



가톨릭대학교
THE CATHOLIC UNIVERSITY OF KOREA

레이블링

미디어기술콘텐츠학과
강호철

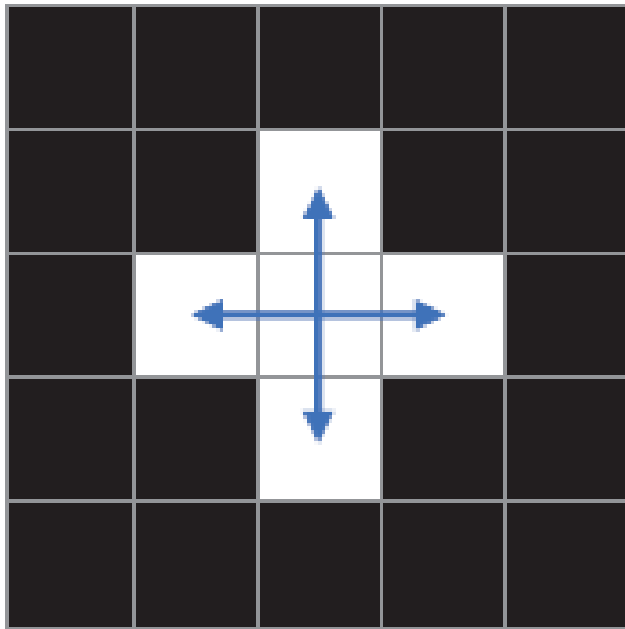
레이블링

- 레이블링의 이해
 - 배경과 객체를 구분 한 후 객체 간 구분 필요
 - 레이블링을 이용하여 객체 간 구분
 - 연결 구성 요소 레이블링 (CCL, Connected Component Labeling)
 - 객체 인식을 위한 전처리 과정으로 사용
 - 이진화 영상에서 수행 (배경: 검은색, 객체: 흰색)
 - 연결성 정의에 따라 결과가 달라질수 있음

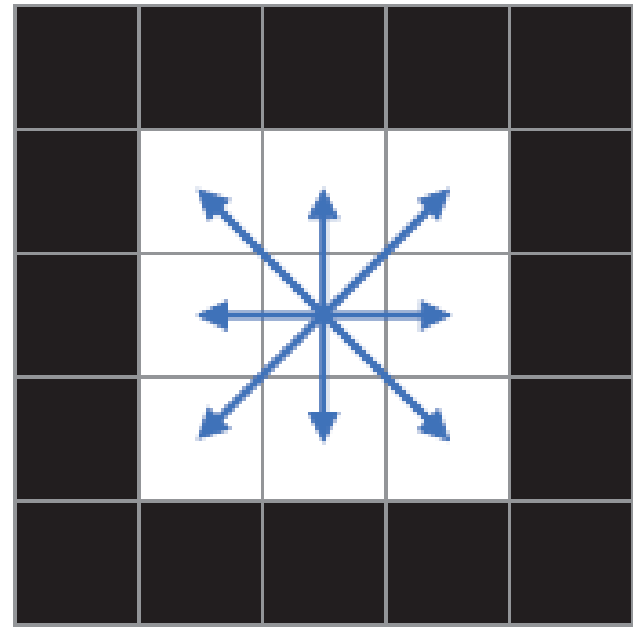


레이블링

- 레이블링의 이해
 - 4-way connectivity
 - 8-way connectivity



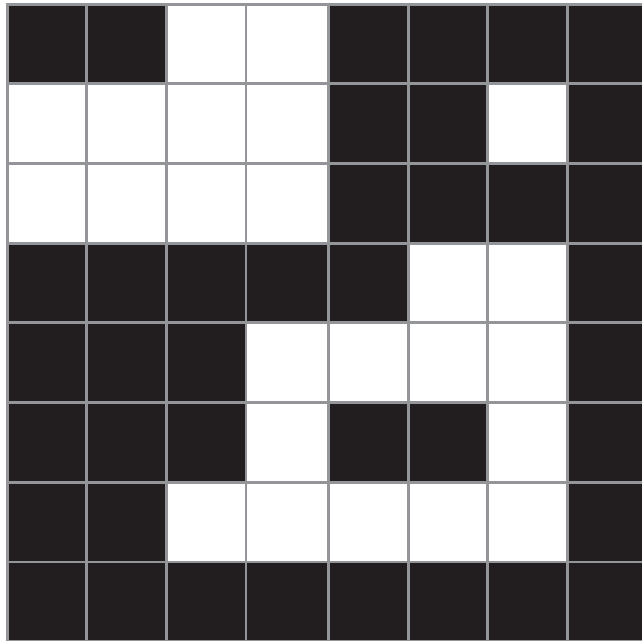
(a)



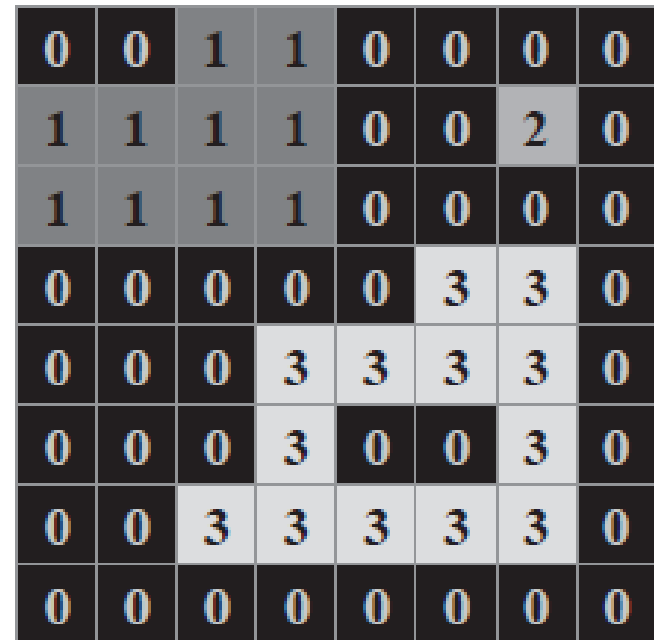
(b)

레이블링

- 레이블링의 이해
 - 레이블링의 입력과 출력



(a)



(b)

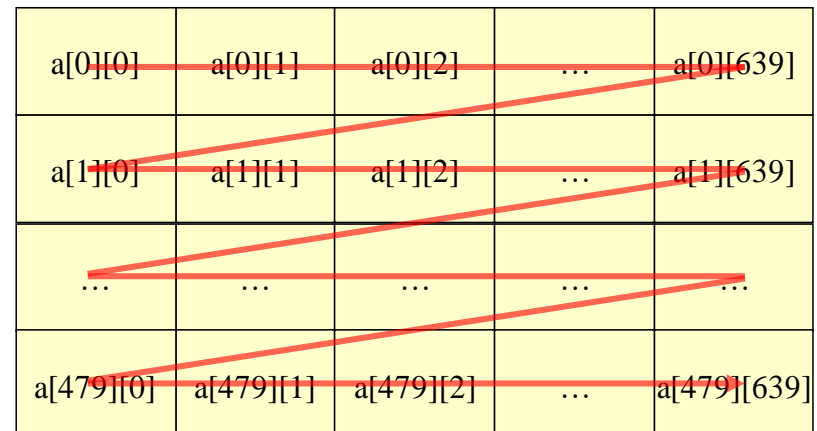
레이블링

■ 고전적 레이블링 기법

- 1966년 A. Rosenfeld 의 논문
- 등가 테이블(equivalent table)을 만들면서 영상을 두 번 스캔(scan)함으로써 레이블링을 수행
- 첫 번째 스캔에서는 레이블을 전파시키면서 등가 테이블을 만들고, 두 번째 스캔에서는 등가 테이블을 참조하여 각 픽셀에 고유의 레이블을 부여
- 4-이웃 연결성 고려

❖ 스캔: 영상의 모든 픽셀을 위에서 아래로, 왼쪽에서 오른쪽 방향으로 순서대로 방문하는 작업

a[0][0]	a[0][1]	a[0][2]	...	a[0][639]
a[1][0]	a[1][1]	a[1][2]	...	a[1][639]
...
a[479][0]	a[479][1]	a[479][2]	...	a[479][639]



레이블링

■ 고전적 레이블링 기법

임의의 위치에서 위쪽과 왼쪽 두 이웃 픽셀을 조사하여

1. 두 이웃 픽셀에 레이블이 존재하지 않은 경우,
 - A. 새로운 레이블을 지정하고,
 - B. 자기 자신의 레이블을 가리키는 새 등가 테이블 항목을 생성한다.
2. 두 이웃 픽셀 중 하나에만 레이블이 존재하는 경우
 - A. 이웃 픽셀과 동일한 레이블을 지정한다.
3. 두 이웃 픽셀 모두에 레이블이 존재하며, 서로 동일한 레이블인 경우
 - A. 이웃 픽셀과 동일한 레이블을 지정한다.
4. 두 이웃 픽셀 모두에 레이블이 존재하며, 서로 다른 레이블인 경우
 - A. 두 레이블 중 작은 번호의 레이블을 지정하고,
 - B. 두 레이블 중 큰 번호의 레이블이 작은 번호의 레이블을 가리키도록 등가 테이블을 조정한다.



레이블링

- 고전적 레이블링 기법
 - 등가 테이블 정리
 - 등가 테이블에 존재하는 두 개의 레이블 쌍 ($L_{\text{large}}, L_{\text{small}}$)이 서로 다른 경우, L_{small} 값을 $E_q(L_{\text{small}})$ 으로 변경
 - 이때 함수 $E_q(L)$ 은 등가 테이블에서 레이블 L 이 가리키고 있는 레이블을 반환하는 함수
 - 두 번째 스캔
 - 영상 전체를 다시 스캔하면서 객체에 해당하는 픽셀에 현재 매겨진 레이블 번호가 L_k 라면, 등가 테이블을 참조하여 $E_q(L_k)$ 값을 새 레이블로 할당

레이블링

- 고전적 레이블링 기법
 - 작은 크기의 영상에 대하여 알고리즘 검증
 - 테스트 영상: 8x8 크기의 이진 영상
 - 1: 오브젝트 픽셀, 빈칸: 배경 픽셀

		1		1		1	
	1	1	1	1		1	
	1	1	1	1		1	
						1	
		1		1	1	1	
	1	1	1	1	1	1	

레이블링

■ 고전적 레이블링 기법

		1		2		3	
	4	1	1	1		3	
	4	1	1	1		3	
						3	
		5		6	6	3	
	7	5	5	5	5	?	

(e)

1	1
2	1
3	3
4	1
5	5
6	5
7	5

		1		2		3	
	4	1	1	1		3	
	4	1	1	1		3	
						3	
		5		6	6	3	
	7	5	5	5	5	3	

(f)

1	1
2	1
3	3
4	1
5	3
6	3
7	5

		1		2		3	
	4	1	1	1		3	
	4	1	1	1		3	
						3	
		5		6	6	3	
	7	5	5	?	*	*	

(c)

1	1
2	1
3	3
4	1
5	5
6	3
7	5

		1		2		3	
	4	1	1	1		3	
	4	1	1	1		3	
						3	
		5		6	6	3	
	7	5	5	5	*	*	

(d)

1	1
2	1
3	3
4	1
5	5
6	5
7	5



레이블링

■ 고전적 레이블링 기법

		1		2		3	
	4	1	1	1		3	
	4	1	1	1		3	
						3	
		5		6	6	3	
	7	5	5	5	5	?	

(e)

1	1
2	1
3	3
4	1
5	5
6	5
7	5

		1		2		3	
	4	1	1	1		3	
	4	1	1	1		3	
						3	
		5		6	6	3	
	7	5	5	5	5	3	

(f)

1	1
2	1
3	3
4	1
5	3
6	3
7	5

		1		2		3	
	4	1	1	1		3	
	4	1	1	1		3	
						3	
		5		6	6	3	
	7	5	5	5	5	3	

(g)

1	1
2	1
3	3
4	1
5	3
6	3
7	3

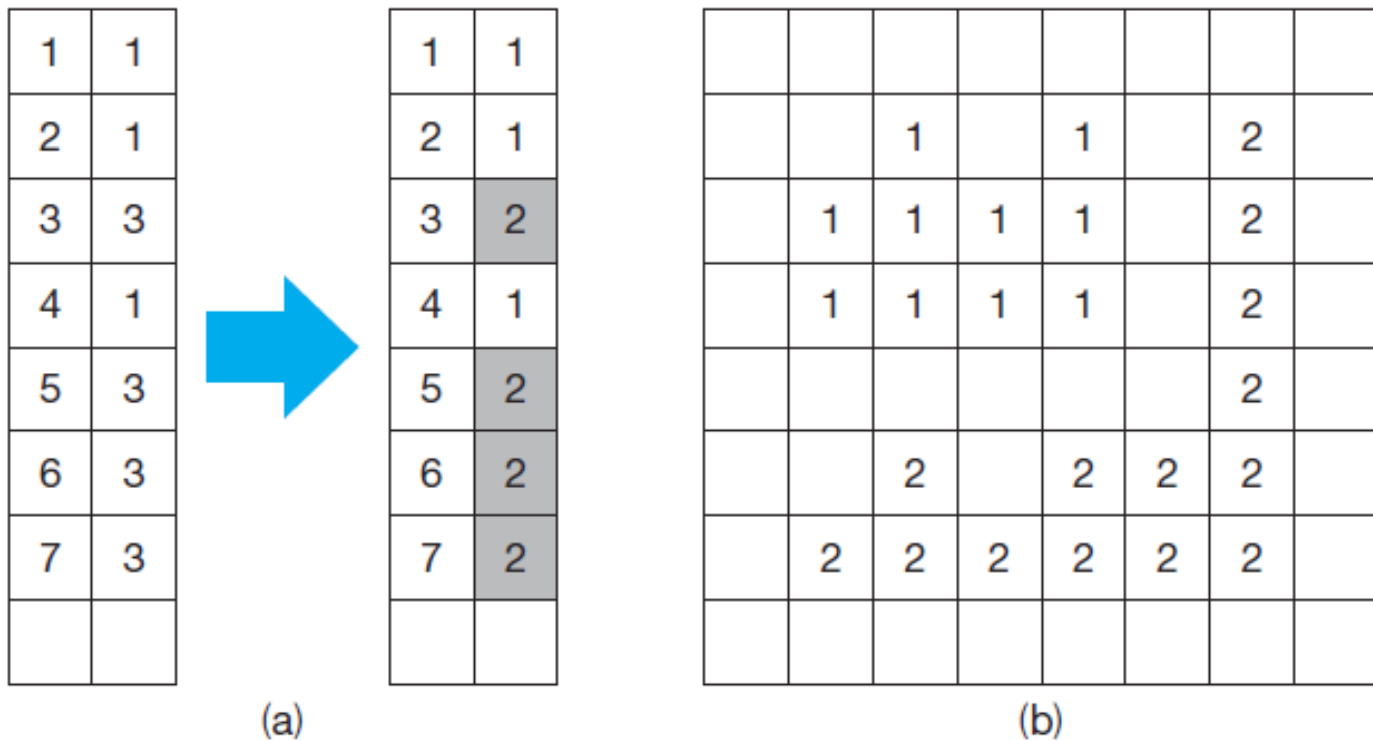
		1		1		3	
	1	1	1	1		3	
	1	1	1	1		3	
						3	
		3		3	3	3	
	3	3	3	3	3	3	

(h)

1	1
2	1
3	3
4	1
5	3
6	3
7	3

레이블링

- 고전적 레이블링 기법
 - 레이블이 1부터 순서대로 증가하도록 등가 테이블 정리



레이블링

■ 실습

■ OpenCV 함수

`cv2.connectedComponents(image[,labels[,connectivity[,itype]]]) :`

연결된 components의 개수를 반환하고 Labeling을 해준다.

parameter

- image: input
- labels: output
- connectivity: 4 or 8이웃 연결성
- itype: output image label type

`cv2.connectedComponentsWithStats(image[,labels[,stats[,centroids[,c`

위의 기능 및 중심 좌표 및 범위를 알려주기 때문에 Component를
특정화 시킬 수 있다.

return

- stats: x,y,width,height,area
- centroids: 중심 좌표

출처: [https://wjddy66.github.io/opencv/OpenCV\(6\)/#%EC%97%B0%EA%B2%B0%EC%9A%94%EC%86%8C-%EA%B2%80%EC%B6%9C](https://wjddy66.github.io/opencv/OpenCV(6)/#%EC%97%B0%EA%B2%B0%EC%9A%94%EC%86%8C-%EA%B2%80%EC%B6%9C)



레이블링

- 실습
 - OpenCV 함수

labels

0	0	1	1	0	0	0	0
1	1	1	1	0	0	2	0
1	1	1	1	0	0	0	0
0	0	0	0	0	3	3	0
0	0	0	3	3	3	3	0
0	0	0	3	0	0	3	0
0	0	3	3	3	3	3	0
0	0	0	0	0	0	0	0

(a)

stats

0	0	8	8	38	← 배경
0	0	4	3	10	← 1번 객체
6	1	1	1	1	← 2번 객체
2	3	5	4	14	← 3번 객체

바운딩 박스 정보
(x, y, width, height)

면적

(b)

centroids

3.615	3.692	← 배경
1.7	1.2	← 1번 객체
6	1	← 2번 객체
4.285	4.785	← 3번 객체

무게 중심의 x 좌표

무게 중심의 y 좌표

(c)

화이트 보드



영상처리 프로그래밍 기초

- Python으로 배우는 OpenCV 프로그래밍
 - 김동근 지음
 - 가메출판사, 2018
- OpenCV4 로 배우는 컴퓨터 비전과 머신러닝
 - 황선규 지음
 - 길벗, 2019
- Visual C++ 영상처리 프로그래밍
 - 황선규 지음
 - 길벗, 2015

