



가톨릭대학교
THE CATHOLIC UNIVERSITY OF KOREA

외곽선 검출

미디어기술콘텐츠학과
강호철

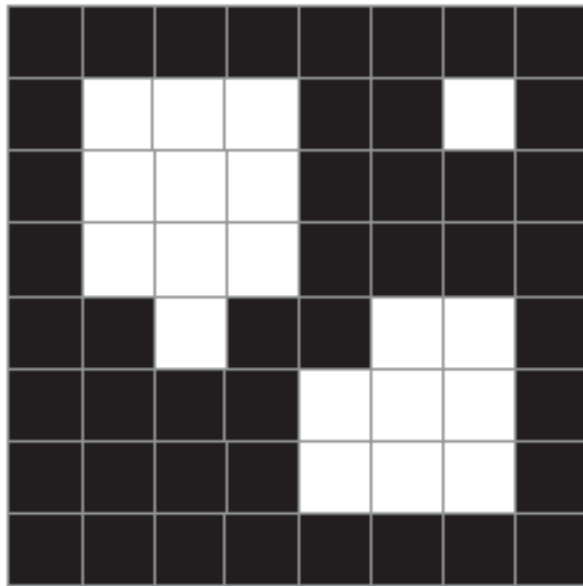
외곽선 검출

- 외곽선 검출의 이해
 - 객체의 모양을 분석하기 위해 외곽선 검출 필요
 - 외곽선(contour)은 객체 영역 픽셀 중 배경 영역과 인접한 일련의 픽셀 의미
 - 흰색 객체 영역에서 가장 최 외곽에 있는 픽셀을 찾아 외곽선으로 정의
 - 객체 안에 홀(hole)이 존재한다면 홀에 대한 외곽선도 검출 될수 있음
 - 바깥쪽 외곽선과 안쪽 외곽선으로 구분됨

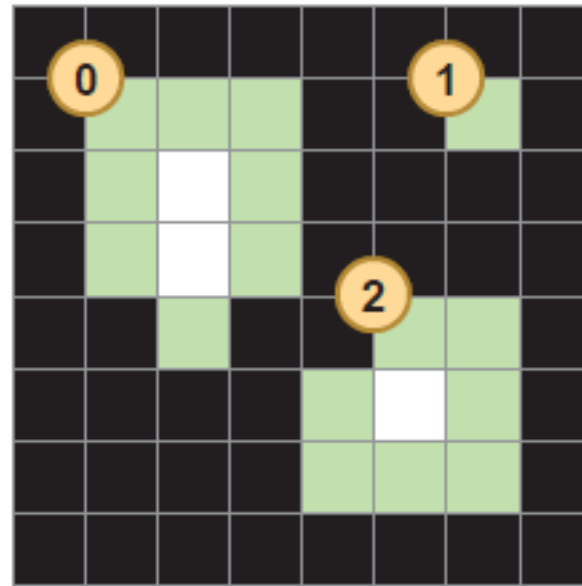


외곽선 검출

- 외곽선 검출의 이해
 - 외곽선은 여러 개의 점으로 구성됨
 - 테스트 영상 외곽선 검출 8 x 8



(a)



(b)

외곽선 검출

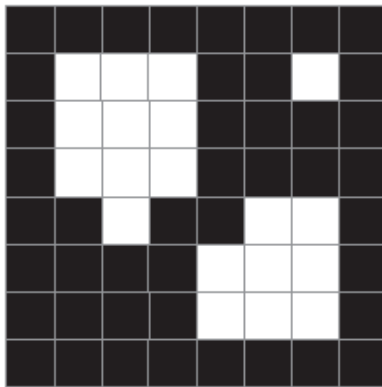
- 외곽선 검출의 이해

- 외곽선 점 좌표

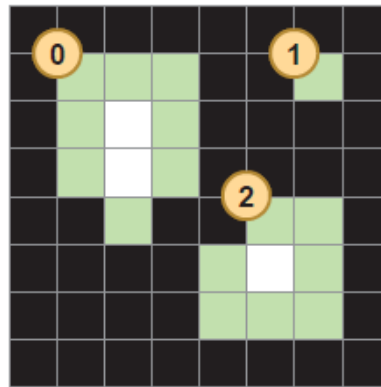
contours[0] : [1, 1], [1, 2], [1, 3], [2, 4], [3, 3], [3, 2], [3, 1], [2, 1]

contours[1] : [6, 1]

contours[2] : [5, 4], [4, 5], [4, 6], [5, 6], [6, 6], [6, 5], [6, 4]



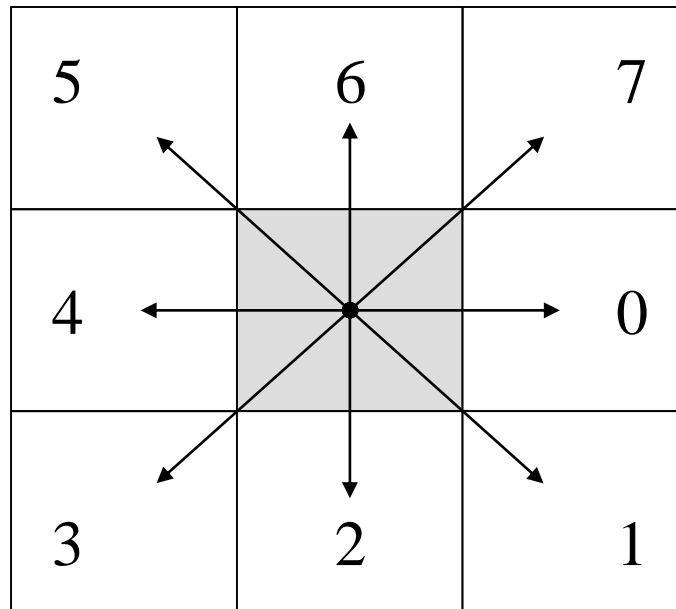
(a)



(b)

외곽선 검출

- 외곽선 검출 알고리즘
 - 객체의 외곽선을 따라 이동하는 기법
 - 객체의 경계선 추적이라고도 함
 - 8방향 연결성 고려 외곽선 추적 진행 방향



외곽선 검출

■ 외곽선 검출 알고리즘

1. 영상을 위에서 아래로, 왼쪽에서 오른쪽으로 스캔하면서 객체 픽셀을 찾는다. 객체 픽셀을 찾으면 이 좌표를 시작으로 외곽선 추적을 시작한다. 초기 외곽선 추적 진행 방향은 $d = 0$ 으로 설정한다.
2. 외곽선 추적 진행 방향에 객체 픽셀이 존재하는지를 판단한다.
3. 만약 진행 방향에 객체 픽셀이 존재하면 해당 픽셀로 이동한다.
 - A. 외곽선 추적 방향을 $d = d - 2$ 로 변화시키고 2번으로 간다.
4. 만약 진행 방향에 객체 픽셀이 존재하지 않으면 외곽선 추적 방향을 $d = d + 1$ 로 변화시키고, 2번으로 간다.
 - A. 모든 방향에 대하여 객체 픽셀이 존재하지 않으면 1 픽셀 짜리 객체이므로, 외곽선 추적을 종료한다.
5. 외곽선 추적 중, 현재 픽셀 위치가 외곽선 추적 시작 좌표와 같고, 진행 방향이 0인 경우 외곽선 추적을 종료한다.



외곽선 검출

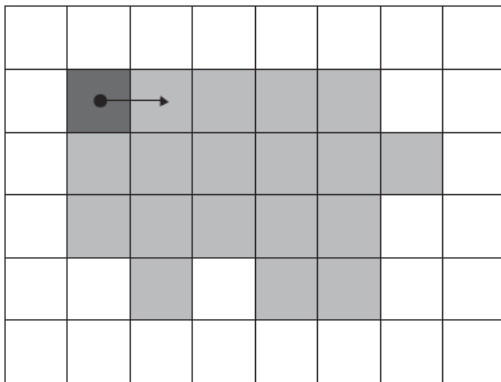
- 외곽선 검출 알고리즘
 - 테스트 8x6 이진 영상

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

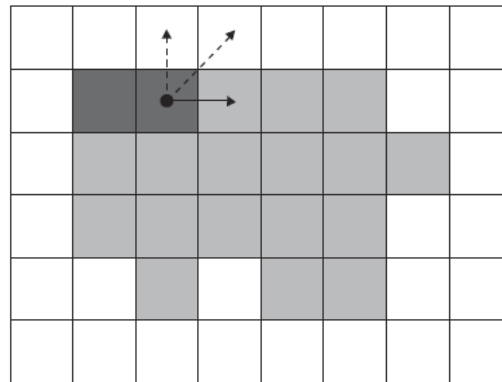
외곽선 검출

■ 외곽선 검출 알고리즘

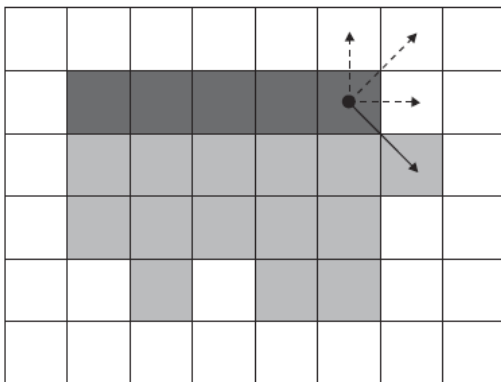
1. 영상을 위에서 아래로, 왼쪽에서 오른쪽으로 스캔하면서 객체 픽셀을 찾는다. 객체 픽셀을 찾으면 이 좌표를 시작으로 외곽선 추적을 시작한다. 초기 외곽선 추적 진행 방향은 $d = 0$ 으로 설정한다.
2. 외곽선 추적 진행 방향에 객체 픽셀이 존재하는지를 판단한다.
3. 만약 진행 방향에 객체 픽셀이 존재하면 해당 픽셀로 이동한다.
A. 외곽선 추적 방향을 $d = d - 2$ 로 변화시키고 2번으로 간다.
4. 만약 진행 방향에 객체 픽셀이 존재하지 않으면 외곽선 추적 방향을 $d = d + 1$ 로 변화시키고, 2번으로 간다.
A. 모든 방향에 대하여 객체 픽셀이 존재하지 않으면 1 픽셀 짜리 객체이므로, 외곽선 추적을 종료한다.
5. 외곽선 추적 중, 현재 픽셀 위치가 외곽선 추적 시작 좌표와 같고, 진행 방향이 0인 경우 외곽선 추적을 종료한다.



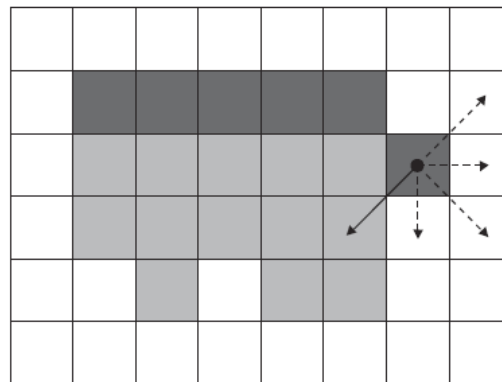
(a)



(b)



(c)

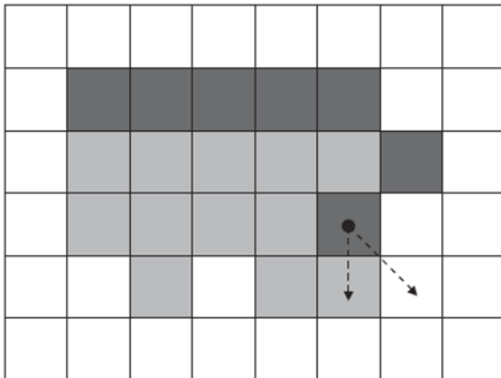


(d)

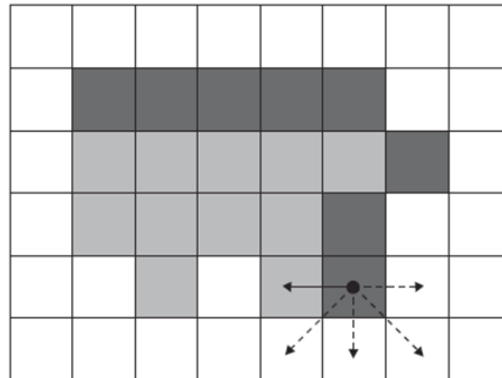
외곽선 검출

■ 외곽선 검출 알고리즘

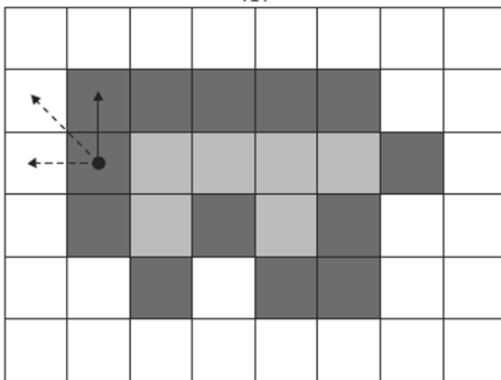
1. 영상을 위에서 아래로, 왼쪽에서 오른쪽으로 스캔하면서 객체 픽셀을 찾는다. 객체 픽셀을 찾으면 이 좌표를 시작으로 외곽선 추적을 시작한다. 초기 외곽선 추적 진행 방향은 $d = 0$ 으로 설정한다.
2. 외곽선 추적 진행 방향에 객체 픽셀이 존재하는지를 판단한다.
3. 만약 진행 방향에 객체 픽셀이 존재하면 해당 픽셀로 이동한다.
A. 외곽선 추적 방향을 $d = d - 2$ 로 변화시키고 2번으로 간다.
4. 만약 진행 방향에 객체 픽셀이 존재하지 않으면 외곽선 추적 방향을 $d = d + 1$ 로 변화시키고, 2번으로 간다.
A. 모든 방향에 대하여 객체 픽셀이 존재하지 않으면 1 픽셀 짜리 객체이므로, 외곽선 추적을 종료한다.
5. 외곽선 추적 중, 현재 픽셀 위치가 외곽선 추적 시작 좌표와 같고, 진행 방향이 0인 경우 외곽선 추적을 종료한다.



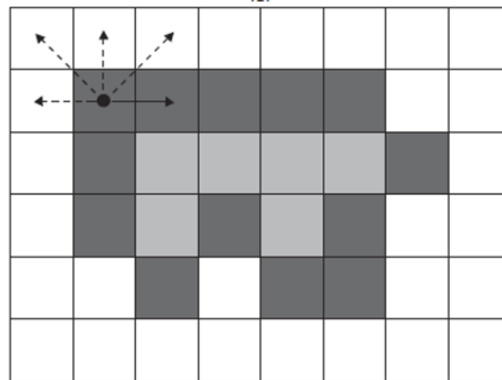
(f)



(g)



(h)



(i)

외곽선 검출

■ 실습

■ OpenCV 함수

`cv2.drawContours(image, contours, contourIdx, color[, thickness[, lineType[, hierarchy[, maxLevel[, offset]]]]) :`

컨투어 정보에서 비트맵 이미지 생성

parameter

- image: 원본 이미지
- contours: 컨투어 라인 정보
- contourIdx: 컨투어 라인 번호
- color: 색상

`cv2.findContours(image, mode, method[, contours[, hierarchy[, offset]])`
image의 contour를 반환한다.

parameter

- image: 흑백이미지 또는 이진화된 이미지
- mode: 컨투어를 찾는 방법
- cv2.RETR_EXTERNAL: 가장 외각의 윤곽선만 찾는다
- cv2.RETR_LIST: 모든 윤곽선을 찾는다. 계층관계를 설정하지 않는다.
- cv2.RETR_CCOMP: 2레벨 계층 구조로 모든 윤곽선을 찾는다.
- cv2.RETR_TREE: 모든 윤곽선을 계층적 트리 형태로 찾는다.
- method: 컨투어를 찾을 때 사용하는 근사화 방법
- cv2.CHAIN_APPROX_NONE: 체인 코드로 표현된 윤곽선의 모든 좌표를 반환한다.
- cv2.CHAIN_APPROX_SIMPLE: 윤곽선의 다각형 근사 좌표를 반환한다.
- cv2.CHAIN_APPROX_TC89_L1: Teh-Chin의 체인 코드 알고리즘으로 근사화 한다.
- cv2.CHAIN_APPROX_RC89_KCOS: 위와 같음

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)

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

