출력 이미지

0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0

입력 이미지

(b)

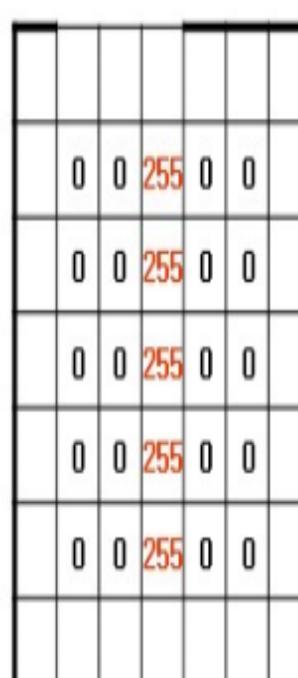


출력 이미지

0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0
0	0	255	255	255	0	0

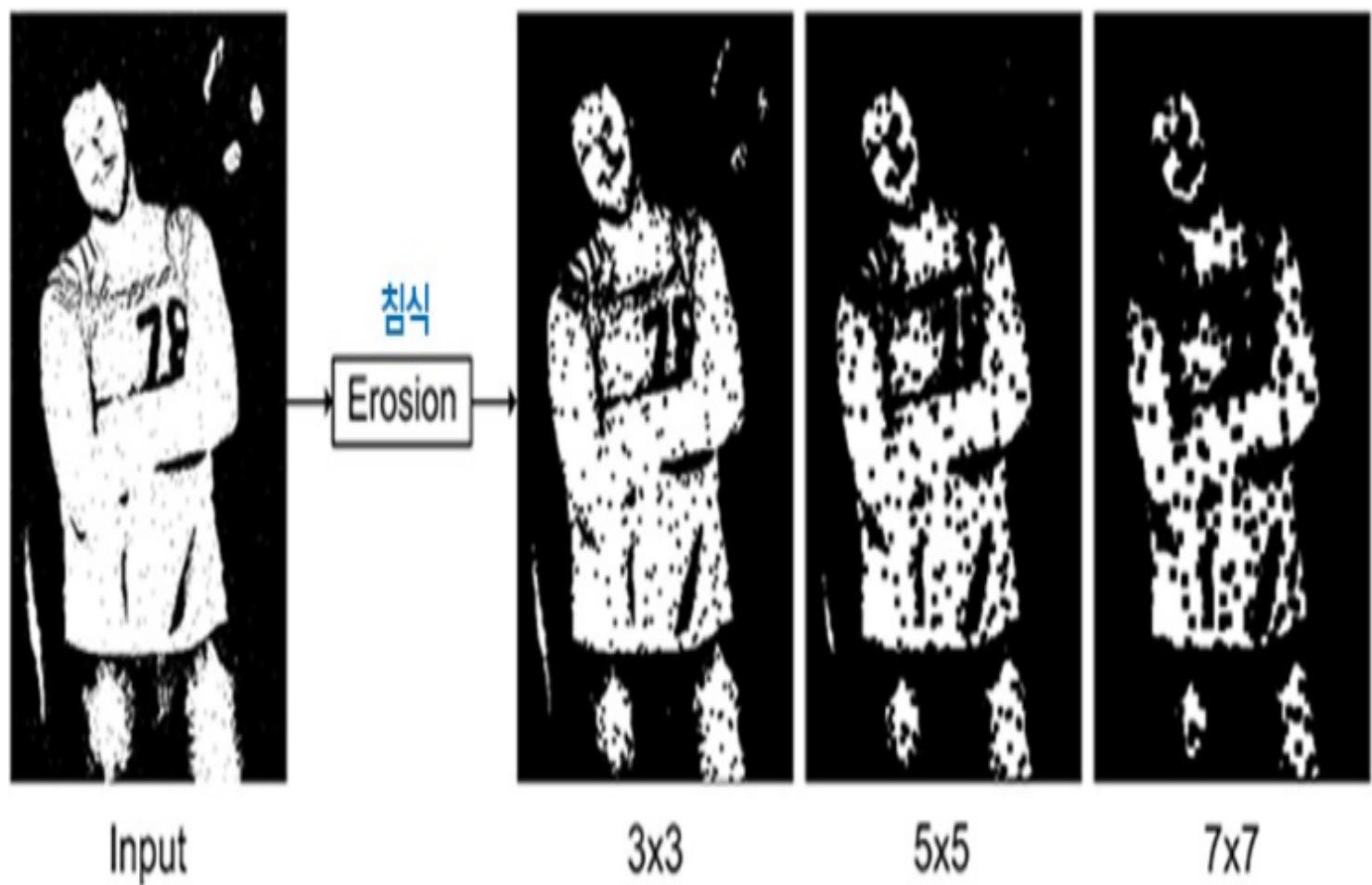
입력 이미지

(d)



출력 이미지

침식 연산

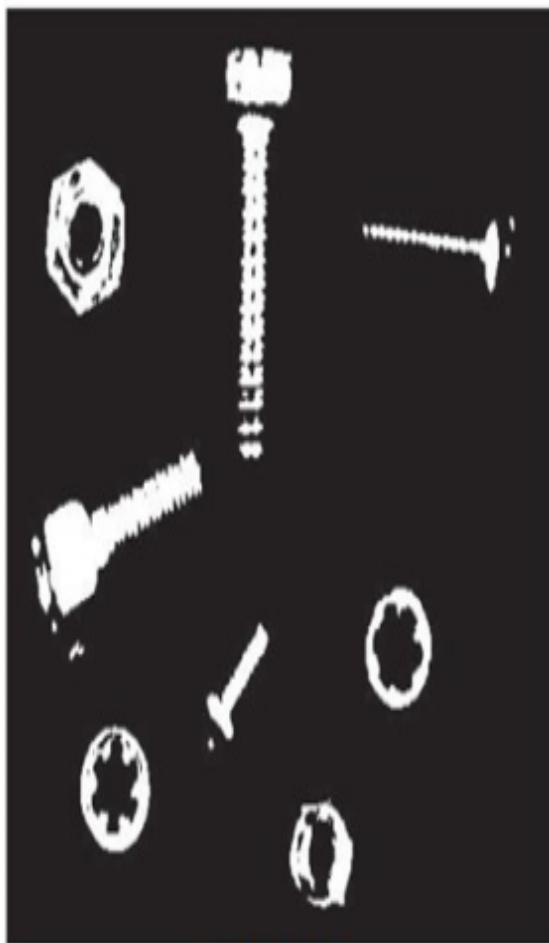


출처 : <http://what-when-how.com/introduction-to-video-and-image-processing/morphology-introduction-to-video-and-image-processing-part-2/>

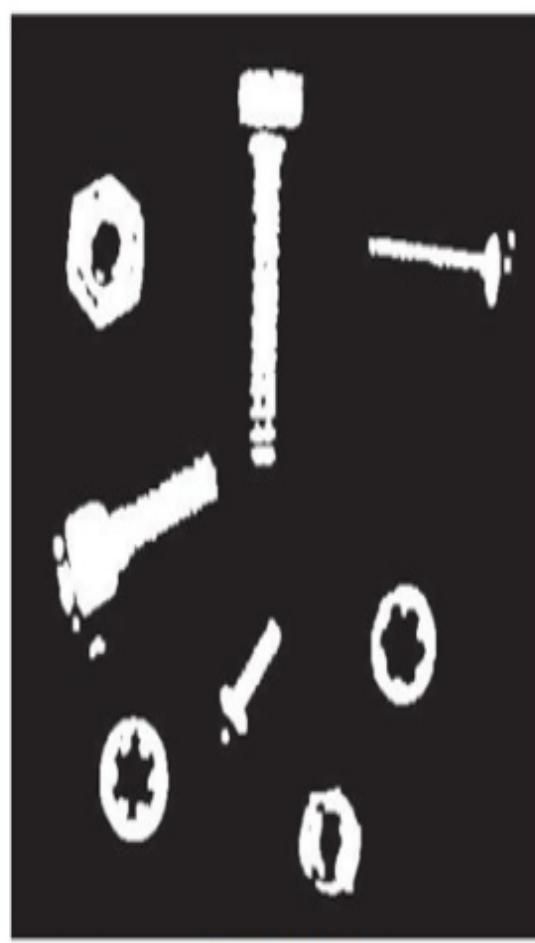
팽창 연산

■ 팽창 연산

- 흰색 물체의 크기를 확장하고 검정색 배경을 축소



입력 이미지

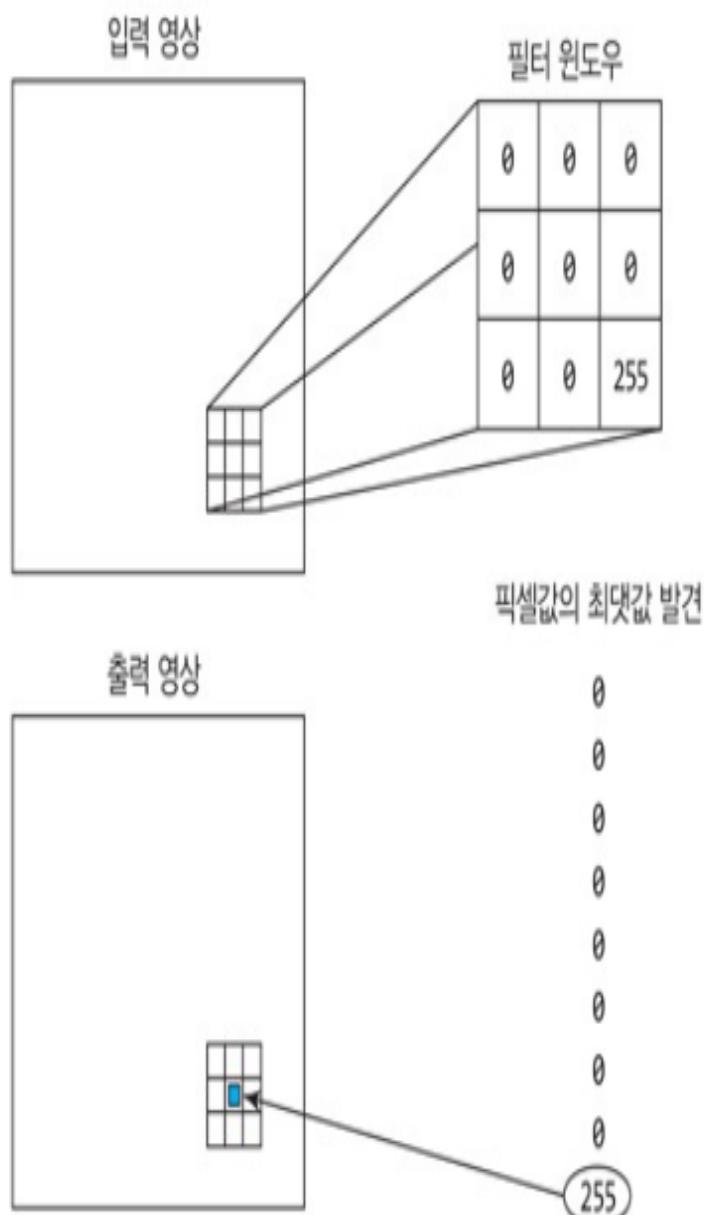


팽창 이미지

팽창 연산

■ 팽창 연산의 아이디어

- 3x3 영역 내의 픽셀값의 최댓값을 그 영역의 중심 픽셀값으로 사용

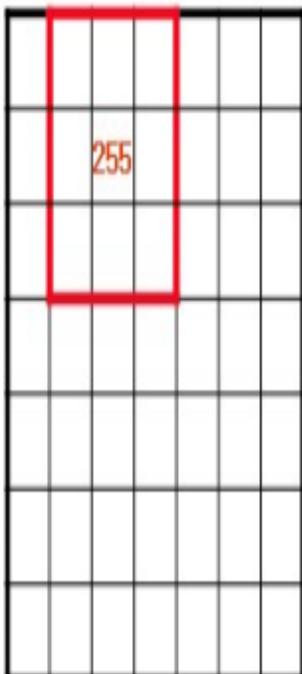


팽창 연산

0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0

입력 이미지

(a)

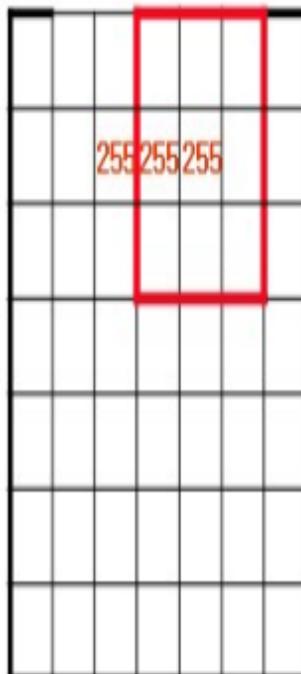


출력 이미지

0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0

입력 이미지

(c)

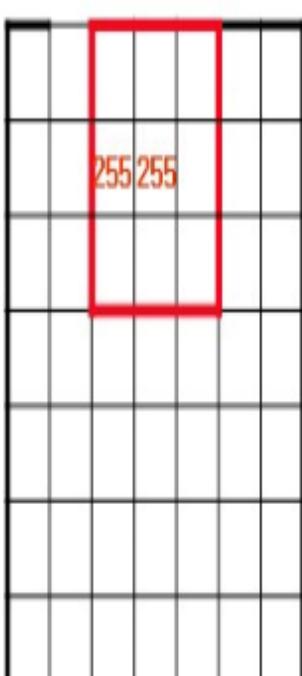


출력 이미지

0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0

입력 이미지

(b)

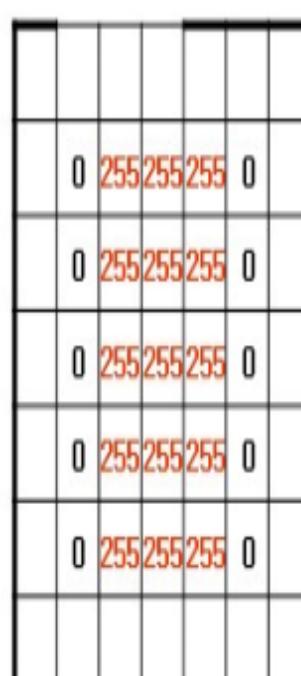


출력 이미지

0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0
0	0	0	255	0	0	0

입력 이미지

(d)

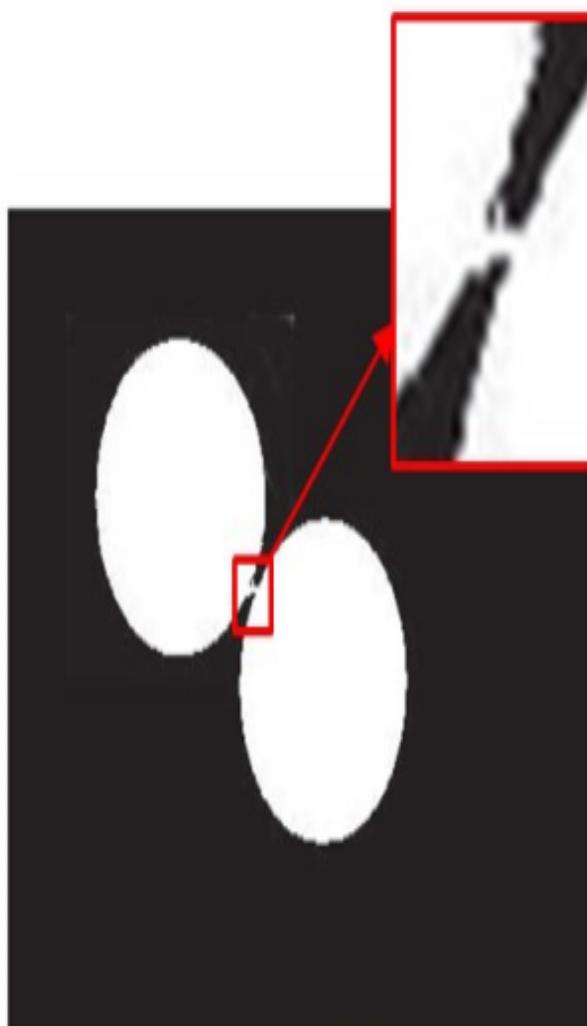


출력 이미지

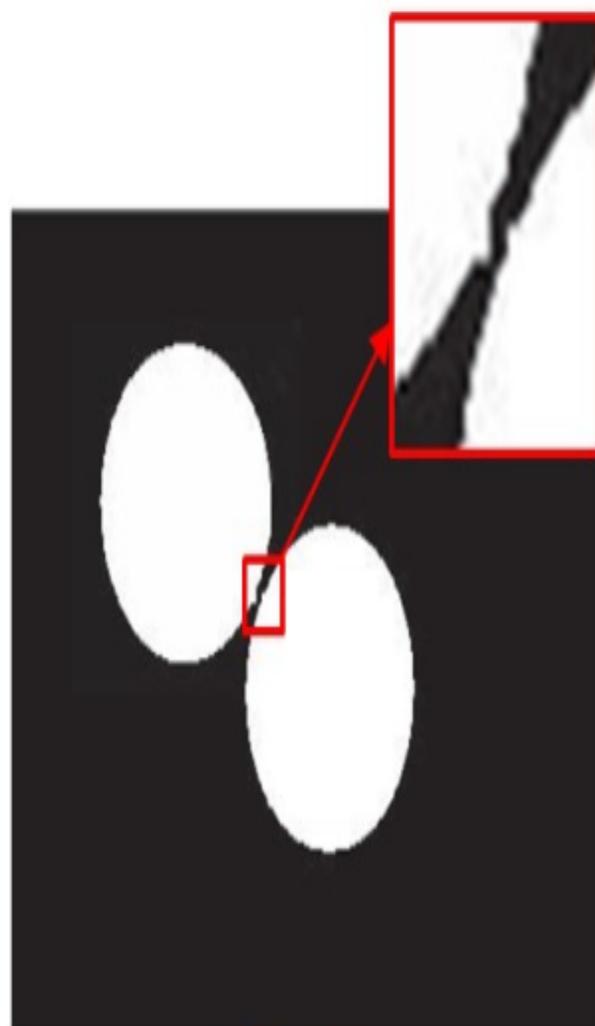
열림 연산

■ 열림 연산

- 침식 연산을 적용한 다음에 팽창 연산을 적용하는 연산
- 이미지에서 약한 부분을 잘라내어 물체를 서로 분리하는데 유용



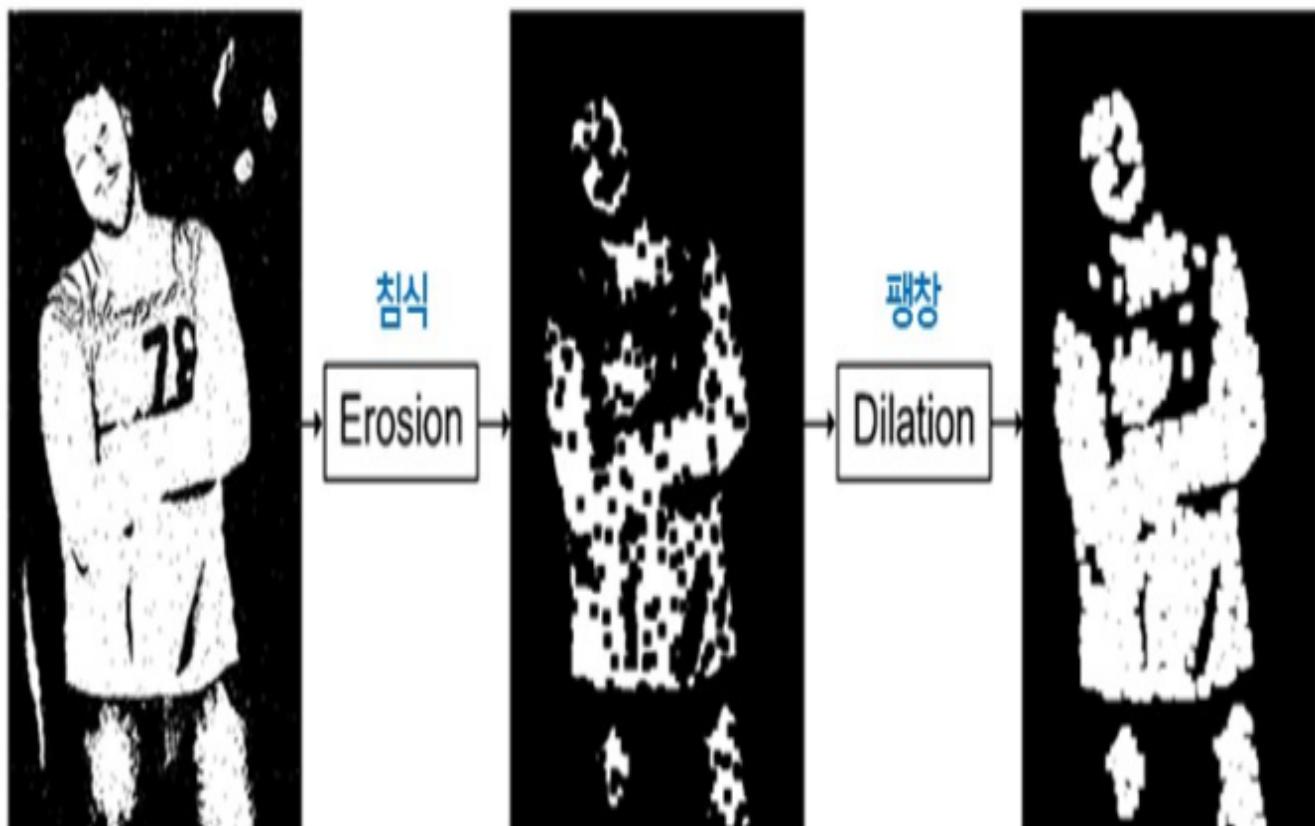
입력 이미지



열림 이미지

얼림 연산

■ 얼림 연산



출처 : <http://what-when-how.com/introduction-to-video-and-image-processing/morphology-introduction-to-video-and-image-processing-part-2/>

닫힘 연산

▪ 닫힘 연산

- 팽창 연산을 적용한 다음에 침식 연산을 적용하는 연산
- 이미지에서 작은 구멍을 메우거나 분리된 물체를 결합하는 역할

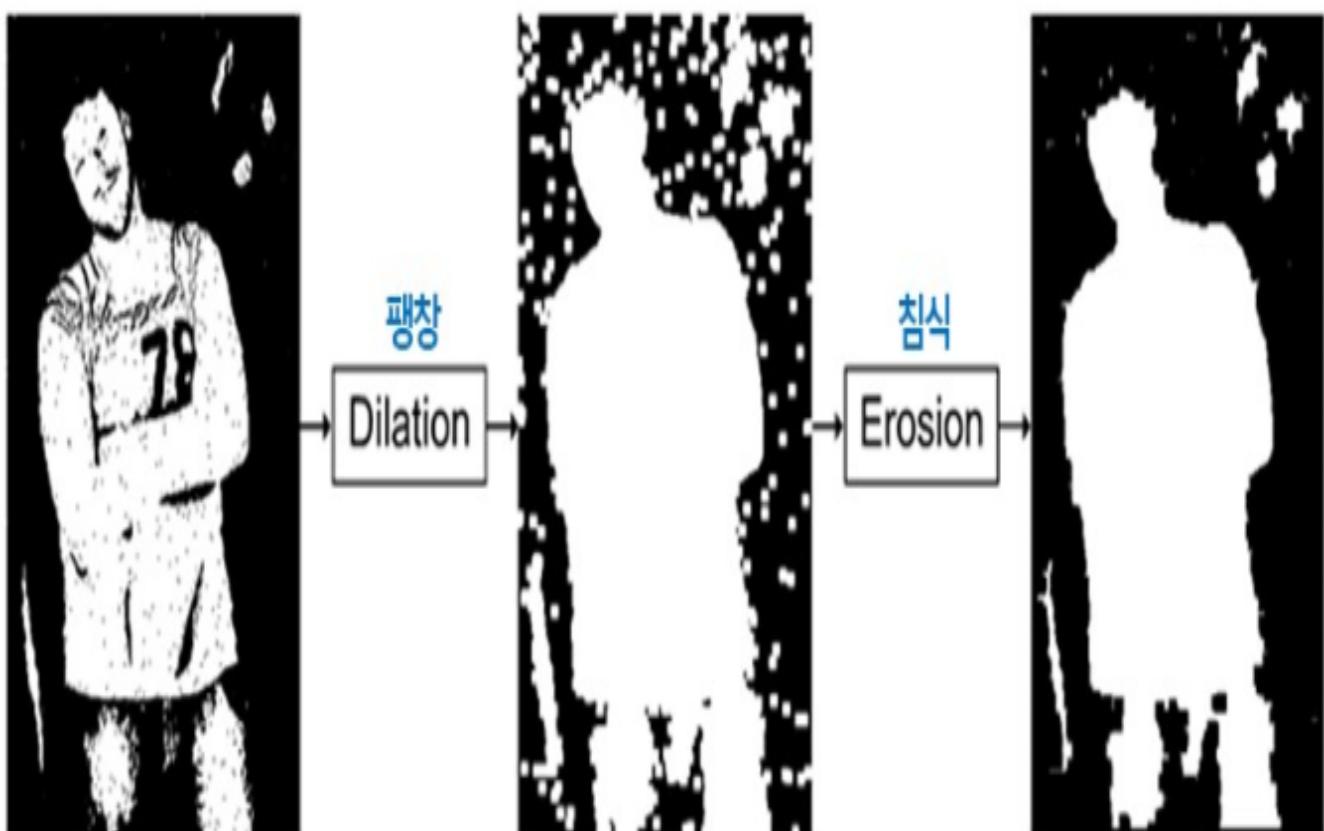


입력 이미지

닫힘 이미지

닫힘 연산

▪ 닫힘 연산



출처 : <http://what-when-how.com/introduction-to-video-and-image-processing/morphology-introduction-to-video-and-image-processing-part-2/>