



Image Processing

Prof. KangSeung Lee



제 10 장

이진 영상의 처리



- Contents -

10.1 이진 영상 처리

10.2 영상의 이진화 처리

10.3 이진 영상의 성질



10.1 이진 영상 처리

- 문자, 도면 등 실제적으로 이진 영상으로 존재하는 것이 많고 이와 같은 영상은 관측시 많은 양의 데이터가 입력되어도 적당한 이진화 처리에 의하여 본래의 흑백 정보를 복원하는 것이 가능

10.2 영상의 이진화 처리

$$f_t(i,j) = \begin{cases} 1 & f(i,j) \geq t \text{의 경우} \\ 0 & f(i,j) < t \text{의 경우} \end{cases} \quad (10.1)$$

10.3 이진 영상의 성질

- 문명도 0의 배경부 픽셀을 “배경 픽셀”, 명도 1의 픽셀을 “대상 픽셀”이라고 부름

10.3.1 면적(Area)

$$A = \sum_{i=1}^n \sum_{j=1}^m f(i, j) \quad (10.3)$$

10.3 이진 영상의 성질

10.3.2 위치(Position)

- 면적의 중심을 구하기 위한 식

$$\bar{X} = \frac{\sum_{i=1}^n \sum_{j=1}^m j f(i, j)}{A} \quad (10.4)$$

$$\bar{Y} = \frac{\sum_{i=1}^n \sum_{j=1}^m i f(i, j)}{A} \quad (10.5)$$



10.3.3 모멘트(Moment)

- 모멘트란 어떤 축을 기준으로 한 값들의 분포의 척도
- 모멘트는 스칼라 양이고 패턴 인식이나 패턴 해석에서 대상 물체를 기술하는 우수한 특성
- 영상 $f(i,j)$ 의 $(p+q)$ 차 모멘트 m_{pq} 는

$$m_{pq} = \sum_i \sum_j i^p j^q f(i,j) \quad (10.6)$$

으로 정의



10.3.3 모멘트(Moment)

- 이진 영상인 경우 $f(i,j)$ 는 는 대상 물체 내 배경에서는 0이 됨
- 식(10.6)에서 이진 영상의 0차 모멘트 m_{00} 은 $f(i,j)$ 의 총합이 되므로 면적과 동일
- 1차 모멘트 m_{10} , m_{01} 을 m_{00} 으로 정규화하면 무게 중심 좌표를 얻을 수 있다

$$i_G = \frac{m_{10}}{m_{00}}, \quad j_G = \frac{m_{01}}{m_{00}} \quad (10.7)$$

- 중심 근처에 있는 모멘트를 특히, 중심 모멘트(central moment)라고 불러

$$M_{pq} = \sum_i \sum_j (i - i_G)^p (j - j_G)^q f(i,j) \quad (10.8)$$

이 된다

10.3.3 모멘트(Moment)

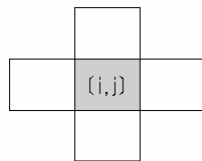
- 2차 모멘트는 관성 모멘트라고 불린다
- 영상 $f(i,j)$ 의 2차 중심 모멘트는 식 (10.8)에서 $p=0, q=2$ 또는 $p=2, q=0$ 로 하여 구하여지므로 다음과 같은 식이 된다

$$M_f = \sum_i \sum_j \{ (j-j_G)^2 + (i-i_G)^2 \} f(i,j) = M_{02} + M_{20} \quad (10.9)$$

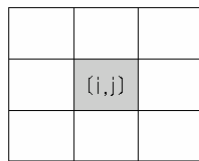
- 식(10.9)에서 중심을 원점으로 하는 경우 x 축 및 y 축에 대하여 2차 모멘트를 각각 μ_{02} 및 μ_{20} 으로 하면, 원점 근처에 있는 2차 모멘트 μ_2 는 다음과 같이 나타낸다

$$\mu_2 = \mu_{02} + \mu_{20} = \sum_i \sum_j j^2 f(i,j) + \sum_i \sum_j i^2 f(i,j) \quad (10.10)$$

10.3.4 4-이웃(Neighborhood)과 8-이웃



(a) 4-이웃



(b) 8-이웃

그림 10.2 4-이웃과 8-이웃

10.3.5 4연결과 8연결

- ▶ 그림 10.3에서 나타난 것과 같이 픽셀 P의 4이웃에 P와 같은 동일색의 픽셀 Q가 존재하는 경우, 픽셀 Q는 픽셀 P에 대해서 4연결로 접속되어 있다고 이야기할 수 있음
- ▶ P의 이웃한 8개의 픽셀중 동일색의 픽셀 R은 P에 대하여 8연결되어 있다고 말할 수 있음

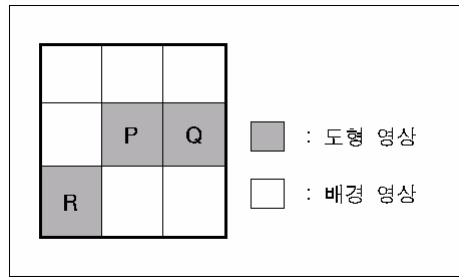


그림 10.3 4연결과 8연결

10.3.6 고립점, 내부점, 경계점

- $N=0$: 고립점
- $N=8$: 내부점
- $1 \leq N \leq 7$: 경계점

- ▶ 어떤 픽셀의 이웃 픽셀 중에서 자신과 동일한 색을 갖는 픽셀의 갯수 N 을 기준으로 다음과 같이 분류됨

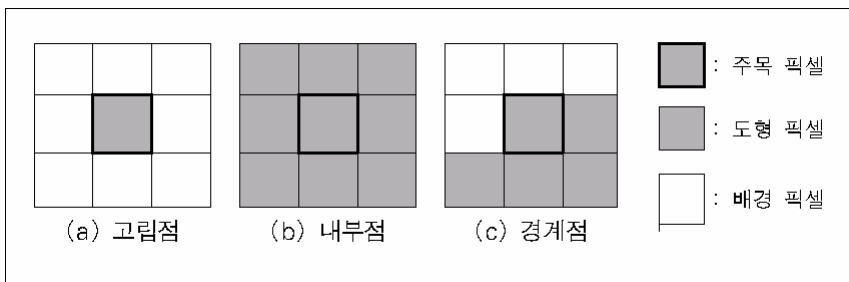


그림 10.4 도형 픽셀의 고립점, 내부점, 경계점

10.3.7 거리 척도(Distance)

➤ 거리 척도는 일반적으로 픽셀 p, q, r 에 대하여 다음의 3가지 성질을 만족함

1. $d(p, q) \geq 0$ and $d(p, q) = 0$ iff $p = q$
2. $d(p, q) = d(q, p)$
3. $d(p, r) \leq d(p, q) + d(q, r)$

➤ 유클리디안(Euclidean)

$$d_{Euclidean}([i_1, j_1], [i_2, j_2]) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2} \quad (10.12)$$

➤ 시티블록(city-block)

$$d_{city} = |i_1 - i_2| + |j_1 - j_2| \quad (10.13)$$

➤ 체스판(chessboard)

$$d_{chess} = \max(|i_1 - i_2|, |j_1 - j_2|) \quad (10.14)$$

10.3.7 거리 척도(Distance)

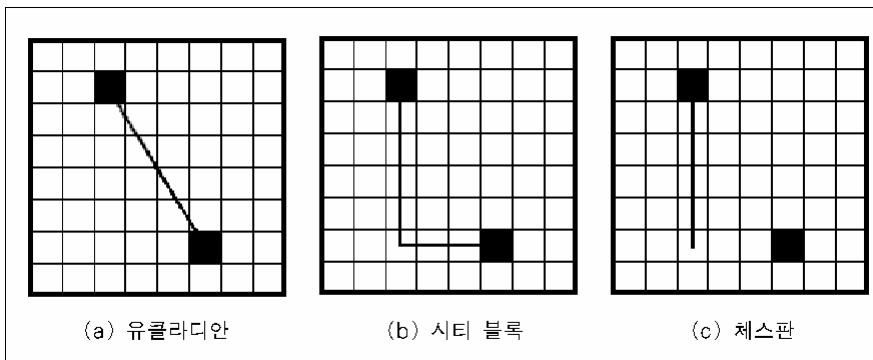
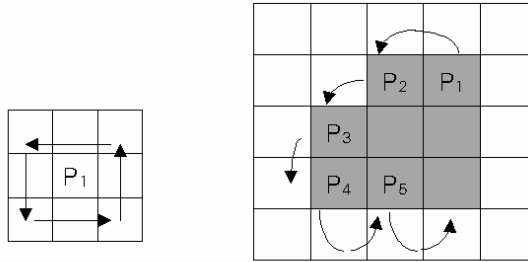


그림 10.5 여러 가지 거리 척도

10.3.8 경계선 추적

- ▶ 경계선 추적에 의해 면적, 둘레, 형태, 연결 성분의 포함 관계 등의 정보를 얻을 수 있음
- ▶ $p_n = p_1$ 로 된 경우에는 처리를 종료하여 픽셀 p_1 에서 p_{n-1} 까지의 픽셀 열을 경계선으로 한다.



(a) 시계 반대 방향으로 탐색 (b) 탐색 과정

그림 10.6-8-연결 경계선 추적 과정

10.3.9 체인 코드(Chain Code)

- ▶ 경계선 추적에 의하여 얻어진 경계선이나 또는 8-연결 또는 4-연결된 선들로 이루어진 그림을 저장하는 경우 가장 효율적인 방법은 맨 처음 픽셀의 위만 기록하고 나머지 픽셀들의 위치는 이전 픽셀에 대해 상대적으로 표현하는 것
- ▶ 방향이 숫자로 표시된 것을 방향 코드 또는 체인 코드(chain code)라고 부르며 디지털 선도형의 기본적인 기술 방법의 하나
- ▶ 대폭적인 데이터의 압축이 가능함
- ▶ 체인 코드에서는 방향 변화가 8가지가 있고 따라서 이를 표현하는 데는 3비트면 충분함

10.3.9 체인 코드(Chain Code)

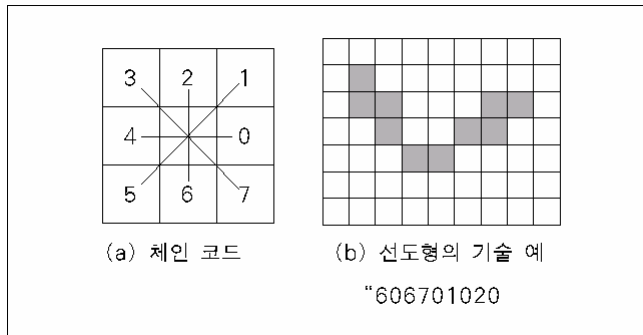


그림 10.7 체인 코드

10.3.10 세선화(Thining)

- 도형에서 선 폭을 섬세하게 하여 한 픽셀로 이루어진 중심선을 추출하는 조작을 세선화(thining)라고 한함
- 세선화는 원래 도형의 연결성은 변화시키지 않고, 선으로 변화시키는 것이 필요함
- 글씨나 도면 등에서 선의 구조를 해석하는데 불가결한 처리

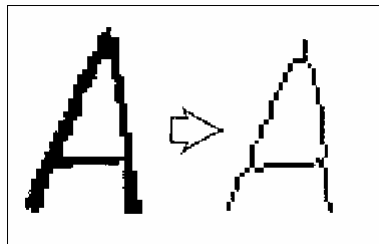


그림 10.8 세선화의 예



10.3.10 세션화(Thining)

- 흔히 사용되는 알고리즘은 영상 안의 각 픽셀을 의 이웃 윈도우 안에서 조사하여 각 영역이 세션화할 때까지 각 영역의 경계선을 한번에 한 픽셀 두께씩 벗겨내는 것이다. 이런 처리가 반복적으로 되풀이됨
- 형태 처리 연산에서 침식(erosion) 연산과 비슷함
- 그러나 침식은 어떤 영역을 완전히 없애 버릴 수가 있지만 세션화는 그래서 안됨
- 세션화 알고리즘이 지켜야될 제약점
 - 한 픽셀 두께의 영역으로 되어 있어야 한다.
 - 영역의 단면의 중심에 위치해야 한다.
 - 세션을 이루고 있는 픽셀들은 서로 서로 연결되어 있어야 한다.



10.3.10 세션화(Thining)

□ 여기서는 많이 쓰이는 Zhang-Suen 알고리즘에 대하여 살펴보자

- 기본적인 아이디어는 한 픽셀의 8개의 이웃 픽셀을 검사하여 그 픽셀을 제거할 수 있는지 결정을 내리는 것
- 픽셀을 제거하기 위한 4가지의 조건
 - 여기서는 간단한 2가지 조건만 설명함
 - 2개 이상, 6개 이하 이웃 픽셀이 있어야 한다. 이 조건은 끝점이나 영역 내부의 점은 지워서는 안된다는 것을 의미한다.
 - 두 개 영역을 연결하는 픽셀은 지워서는 안된다. 이 픽셀을 지운다면 영역 갯수가 달라지게 된다.

10.3.11 연결 성분(Connected Component)

- 이진 영상 중에서 서로 연결되어 있는 픽셀의 집합을 연결 성분 (connected component)라고 함
- 연결 성분에서 다른 연결 성분이 내부에 포함하는 경우를 홀 (hole)라고 부름
- 같은 연결 성분에 속하는 픽셀에 같은 라벨(번호)을 할당하고, 다른 연결 성분에는 서로 다른 레벨을 할당하는 조작을 연결 성분 라벨링(connected component labeling)이라 함
- 연결 성분의 속성을 해석하기 전에 각 성분을 추출하는 라벨링 조작이 필요

10.3.11 연결 성분(Connected Component)

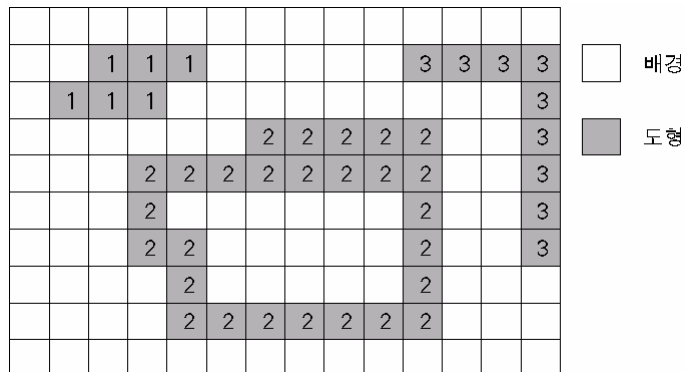


그림 10.9 연결 성분에 대한 라벨링

10.3.11 연결 성분(Connected Component)

연결 성분 라벨링의 2가지의 알고리즘

□ 재귀 알고리즘

알고리즘: 재귀 연결 성분 알고리즘

1. 영상을 스캔하여 라벨링되어 있지 않은 도형 픽셀을 찾아 새로운 라벨을 부여한다.
2. 재귀적으로 라벨을 모든 이웃의 도형 픽셀(4-이웃, 8-이웃)에 부여한다.
3. 더 이상 라벨링되어 있지 않은 도형 픽셀이 없으면 멈춘다.
4. 단계 1로 간다.

➤ 재귀 알고리즘은 하나의 CPU만을 가진 순차 컴퓨터에서는 매우 비효율적이지만 병렬 컴퓨터에서는 많이 쓰임

10.3.11 연결 성분(Connected Component)

□ 순차 알고리즘

➤ 알고리즘은 어떤 한순간에 영상의 2개의 행만을 가지고 처리하므로 영상이 디스크의 파일 형태로 저장되어 있거나 메모리가 작아 전체 영상을 불러올 수 없는 경우에도 사용할 수 있음

10.3.11 연결 성분(Connected Component)

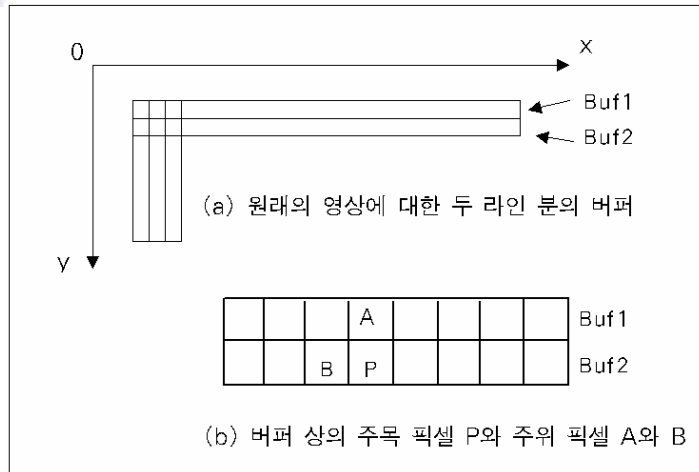


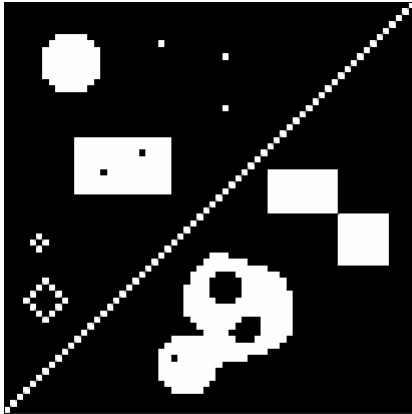
그림 10.10 연결 성분 라벨링 알고리즘

10.3.11 연결 성분(Connected Component)

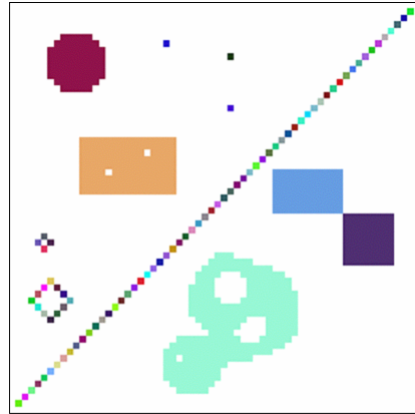
알고리즘: 4-연결을 이용한 순차 연결 성분 알고리즘

1. 영상을 위에서 아래로, 왼쪽으로 오른쪽으로 스캔한다.
2. 주목 픽셀이 1이면
 - a. 위쪽과 왼쪽 픽셀중 하나만이 라벨을 가지면 그 라벨을 주목 픽셀에 부여한다.
 - b. 위쪽과 왼쪽 픽셀이 모두 같은 라벨을 가지면 그 라벨을 주목 픽셀에 부여한다.
 - c. 위쪽과 왼쪽 픽셀이 다른 라벨을 가지면 위쪽의 라벨을 동치 테이블에 동치 라벨로 입력한다.
 - d. 위의 경우가 아니면 이 픽셀에 새로운 라벨을 부여하고 동치 테이블에 이 라벨을 입력한다.
3. 고려해야 할 더 이상의 픽셀이 없으면 멈춘다.
4. 동치 테이블에서 각 동치 라벨 집합에서 최소의 라벨을 찾는다.
5. 영상을 조사하여 각 라벨을 동치 집합의 최소 라벨로 바꾼다.

10.3.11 연결 성분(Connected Component)

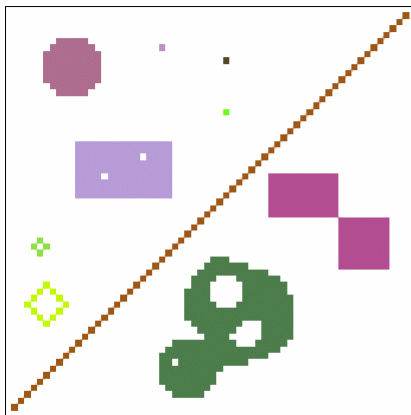


(a)



(b)

10.3.11 연결 성분(Connected Component)



(c)

- (a) 입력 이진 영상
- (b) 4-연결 성분 라벨링
- (c) 8-연결 성분 라벨링

그림 10.11 연결 성분 라벨링



10.3.12 투상(Projection)

- 이진 영상을 수평이나 수직 방향으로 투상(projection)하는 것은 수평이나 수직 방향의 라인상에 존재하는 픽셀값이 1인 픽셀의 갯수를 세는 것
- 이진 영상의 수평 투상 $H[i]$ 와 수직 투상 $V[j]$ 은 다음 식으로

$$H[i] = \sum_{j=1}^m B[i, j]$$

$$V[j] = \sum_{i=1}^n B[i, j]$$

- 투상의 성질로 원영상의 1차 모멘트와 투상의 1차 모멘트는 같다는 것을 들 수 있음



10.3.12 투상(Projection)

$$Area = \sum_{j=1}^m V[j] = \sum_{i=1}^n H[i]$$

$$\bar{y} = \frac{\sum_{i=1}^n i \cdot H[i]}{A}$$

$$\bar{x} = \frac{\sum_{j=1}^m j \cdot V[j]}{A}$$

- 투상은 물체의 인식을 위한 중요한 특징으로 이용된다. 투상은 보다 콤팩트한 표현이며 따라서 알고리즘을 보다 빠르게 만듦