

# CHAPTER 07

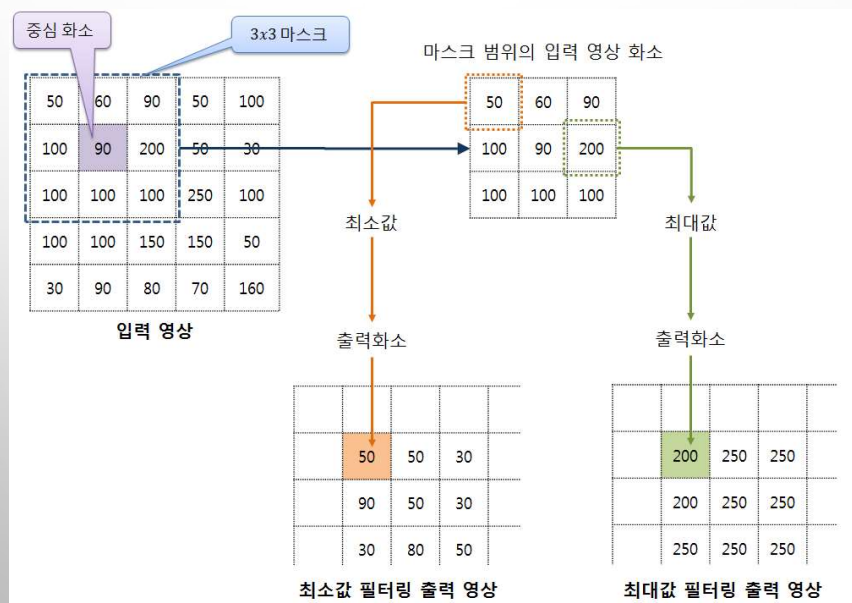
## 회선(Convolution)

### 7.4 모폴로지(MORPHOLOGY)

1

### 7.3.1 최댓값/최솟값 필터링

- ◆입력 영상의 해당 화소(중심화소)에서 마스크로 씌워진 영역의 입력화소들을 가져와서 그 중에 최댓값/최솟값을 출력화소로 결정하는 방법



2

## 7.3.1 최댓값/최솟값 필터링

### ◆ 최댓값 필터링

- ❖ 가장 큰 값인 밝은 색들로 출력화소가 구성
- ❖ 돌출되는 어두운 값이 제거 전체적으로 밝은 영상이 됨 => 팽창

### ◆ 최솟값 필터링

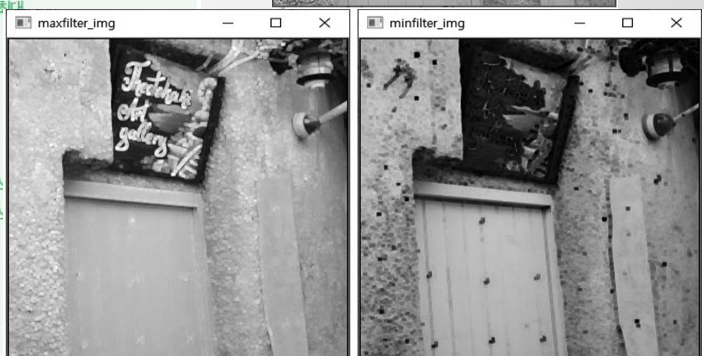
- ❖ 가장 작은 값들인 어두운 색들로 출력화소가 구성
- ❖ 돌출되는 밝은 값들이 제거되며, 전체적으로 어두운 영상 됨 => 침식

3

예제 7.3.1 최솟값-최댓값 필터링 - 09.filter\_minMax.py

```
01 import numpy as np, cv2
02
03 def minmax_filter(image, ksize, mode): # 최솟값&최댓값 필터링 함수
04     rows, cols = image.shape[:2]
05     dst = np.zeros((rows, cols), np.uint8)
06     center = ksize // 2 # 마스크 절반 크기
07
08     for i in range(center, rows - center): # 입력영상 끝단 마스크
09         for j in range(center, cols - center): # 절반 크기 조희 안함
10             ## 마스크 영역 행렬 처리 방식
11             y1, y2 = i - center, i + center + 1 # 마스크 높이 범위
12             x1, x2 = j - center, j + center + 1 # 마스크 너비 범위
13             mask = image[y1:y2, x1:x2] # 마스크 영역
14             dst[i, j] = cv2.minMaxLoc(mask)[mode] # 최소 or 최대
15     return dst
16
17 image = cv2.imread("images/min_max.jpg", cv2.IMREAD_GRAYSCALE)
18 if image is None: raise Exception("영상파일 읽기 오류")
19
20 minfilter_img = minmax_filter(image, 3, 0) # 3x3 마스크
21 maxfilter_img = minmax_filter(image, 3, 1) # 3x3 마스크
22
23 cv2.imshow("image", image)
24 cv2.imshow("minfilter_img", minfilter_img)
25 cv2.imshow("maxfilter_img", maxfilter_img)
26 cv2.waitKey(0)
```

## 7.3.1 최댓값/최솟값 필터링



4

## 7.4 모폴로지(morphology)

### ◆모폴로지 – 형태학

#### ❖ 생물학

- 생물형태의 기술과 그 법칙성의 탐구를 목적으로 일반적으로 해부학과 발생학을 합쳐서 형태학 이라 부름

#### ❖ 의학

- 인체 형태학이란 의미로 사용

#### ❖ 체육학

- 스포츠 운동 형태학이란 개념으로 사용

### ◆영상 처리에서 형태학

#### ❖ 영상의 객체들의 형태(shape)를 분석하고 처리하는 기법

#### ❖ 영상의 경계, 골격, 블록 등의 형태를 표현하는데 필요한 요소 추출

#### ❖ 영상 내에 존재하는 객체의 형태를 조금씩 변형시킴으로써 영상 내에서 불필요한 잡음 제거하거나 객체를 뚜렷하게 함

5

## 7.4.1 침식 연산

### ◆객체를 침식시키는 연산 (Erode)

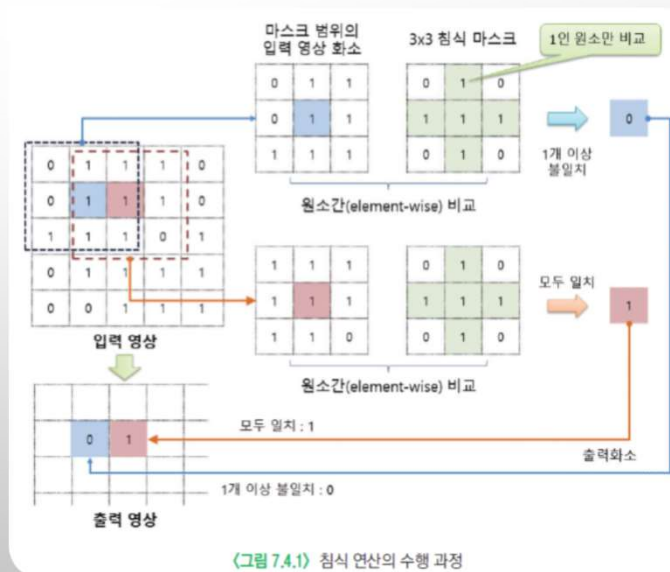
#### ❖ 객체의 크기 축소, 배경 확장

#### ❖ 영상 내에 존재하는 잡음 같은 작은 크기의 객체 제거 가능

#### ❖ 소금-후추 잡음과 같은 임펄스 잡음 제거

#### ❖ 그레이 영상 : 최소값 필터

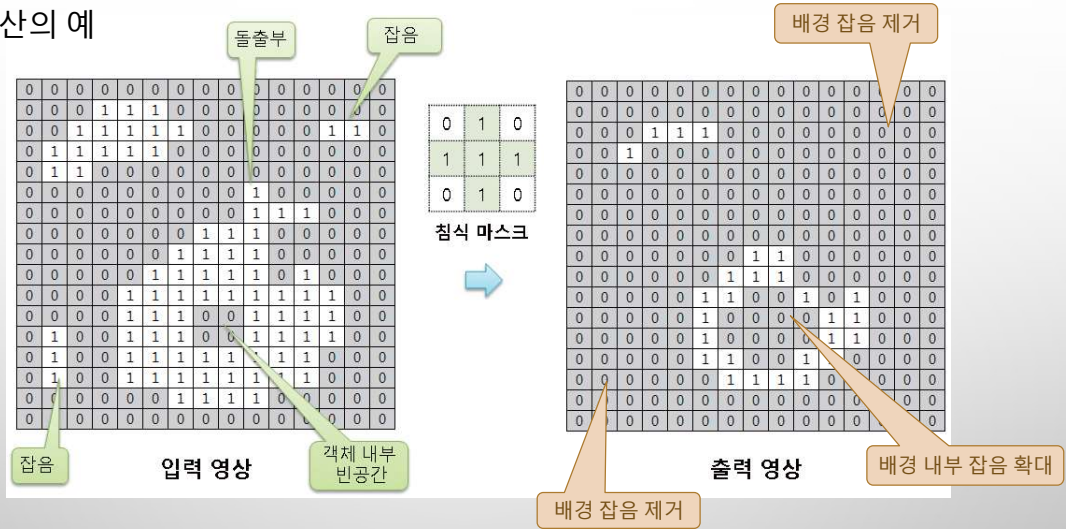
#### ❖ 이진 영상 :



6

## 7.4.1 침식 연산

### ◆ 침식 연산의 예



### 모폴로지 in OpenCV

cv2.morphologyEx() - 옵션 : MORPH\_ERODE  
cv2.erode()

7

## 7.4.1 침식 연산

### 예제 7.4.1 모폴로지 침식 연산 - 14.erode.py

```
01 import numpy as np, cv2
02
03 def erode(img, mask=None): # 침식 연산 함수
04     dst = np.zeros(img.shape, np.uint8)
05     if mask is None: mask = np.ones((3, 3), np.uint8)
06     ycenter, xcenter = np.divmod(mask.shape[:2], 2)[0] # 마스크 중심 좌표
07
08     mcnt = cv2.countNonZero(mask) # 마스크 1인 원소 개수
09     for i in range(ycenter, img.shape[0] - ycenter): # 입력 행렬 반복 순회
10         for j in range(xcenter, img.shape[1] - xcenter):
11             y1, y2 = i - ycenter, i + ycenter + 1 # 마스크 높이 범위
12             x1, x2 = j - xcenter, j + xcenter + 1 # 마스크 너비 범위
13             roi = img[y1:y2, x1:x2] # 마스크 영역
14             temp = cv2.bitwise_and(roi, mask) # 논리곱으로 일치 원소 지정
15             cnt = cv2.countNonZero(temp) # 일치 원소 개수 계산
16             dst[i, j] = 255 if (cnt == mcnt) else 0 # 출력 화소에 저장
17     return dst
18
19 image = cv2.imread("images/morph.jpg", cv2.IMREAD_GRAYSCALE)
20 if image is None: raise Exception("영상파일 읽기 오류")
```

8



## 7.4.1 침식 연산

```

19 image = cv2.imread("images/morph.jpg", cv2.IMREAD_GRAYSCALE)
20 if image is None: raise Exception("영상파일 읽기 오류")
21
22 data = [ 0, 1, 0,
23         1, 1, 1,
24         0, 1, 0]
25 mask = np.array(data, np.uint8).reshape(3, 3)
26 th_img = cv2.threshold(image, 128, 255, cv2.THRESH_BINARY)[1]
27
28 dst1 = erode(th_img, mask)
29 dst2 = cv2.erode(th_img, mask)
30 # dst2 = cv2.morphologyEx(th_img, cv2.MORPH_ERODE, mask)
31
32 cv2.imshow("image", image)
33 cv2.imshow("binary image", th_img)
34 cv2.imshow("User erode", dst1)
35 cv2.imshow("OpenCV erode", dst2)
36 cv2.waitKey(0)

```

침식 마스크 원소

# 마스크 원소 지정  
128보다 큰 화소는 255, 작으면 0으로 이진화

# 마스크 행렬 생성

# 영상 이진화

# 사용자 정의 침식 함수

# OpenCV의 침식 함수

# OpenCV의 침식 함수2

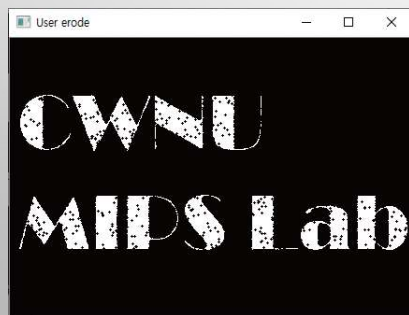
9

## 7.4.1 침식 연산

### ◆ 실행 결과



입력영상



배경 잡음 제거

객체 내부 잡음 확대

10

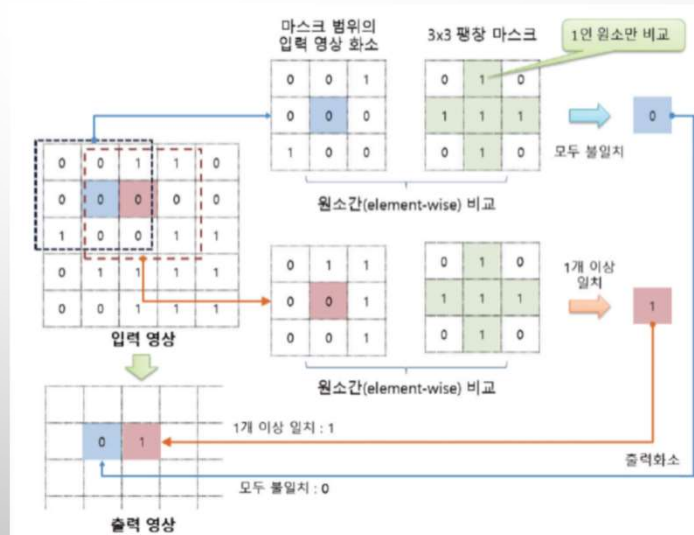
## 7.4.2 팽창 연산

## ◆객체를 팽창시키는 연산(Dilate)

- ❖ 객체의 최외곽 화소를 확장시키는 기능 → 객체 크기 확대, 배경 축소
- ❖ 객체 팽창으로 객체 내부의 빈 공간도 메워짐
  - → 객체 내부 잡음 제거

### ◆그레이 영상 : 최대값 필터

◆이진 영상 :

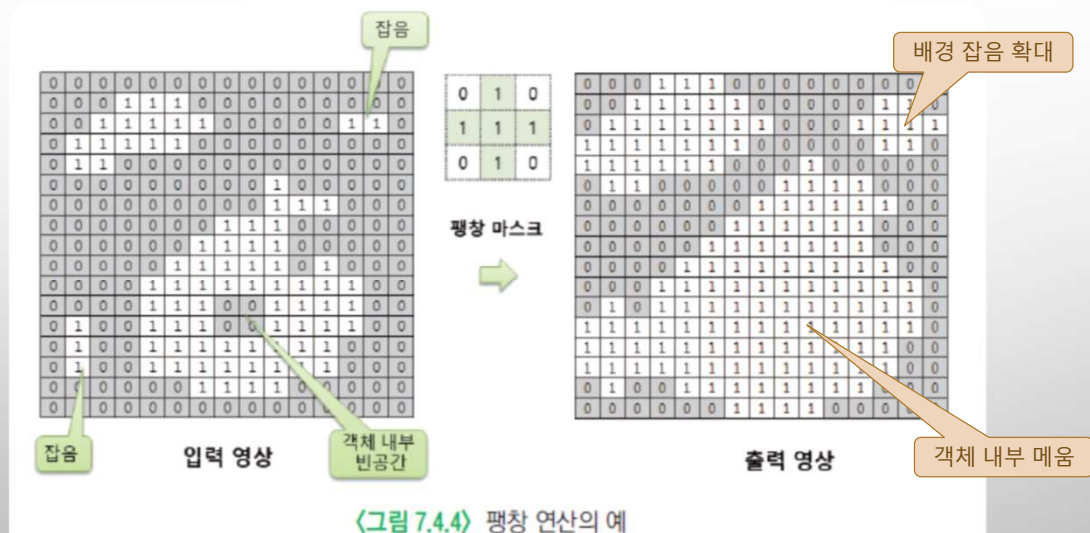


〈그림 7.4.3〉 팽창 연산의 수행 과정

11

## 7.4.2 팽창 연산

### ◆ 팽창 연산의 예



## 모폴로지 in OpenCV

```
cv2.morphologyEx() - 옵션 : MORPH_DILATE
cv2.dilate()
```

12

## 7.4.2 팽창 연산

예제 7.4.2 모폴로지 팽창 연산 - 15.dilate.py

```

01 import numpy as np, cv2
02
03 def dilate(img, mask):                # 팽창 수행 함수
04     dst = np.zeros(img.shape, np.uint8)
05     if mask is None: mask = np.ones((3, 3), np.uint8)
06     ycenter, xcenter = np.divmod(mask.shape[:2], 2)[0]    # 마스크 중심 좌표
07
08     for i in range(ycenter, img.shape[0] - ycenter):      # 입력 행렬 반복 순회
09         for j in range(xcenter, img.shape[1] - xcenter):
10             y1, y2 = i - ycenter, i + ycenter + 1        # 마스크 높이 범위
11             x1, x2 = j - xcenter, j + xcenter + 1        # 마스크 너비 범위
12             roi = img[y1:y2, x1:x2]                      # 입력 영상의 마스크 범위
13             temp = cv2.bitwise_and(roi, mask)
14             cnt = cv2.countNonZero(temp)
15             dst[i, j] = 0 if (cnt == 0) else 255          # 출력 화소에 저장
16     return dst
17
18 image = cv2.imread("images/morph.jpg", cv2.IMREAD_GRAYSCALE)

```

모두 불일치시 0, 하나라도 일치시 255,

13

## 7.4.2 팽창 연산

```

18 image = cv2.imread("images/morph.jpg", cv2.IMREAD_GRAYSCALE)
19 if image is None: raise Exception("영상파일 읽기 오류")
20
21 mask = np.array([[0, 1, 0],          # 마스크 계수 초기화
22                 [1, 1, 1],
23                 [0, 1, 0]]).astype('uint8')
24 th_img = cv2.threshold(image, 128, 255, cv2.THRESH_BINARY)[1] # 영상 이진화
25 dst1 = dilate(th_img, mask)      # 사용자 정의 팽창
26 dst2 = cv2.dilate(th_img, mask)  # OpenCV의 팽창
27 # dst2 = cv2.morphologyEx(th_img, cv2.MORPH_DILATE, mask)
28
29 cv2.imshow("User dilate", dst1)
30 cv2.imshow("OpenCV dilate", dst2)
31 cv2.waitKey(0)

```

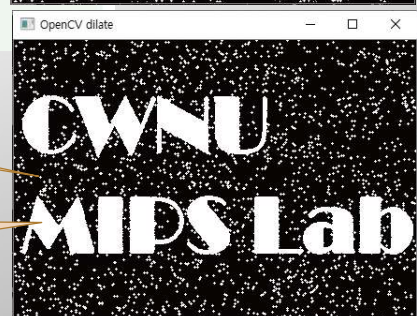
128보다 큰 화소는 255, 작으면 0으로 이진화

입력영상



배경 잡음 증가

객체 내부 잡음 제거



14



## 7.4.3 열림 연산과 닫힘 연산

◆ 침식 연산과 팽창 연산의 순서를 조합하여 수행

❖ 열림 연산(Open) - 침식 연산 → 팽창 연산

- 침식 연산으로 인해서 객체는 축소되고, 배경 부분의 미세한 잡음 제거
- 팽창 연산으로 인해서 축소되었던 객체들이 다시 원래 크기로

잡음 제거 후  
제자리로



〈그림 7.4.5〉 열림 연산의 과정

모폴로지 in OpenCV

`cv2.morphologyEx()` - 옵션 : `MORPH_OPEN`

15

## 7.4.3 열림 연산과 닫힘 연산

❖ 닫힘 연산(Close) : 팽창 연산 → 침식 연산

- 팽창 연산으로 객체가 확장되어서 객체 내부의 빈 공간이 메워짐
- 침식 연산으로 다음으로 확장되었던 객체의 크기가 원래대로 축소

물체 내부 빈  
공간 채운 후  
제자리로



〈그림 7.4.6〉 닫힘 연산의 과정

모폴로지 in OpenCV

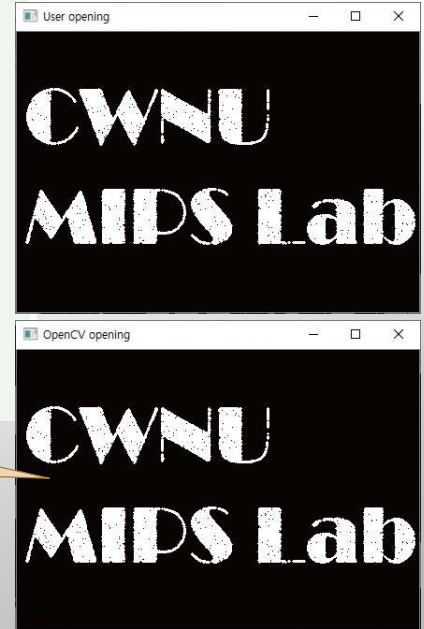
`cv2.morphologyEx()` - 옵션 : `MORPH_CLOSE`

16



## 7.4.3 열림 연산과 닫힘 연산

```
01 import numpy as np, cv2
02 from Common.filters import erode, dilate # filters 모듈의 저자 구현 함수 импорт
03
04 def opening(img, mask): # 열림 연산 함수
05     tmp = erode(img, mask) # 침식
06     dst = dilate(tmp, mask) # 팽창
07     return dst
08
09 def closing(img, mask): # 닫힘 연산 함수
10     tmp = dilate(img, mask) # 팽창
11     dst = erode(tmp, mask) # 침식
12     return dst
13
14 image = cv2.imread("images/morph.jpg", cv2.IMREAD_GRAYSCALE)
15 if image is None: raise Exception("영상파일 읽기 오류")
16
```

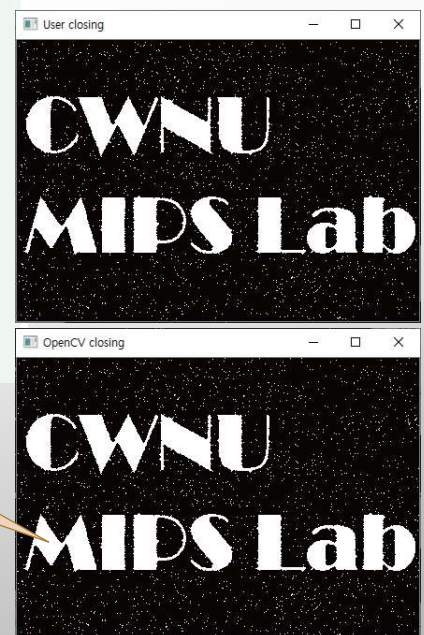


배경 잡음 제거

17

```
14 image = cv2.imread("images/morph.jpg", cv2.IMREAD_GRAYSCALE)
15 if image is None: raise Exception("영상파일 읽기 오류")
16
17 mask = np.array([[0, 1, 0], # 마스크 생성 및 초기화
18                 [1, 1, 1],
19                 [0, 1, 0]]).astype('uint8')
20 th_img = cv2.threshold(image, 128, 255, cv2.THRESH_BINARY)[1] # 영상 이진화
21
22 dst1 = opening(th_img, mask) # 저자 구현 열림 함수
23 dst2 = closing(th_img, mask) # 저자 구현 닫힘 함수
24 dst3 = cv2.morphologyEx(th_img, cv2.MORPH_OPEN, mask) # OpenCV의 열림 함수
25 dst4 = cv2.morphologyEx(th_img, cv2.MORPH_CLOSE, mask, iterations=1) # 닫힘 함수
26
27 cv2.imshow("User_opening", dst1); cv2.imshow("User_closing", dst2)
28 cv2.imshow("OpenCV_opening", dst3); cv2.imshow("OpenCV_closing", dst4)
29 cv2.waitKey(0)
```

## 7.4.3 열림 연산과 닫힘 연산



열림 연산 옵션

반복횟수

닫힘 연산 옵션

객체 내부 잡음 제거

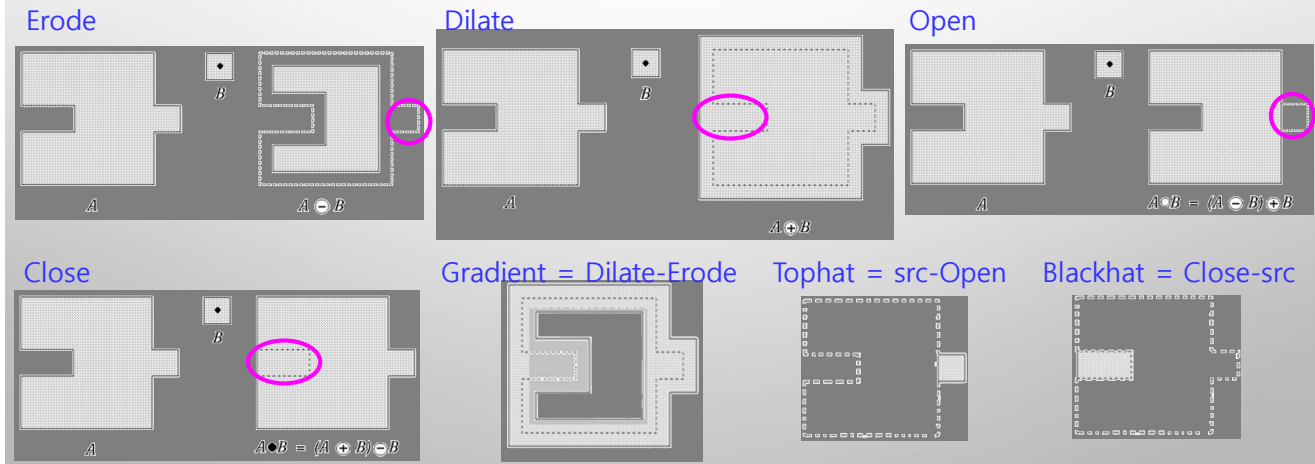
18

## 형태학적 처리

### ◆형태학적 처리 수행하기

➢ `cv2.morphologyEx(src, op, kernel[, dst[, anchor[, iterations[, borderType[, borderValue]]]])` → `dst`

● `op` : `CV_MOP_ERODE`, `CV_MOP_DILATE`, `CV_MOP_OPEN`, `CV_MOP_CLOSE`,  
`CV_MOP_GRADIENT`, `CV_MOP_TOPHAT`, `CV_MOP_BLACKHAT`



19

## 형태학적 처리

### ◆형태학적 처리 수행하기

➢ `cv2.getStructuringElement(shape, ksize[, anchor])` → `retval`

- `shape` - Element(마스크)의 모양.
  - `MORPH_RECT` : 사각형 모양
  - `MORPH_ELLIPSE` : 타원형 모양
  - `MORPH_CROSS` : 십자 모양
- `ksize` - structuring element(마스크) 크기
- `ksize = 5`인 경우

$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
---------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------

20

```

01 import numpy as np, cv2
02
03 while True:
04     no = int(input("차량 영상 번호( 0:종료 ) : "))
05     if no == 0: break
06
07     fname = "images/test_car/{0:02d}.jpg".format(no)
08     image = cv2.imread(fname, cv2.IMREAD_COLOR)
09     if image is None:
10         print(str(no) + "번 영상파일이 없습니다.")
11         continue
12
13     mask = np.ones((5, 17), np.uint8)
14     gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
15     gray = cv2.blur(gray, (5, 5))
16     gray = cv2.Sobel(gray, cv2.CV_8U, 1, 0, 5)
17
18     ## 이진화 및 닫힘 연산 수행
19     th_img = cv2.threshold(gray, 120, 255, cv2.THRESH_BINARY)[1]
20     morph = cv2.morphologyEx(th_img, cv2.MORPH_CLOSE, mask, iterations=3)
21
22     cv2.imshow("image", image)
23     cv2.imshow("binary image", th_img)
24     cv2.imshow("opening", morph)
25     cv2.waitKey()
    
```

◆차량사진에서 번호판 검출 위한 전처리

차량 영상 파일은 심화예제 프로젝트  
폴더 상단(..)에서 test\_car/ 폴더에 위치  
파일명은 숫자로 구성

차량영상 번호 입력

가로로 긴 마스크

수직(dx) 마스크 적용하여  
수직 에지 검출

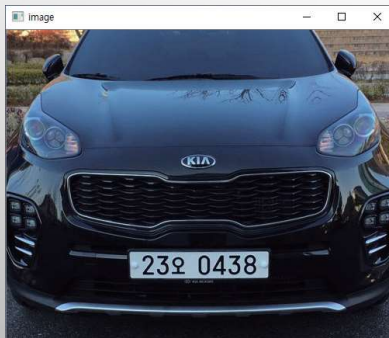
21

모폴로지 심화예제

◆실행결과

```

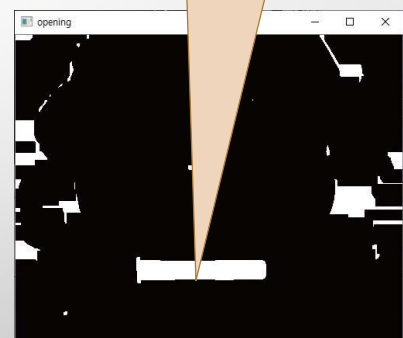
Run: 19.detect_plate
C:\Python\python.exe
D:/source/chap07/19.detect_plate.py
차량 영상 번호( 0:종료 ) : 5
차량 영상 번호( 0:종료 ) : 2
차량 영상 번호( 0:종료 ) : 1
|
    
```



입력영상



가로로 긴 마스크로  
침식 및 팽창 연산을 수행해서  
번호판과 비슷한 객체 내부 채움



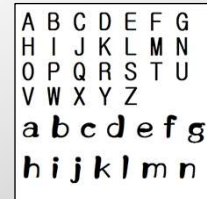
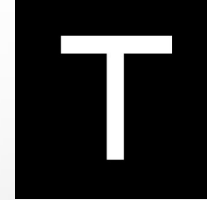
22

## 예제 : 골격화(SKELETON)

```
src = cv2.imread('./data/T.jpg', cv2.IMREAD_GRAYSCALE)
##src = cv2.imread('alphabet.bmp', cv2.IMREAD_GRAYSCALE)
##src = cv2.bitwise_not(src)

ret, A = cv2.threshold(src, 128, 255, cv2.THRESH_BINARY)
skel_dst = np.zeros(src.shape, np.uint8)

shapel=cv2.MORPH_CROSS
shape2=cv2.MORPH_RECT
B= cv2.getStructuringElement(shape=shapel, ksize=(3,3))
done = True
while done:
    erode = cv2.erode(A, B)
    ## opening = cv2.dilate(erode,B)
    opening = cv2.morphologyEx(erode, cv2.MORPH_OPEN, B)
    tmp = cv2.subtract(erode, opening) # cv2.absdiff(erode, opening)#TOPHAT
    skel_dst = cv2.bitwise_or(skel_dst, tmp)
    A = erode.copy()
    done = cv2.countNonZero(A) != 0
cv2.imshow('skeleton', skel_dst)
```



23

## 단원 요약

- ◆ 최대 최솟값 필터링은 마스크 범위에서 최솟값 혹은 최댓값을 출력화소로 결정한다. 최솟값 필터링은 영상이 전반적으로 어두워지며, 최댓값 필터링은 영상이 밝아진다.
- ◆ 모폴로지는 행태학적 방법을 영상 처리에 적용한 것으로서 침식과 팽창 연산이 있다. 침식연산은 객체의 크기가 축소되기 때문에 영상 내에 존재하는 작은 크기의 잡음을 제거하는 효과적이다. 팽창 연산은 객체의 크기가 확대되어 객체 내부의 빈 공간을 메우는 역할을 한다.
- ◆ 열림 연산은 침식 연산 수행 후에 팽창 연산을 수행한다. 침식 연산으로 객체는 축소되고, 배경의 잡음들은 제거되며, 팽창 연산으로 축소되었던 객체들이 원래 크기로 돌아간다.
- ◆ 닫힘 연산은 팽창 연산 수행 후에 침식 연산을 수행한다. 팽창 연산으로 객체가 확장되어 객체 내부의 빈 공간이 메워진다. 다음으로 침식 연산으로 확장되었던 객체의 크기가 원래대로 축소된다.

24