

인 지 과 학

Korean Journal of Cognitive Science

Vol. 7, No. 1(1996)

가중치를 이용한 병렬 세선화 알고리즘

Parallel Thinning Algorithm using Weighted-Value

한낙희[†], 이필규^{††}

Nak Hee Han, Phil Kyu Rhee

요 약

본 논문은 문자인식, 문자검색, 도형인식 등에 있어서 필수 과정인 세선화 알고리즘에 대하여 논하였다. 접근 방법으로는 외곽선으로부터 특정 조건을 만족시키는 화소들을 동시에 제거해 가는 병렬 세선화 방법을 취했다. 제안된 알고리즘은 가중치 개념을 도입하여 기존의 알고리즘보다 정확성 및 수행속도의 향상을 성취하였다. 실험은 스캐너로 입력된 숫자, 영문자, 도형 등을 기준으로 했으며 특히 다른 알고리즘에서 정확하게 세선화 하지 못하는 데이터들을 사용하였다. 실험결과 모양이나 토폴로지를 잃지 않고 정확히 세선화가 가능했으며 잡음으로부터도 영향을 받지 않았다.

주제어 가중치, 삭제조건, 주변화소

ABSTRACT

This paper addresses an one-pass parallel thinning algorithm which shows effectiveness in both accuracy and speed. The proposed method is based on parallel iterative boundary removal. Image connectivity are preserved and the algorithms' performance is compared to other

[†] 인하대학교 전자계산공학과 석사 3차

인천광역시 남구 용현 4동

인하대학교 전자계산공학과

e-mail: pkrhee@dragon.inha.ac.kr

^{††} 인하대학교 전자계산공학과 부교수

인천광역시 남구 용현 4동

인하대학교 전자계산공학과

e-mail: pkrhee@dragon.inha.ac.kr

algorithms especially to parallel thinning algorithm which is the best parallel algorithm have been proposed. Evaluation result shows that the proposed algorithm compare favorably to others. The result shows exact thinning free from one pixel boundary noise and free from distortion of shape.

Keywords weight-value, elimination condition, boundary pixel

1. 서론

세선화는 문자인식, 회로기판의 결점 검사, 지문 인식 등 여러 가지 분야에서 광범위하게 사용되고 있는 중요한 기술이다. 세선화의 목적은 어떤 이미지 패턴의 자동인식을 위해 필요한 최소한만의 양으로 정보의 양을 줄이는 것이다. 세선화된 이미지는 끝점, 교차점, 성분들 사이의 연결성 등 중요한 특징들을 추출하는 것을 돋는다[Hung83]. 그래서 지금까지 많은 세선화 알고리즘이 개발되었다 [Arce89, Chin87, Chen88, Guo89, Hall89, Holt87, Wang86, Suen86, Jang92, Chin87]. 디지털 패턴의 세선화는 화소에 기반을 둔 반복적 외곽화소 제거 세선화 알고리즘[Chin87, Chen88, Hall89, Holt87, Wang86, Suen86, Jang92]와 화소에 기반을 두지 않는 세선화 알고리즘[Borg84, Rose68, Tori81]으로 크게 나눌 수 있다.

화소에 기반을 둔 반복적 외곽화소 제거 세선화 알고리즘은 골격선이 남을 때까지 외곽

화소를 반복적으로 제거해나가는 것이다. 화소를 기반으로 하지 않는 알고리즘은 각 화소들을 검사하지 않고 직접적으로 중앙선을 찾는 것이다. 화소를 기반으로 하지 않는 알고리즘의 장점은 계산 복잡도가 낮다는 것이나 반면에 스트로크가 스캔 라인에 평행할 때 잡음을 생성할 수 있다는 근본적인 단점이 있다 [Rock82]. 이 방법은 일반적인 응용에는 적당하지 않다. 특히 매우 다양한 모양이나 두께를 가지는 스트로크에는 부적당하다. 반복적 외곽화소 제거에 기반을 둔 세선화 알고리즘은 순차 알고리즘과 병렬 알고리즘으로 구분할 수 있다. 순차 알고리즘에 있어서 n차 화소의 제거는 그때까지 행해졌던 모든 화소 제거에 영향을 받는다. 병렬 알고리즘에 있어서는 n차 화소의 삭제는 단지 (n-1)번째 반복의 결과에만 의존한다[Loui92]. 그렇기 때문에 모든 화소들은 각각의 반복에서 병렬적이며 독립적으로 검사된다. 일반적인 병렬 세선화 알고리즘들의 단점은 연결성 유지를 위해서 이웃한 8화소뿐 아니라 다음 이웃(24화소)까지 검색해야

한다는 것이다. 이러한 문제를 해결하기 위해서 대부분의 병렬 세선화 알고리즘들은 두단계 또는 네단계의 부 반복 단계를 거친다 [Guo89]. 최근의 단일 반복 세선화 알고리즘으로 [Jang92]가 있다. 알고리즘 [Jang92]는 연결성 유지, 단위 두께(unit pixel width)로 의 세선화, 잡음처리능력등 좋은 세선화 능력을 가지는 알고리즘이다. 하지만 이 알고리즘 역시 일반적인 병렬 세선화 알고리즘의 문제점을 가지고 있어서 어떤 화소를 삭제하기 위해 24-이웃 까지 검사해야만 한다. 또한 연결성유지를 위해 삭제된 화소를 복원하는 과정을 거쳐야만 된다.

본 연구에서 제안된 세선화 알고리즘은 [Jang92]와 같은 반복적 외곽선 제거방법을 택한 단일 반복 병렬 세선화 알고리즘이지만 가중치 값을 도입하여 기존의 병렬세선화 알고리즘의 약점을 보완하고 8-이웃만을 검사하여 삭제할 화소를 효율적으로 검사한다. 8-이웃을 검사하는 것보다는 24-이웃을 검사하는 것이 3배의 오버헤드(overhead)가 따른다는 것은 명백한 사실이다. 어떤 화소를 삭제할 것인가 안할 것인가를 결정하기 위하여 비교해야 할 템플리트의 평균개수도 또한 중요한 것이다. 제안된 알고리즘은 최근의 우수한 병렬 세선화 알고리즘[Jang92]과 여러 가지 면에서 비교 평가될 것이다.

2장에서는 가중치의 개념과 세선화 전략을

설명하고, 3장에서는 제안된 세선화 알고리즘과 수행과정, 4장에서는 타 알고리즘과의 성능 평가와 비교, 그리고 5장에서는 결론에 대해서 논할 것이다.

2. 가중치와 세선화 전략

입력된 이미지는 그림 1(a)와 같이 1로 표시되는 검은 화소(유용한 데이터 부분)와 0으로 표시되는 백화소로 구분된다. 어떤 영상 데이터가 입력될 때 이것은 격자무늬 화소들로 유용한 데이터 부분인 흑화소 1과 공백인 백화소 0의 값으로 입력된다(그림 1의 (a)). 이때 각 화소 p 는 그림 2와 같이 1, 3, 5, 7의 4-이웃과 2, 4, 6, 8의 대각 이웃 4개를 갖는다. 이것들을 모두 어떤 화소 p 의 8-이웃이라고 한다.

가중치는 어느 흑화소에게 있어서 그 흑화소의 8-이웃에 위치한 흑화소의 개수를 나타낸다. 이렇게 할 경우 가장 작은 가중치 값은 0이 될 것이다. 즉 어떤 흑화소의 8-이웃 중 흑화소가 하나도 없는 경우 즉 독립된 흑화소이다. 그러나 0은 이미 백화소를 나타내기 위해 쓰였기 때문에 독립된 흑화소의 표현을 위해서는 9를 쓴다. 그림 1은 입력된 이진 영상의 흑화소에 가중치를 주는 과정을 보여 준다.

아래 그림 1의 (a)에서 (2,2)의 경우 8-이웃 중 흑화소의 개수는 3개이므로 (b)의 (2,2)에

	1	2	3	4	5	6	7		1	2	3	4	5	6	7	
1	0	0	0	0	0	0	0		1	0	0	0	0	0	0	
2	0	1	1	0	0	1	0		2	0	3	3	0	0	9	0
3	0	1	1	0	0	0	0		3	0	5	6	0	0	0	0
4	0	1	1	1	0	0	0		4	0	5	7	6	0	0	0
5	0	1	1	1	1	1	0		5	0	5	8	7	6	3	0
6	0	1	1	1	1	1	0		6	0	3	5	5	5	3	0
7	0	0	0	0	0	0	0		7	0	0	0	0	0	0	0

(a)

(b)

(그림 1) 가중치 결정 예제: (a) 입력 이미지 (b) 가중치를 부여한 이미지

x ₄	x ₃	x ₂
x ₅	p	x ₁
x ₆	x ₇	x ₈

(그림 2) 3×3템플리트에서 화소 p의 8-이웃

가중치 값 3이 된다. (a)의 (5,3)의 경우는 8-이웃이 모두 흑화소이므로 (b)에서 가중치 값이 8이 되는 것을 보여준다. (a)의 (2,6)의 경우는 동떨어진 단일 흑화소로서 8-이웃 중 흑화소를 하나도 갖지 않으므로 (b)의 (2,6)에 가중치 9가 된다.

화소에 기반을 두고 반복적으로 외곽화소를 제거하는 병렬 세선화 알고리즘에는 삭제조건이 필요하다. 어떤 흑화소의 삭제조건은 그 흑화소를 백화소로 바꿀 것인가 아닌가를 결정

한다. 어떤 흑화소가 한번의 삭제 루틴 후 백화소로 바뀌었으면 그것을 삭제되었다고 한다. 화소에 기반을 두고 반복적으로 외곽화소를 제거하는 세선화 알고리즘에서는 검사되는 검은 화소의 주위 화소 중 제한된 수의 화소만을 검사한다(8-이웃이나 24-이웃까지). 반복적 외곽화소 제거방법의 수행은 삭제를 위한 검사 템플리트의 수와 크기에 의존한다. 가중치를 사용함으로서 제안한 알고리즘은 크기에 있어서 3×3 템플리트만을 필요로 하고 검사

해야할 템플리트의 수도 역시 [Jang92]와 비교하여 상대적으로 적다. 또한 병렬 세선화 알고리즘들이 안고 있는 두줄화소 이미지의 처리에도 아무런 문제없이 세선화가 가능하다.

그림 3은 2화소 넓이와 3화소 넓이의 수직 선 패턴을 보여준다. 그림 3(a)의 (2,2)와 그림 3(b)의

(2,2)를 보면 두화소 모두 외곽화소이지만 3×3 이웃만 보아서는 3줄 넓이인지, 2줄 넓이인지 알 수가 없다. 그림 3(a)의 (6,4)와 그림 3(b)의 (6,3)도 마찬가지로 3×3 이웃만 보아서는 구별할 방법이 없다. 그림 3(a)의 이미지는 홀수줄(3줄)이므로 모든 외곽 화소들은 동시에 삭제되어도 된다. 그림 3(a)에 있어서 중앙의 굵은 글씨는 그 이미지에 있어서 중앙 골격선의 후보이다. 그러나 그림 3(b)는 두줄 넓이의 화소이다. 그러므로 단위넓이의 중앙선만을 남기고 세선화하기 위해서는 두줄중 어느 한줄을 삭제해야만 한다. 다른 병렬 알고리즘들은 이러한 두 이미지를 단지 3×3 이웃만 검사하여서는 구별할 방법이 없었다. 이것을 구분하기 위해서는 4×4 이웃을 검사하거나 아니면 삭제루틴을 제외한 복원 루틴을 하나 더 두어서 두줄중 한줄을 복원하는 방법을 사용하였다[Jang92]. 그러나 가중치를 사용할 경우 이러한 어려움은 단지 3×3 이웃만 검사하여서도 쉽게 해결할 수 있다. 그림 4는 가중치를 이용하여 두 이미지를 구분하는 것을 보여

준다. 그럼 4의 (a)에서 (2,2)와 (b)에서 (2,3)의 경우 둘 다 가중치로서 3을 가지고 있다. 하지만 그 화소의 주위화소가 가지는 가중치의 값은 서로 다름을 알 수 있다. 그럼 4(a)의 (2,2)는 가중치 8인 화소를 이웃으로 가지고 있고 (b)의 (2,2)는 이웃 8화소중 가중치 8인 화소가 없다. 마찬가지로 그림 4(a)의 (6,4)와 4(b)의 (6,3)도 서로 다른 가중치를 가진 8-이웃을 가지고 있음을 알 수 있다. 이들이 서로 다른 가중치를 가진 8-이웃을 가졌다고 해서 이들 8-이웃의 가중치를 모두 비교검사해야만 하는 것은 아니다. 단지 8-이웃중 가중치 8을 가진 이웃이 있는가를 검사하면 된다. 어떤 흑화소가 8-이웃 화소중 가중치 8을 이웃으로 한다는 것은 그 흑화소는 삭제가 가능한 주변 화소라는 것을 의미한다.

반복적으로 외곽화소를 제거하는 과정에서 어떤 입력 이미지는 단위 두께(one pixel width)로 남거나 두줄 두께로 남는다. 이 경우 위의 5가지 평가 기준중 2번째 평가기준을 만족시키느냐 혹은 3번째 평가기준을 만족시키느냐의 트레이드오프가 발생한다. 일반적인 이미지 데이터의 경우 정확히 짹수 넓이의 이미지만이 존재하지 않으므로 제안된 알고리즘에서는 이러한 경우 수직선인경우 오른쪽부분을 수평선인경우 위쪽부분을 삭제하는 것을 원칙으로 하고 있다. 다음 절에서 다를 삭제조건에서는 이러한 원칙 하에 삭제조건이 형성

	1	2	3	4	5
1	0	0	0	0	0
2	0	1	1	1	0
3	0	1	1	1	0
4	0	1	1	1	0
5	0	1	1	1	0
6	0	1	1	1	0
7	0	1	1	1	0
8	0	1	1	1	0
9	0	0	0	0	0

(a)

	1	2	3	4
1	0	0	0	0
2	0	1	1	0
3	0	1	1	0
4	0	1	1	0
5	0	1	1	0
6	0	1	1	0
7	0	1	1	0
8	0	1	1	0
9	0	0	0	0

(b)

(그림 3) 두화소 두께와 세화소 두께의 수직 이미지: (a) 세화소 두께의 수직이미지

b) 두화소 두께의 수직 이미지

	1	2	3	4	5
1	0	0	0	0	0
2	0	3	5	3	0
3	0	5	8	5	0
4	0	5	8	5	0
5	0	5	8	5	0
6	0	5	8	5	0
7	0	5	8	5	0
8	0	3	5	3	0
9	0	0	0	0	0

(a)

	1	2	3	4
1	0	0	0	0
2	0	3	3	0
3	0	5	5	0
4	0	5	5	0
5	0	5	5	0
6	0	5	5	0
7	0	5	5	0
8	0	3	3	0
9	0	0	0	0

(b)

(그림 4) (그림 3)의 이미지에 가중치를 부여한 결과

되어갈 것이다.

3. 제안된 세선화 알고리즘

이 절에서는 가중치를 기반으로 하는 병렬 세선화 알고리즘에 대해서 좀더 상세히 다룬 것이다. 먼저 삭제 조건은 각 가중치에 따라 결정된다. 다른 알고리즘들[Chin87, Guo89, Jang92]처럼 모든 삭제조건을 다 검색해야만 되는 것이 아니고 각 가중치에 해당하는 삭제 조건만을 검색하면 된다. 세선화는 어떤 이미지 패턴의 외곽화소를 삭제할 것인가 아닌가를 결정하고 그것을 제거 즉 백화소로 만드는 과정으로 수행된다. 여기서 외곽화소라는 것은 주위 8-이웃 중 적어도 하나의 백화소를 가지는 임의의 흑화소를 말한다. 즉 가중치 1에서 7까지의 값을 가지는 흑화소를 외곽화소라 한다. 가중치 8을 가진 흑화소는 이웃 8화소 모두가 흑화소로 둘러 쌓인 흑화소로 이것은 골격선의 후보가 되며 외곽화소가 아님을 알 수 있다. 삭제가능한 외곽화소의 삭제는 3×3 템플리트의 검색을 통하여 이루어진다. 만일 어떤 이미지의 외곽에 위치한 흑화소의 3×3 템플리트(8-이웃)를 검색하여 그것이 삭제조건을 만족하면 그 흑화소는 삭제되어 백화소로 바뀐다.

제안된 세선화 방법의 개략적인 알고리즘은 다음과 같다.

Step 1. 각 흑화소의 가중치를 결정.

Step 2. 외곽화소들은 각각의 가중치 K를 가지고 있으므로 가중치 K에 해당하는 삭제조건을 검색하여 만일 그 삭제조건을 만족하면 삭제한다(가중치 9를 위한 삭제조건 포함).

Step 3. 각 흑화소의 가중치를 결정.

Step 4. 외곽화소들은 각각의 가중치 K를 가지고 있으므로 가중치 K에 해당하는 삭제조건을 검색하여 만일 그 삭제조건을 만족하면 삭제한다(가중치 9를 위한 삭제조건 제외).

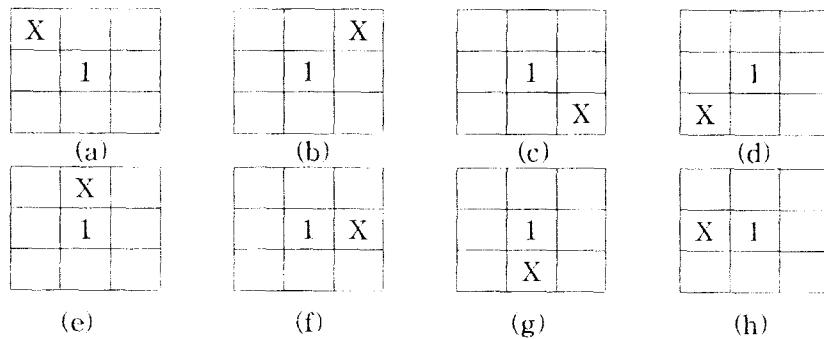
Step 5. 삭제조건을 만족하는 흑화소가 없을 때까지 Step 3과 Step 4를 계속하여 반복한다.

다음은 각 가중치에 따른 삭제조건에 대한 설명이다.

삭제 조건 I (가중치 1을 가지고 있는 외곽화소의 삭제조건): 이웃한 흑화소의 가중치가 3이상이면 삭제한다. 그림 5은 이러한 경우의 삭제를 위한 템플리트의 예이다. 이 삭제조건으로 외곽의 단일 화소 잡음과 그림 5-1의 (6,5) (6,6)과 같이 단일 화소 잡음이 연이어서 두개까지 있는 외곽화소도 제거가 가능하다. 가중치 1을 가지고 있는 화소는 그림 5-1의 화소((1,5) (6,6))들과 같이 이웃 8화소중

단 하나의 흑화소만을 이웃으로 하고 있는 경우이다. 이 조건은 외곽의 단일화소 잡음이나 기타 잡음으로부터 토폴로지를 유지하기 위한 삭제조건으로 그림 5-1에 간단한 예를 보여주고 있다. 이 조건에 의해 단일 화소 잡음을 제거할 수 있다(그림 5-1의 (1,5)). 만일 그림

므로 큰 오버헤드를 감수해야만 한다. 제안된 알고리즘은 단지 3×3 템플리트 4개((e)~(h))만 추가함으로서 이러한 잡음을 효율적으로 처리할 수 있다. 그림 5-2는 알고리즘 [Jang92]와 제안된 알고리즘으로 그림 5-1을 세선화한 결과이다. 그림 5-1의 (6,6)을 처리



(그림 5) 1을 가중치로 가지고 있는 외곽화소의 제거를 위한 템플리트.

(X는 가중치 1에서 8의 값중 하나를 가지는 흑화소이고 공백은 백화소를 의미한다)

6-1의 직사각형 부분만이 유용한 데이터 부분이라 할 때 그 외의 화소들((3,5), (6,5), (6,6) 과 (1,5))은 잡음으로 처리되어야 한다. 기존의 세선화 알고리즘 중 단일 화소의 잡음을 처리하는 알고리즘[Jang92]이 있기는 하나 대부분의 세선화 알고리즘은 외곽의 잡음을 효율적으로 처리하지 못하고 있다[Chin87, Guo89, Hall89, Holt87]. 알고리즘[Jang92]에서도 잡음((6,5), (6,6))을 처리하기 위해서는 3×4 또는 3×5 의 템플리트를 검색해야 하

느냐 못하느냐에 따라 골격선 추출에 있어서 전혀 상이한 결과를 초래할 수가 있음을 알 수 있다.

삭제 조건 Ⅱ (가중치 2를 가지는 흑화소의 삭제조건) : 가중치 2를 가지는 흑화소는 그것의 8-이웃이 가중치에 관계없이 그림 6의 어느 하나의 템플리트와 일치하면 삭제한다.

삭제 조건 Ⅲ (가중치 3을 가지는 흑화소의

	1	2	3	4	5	6	7	
1	0	0	0	0	0	0	0	
2	0	X	X	X	0	0	0	
3	0	X	X	X	0	0	0	
4	0	X	X	X	0	0	0	
5	0	X	X	X	0	0	0	
6	0	X	X	X	4	0	0	
7	0	X	X	X	0	0	0	
8	0	X	X	X	0	0	0	
9	0	X	X	X	0	0	0	
10	0	X	X	X	0	0	0	
11	0	0	0	0	0	0	0	

(그림 5-1) 외곽면 잡음의 예

삭제조건) : 가중치 3을 가진 흑화소는 다음
둘중 하나의 조건을 만족하면 삭제한다. Ⅲ.
i) 그 화소의 8-이웃이 가중치에 관계없이 그
림 7의 (a)-(i)중 하나의 템플리트와 일치하면

삭제한다. Ⅲ. ii) 8-이웃중 가중치 7이상의 이
웃을 적어도 하나 가지고 있고 그림 8의 템
플리트 (j)-(l)중 하나와 일치하면 삭제한다. 템
플리트 (a)-(d)는 외곽의 단일 화소 잡음을

0	0	0	0	.	0	0
0	.	.	.	0	0	0
0	.	.	X	.	0	0
0	.	X	.	0	0	0
0	.	X	.	0	0	0
0	.	X	X	X	0	
0	.	X	.	0	0	0
0	.	X	.	0	0	0
0	.	X	.	0	0	0
0	.	.	.	0	0	0
0	0	0	0	0	0	0

(a)

0	0	0	0	.	0	0
0	.	.	.	0	0	0
0	.	.	X	.	.	0
0	.	X	.	0	0	0
0	.	X	.	0	0	0
0	.	X	.	0	0	0
0	.	X	.	0	0	0
0	.	X	.	0	0	0
0	.	.	.	0	0	0
0	0	0	0	0	0	0

(b)

(그림 5-2) 외곽면의 잡음 처리예:

(a) 알고리즘(Jang92)으로 세선화한 결과 (b) 제안된 알고리즘으로 세선화한 결과

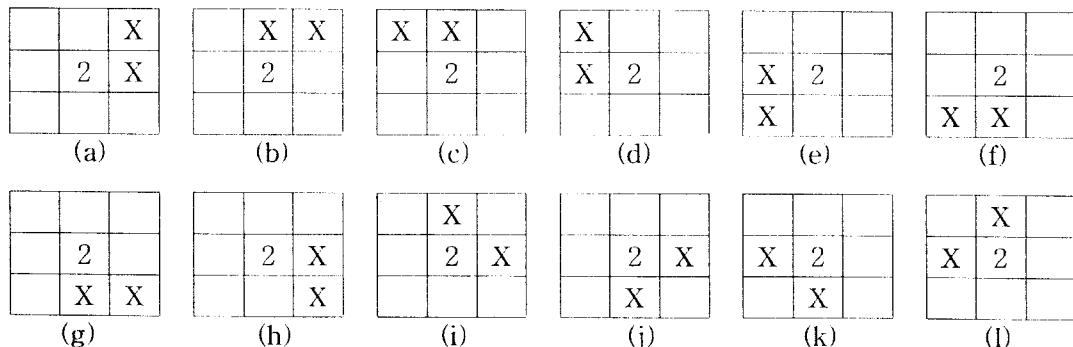
제거하기 위해 부여된 조건이다.

예를 들어 그림 6-2의 화소(3,5)는 단일 화소 잡음이므로 그림 8의 템플리트 (b)를 이용하여 삭제된다. 템플리트 (e)-(h)는 대각선으로 된 화소들을 처리하기 위한 것들이다. 템플리트 (i), (j), (k) 와 (l)은 각각 대칭임을 알 수 있다. 템플리트 (i)의 중앙 화소는 어떤 제약조건 없이 삭제될 수 있다. 이것은 템플리트 (i)가 앞서 기술한 삭제전략을 만족하기 때문이다. 그러나 템플리트 (j), (k) 나 (l)은 8-이웃 중 가중치 7 이상의 값을 가져야 한다는 제약이 필요하다. 이것은 두화소 넓이의 패턴만이 남았을 때의 삭제규칙으로 수평패턴에서는 위쪽, 수직패턴에서는 오른쪽 선을 삭제하는 것을 규칙으로 했기 때문이다. 템플리트 (i)를 적용하여 두줄화소 두께의 수평, 수직 패턴에서 위쪽, 오른쪽 화소가 제거된다. 그림 8-1(a)에

서 화소 ①는 두줄 두께의 수평패턴에서 위쪽 부분의 화소이고, 그림 8-1(d)에서의 화소 ①는 두줄 두께의 수직패턴에서 오른쪽 화소이다. 그러나 화소

①, ⑩, ⑪ 은 위의 두 규칙을 동시에 만족시키지 못한다. 예를 들어 그림 8-1(b)의 화소 ①는 수평패턴에서는 조건을 만족하지만 (c) 와 같이 수직 패턴에서는 조건을 만족하지 못하고 왼쪽부분에 위치한 것을 알 수 있다. 이러한 제약은 화소 ①, ⑩, ⑪가 동시에 삭제되는 것을 방지할 수 있고 어떤 패턴의 세선화 후 길이정보를 최대한 보존할 수 있다. 또한 이러한 이유로 2×2 사각형 모양의 패턴이 완전히 없어지는 것을 방지할 수 있다.

삭제조건 IV (가중치 4를 가지는 흑화소의 삭제조건) : 가중치 4를 가진 흑화소는 이웃한 흑화소의 가중치와 무관하게 그림 8의 템플리



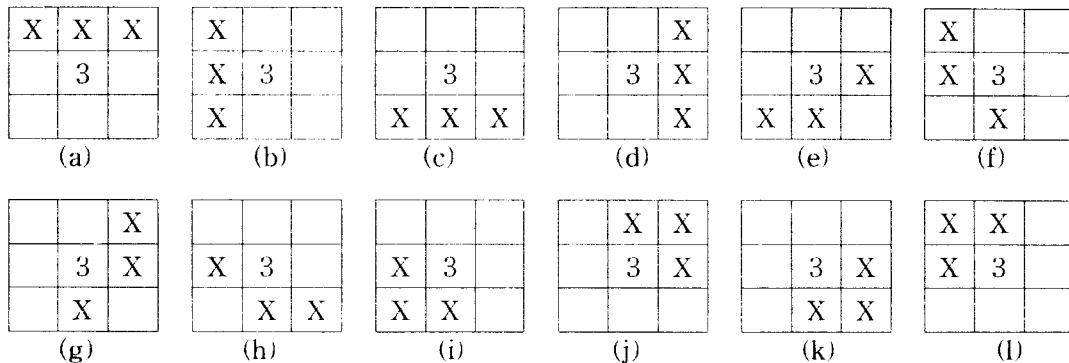
(그림 6)가중치 2를 가진 외곽화소 제거를 위한 템플리트

가중치를 이용한 병렬 세선화 알고리즘 11

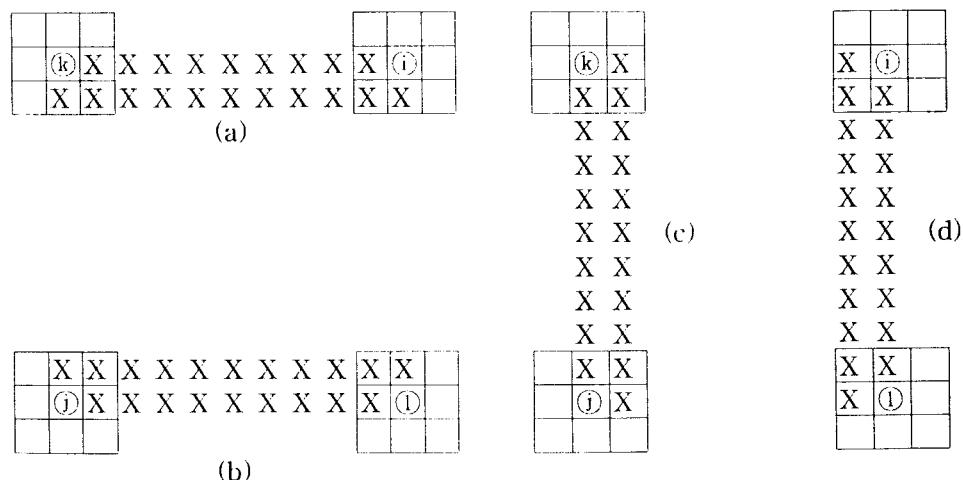
트중 어느 하나와 일치하면 삭제된다. 템플리트 (a)-(h)는 보통 경계화소를 제거하기 위한 조건이고 (i)-(j)는 두줄화소로된 대각선 모양의 화소를 제거하기 위한 조건이다. 이러한 삭제전략에 의하여 대각선 화소의 위쪽화소를

삭제할 수 있다. 그럼 8-2의 ① 와 ②는 이러한 적용의 예이다.

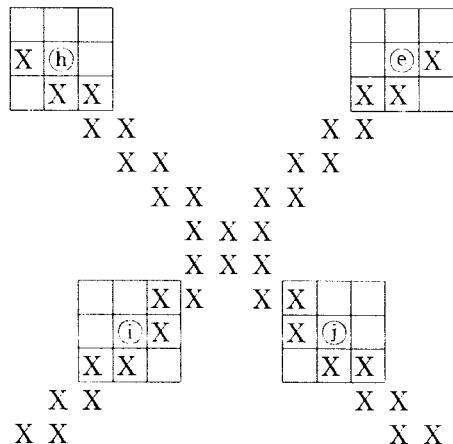
삭제 조건 V(가중치 5를 가지는 혹화소의 삭제조건) : 삭제조건 V.i) 가중치 5를 가진



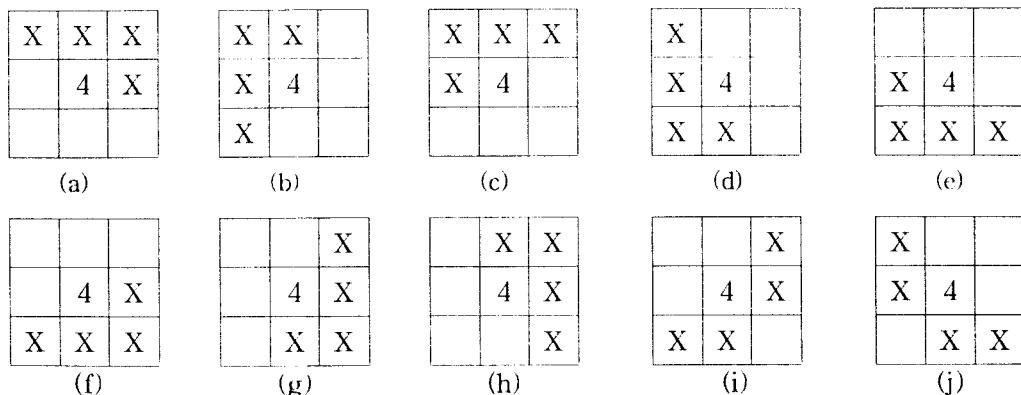
(그림 7) 가중치 3을 가진 외곽화소를 위한 템플리트



(그림 7-1) 가중치 3을 가진 템플리트의 적용예



(그림 7-2) 가중치 3과 4를 가진 템플리트의 적용 예



(그림 8) 가중치 4를 가진 외곽화소 제거를 위한 템플리트

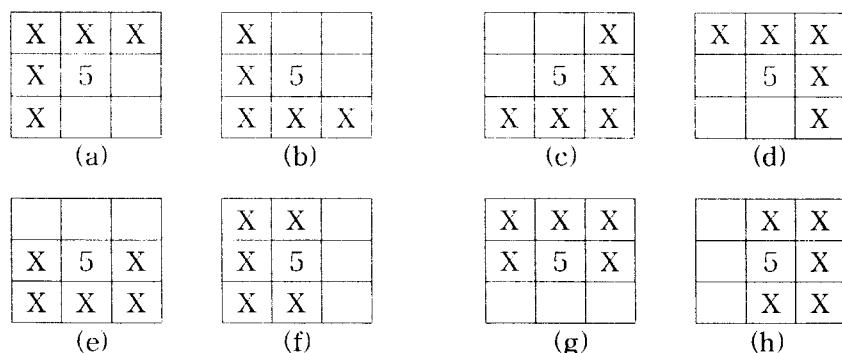
흑화소중 그림 9의 (a)-(f) 중 하나의 템플리트와 일치하면 삭제한다. 삭제조건 V. ii) 템플리트 (g)나 (h)와 일치하고 8-이웃 중 적어도 하나는 가중치 8을 가질 때 삭제한다. 그림 9

를 보면 (e), (f)와 (g), (h)는 서로 대칭인 것을 알 수 있다. 삭제조건 V. ii)는 대칭성을 가지는 이 템플리트들을 서로 다른 조건으로 삭제함으로써 앞의 삭제전략을 만족시키고 두

화소 두께의 패턴만이 남았을 때 연결성을 유지시키기 위한 삭제조건을 만든 것이다. 그럼 4(b)를 보면 화소 (4,2)와 (4,3)은 동시에 삭제되어서는 안된다. (4,2)나 (4,3)중 하나는 흑화소로서 남아있어야 두줄 화소로된 수직 패턴의 연결성을 유지할 수 있다. 조건 V. ii) 는 화소 (4,2)가 조건을 만족하지 못하게 함으로서 (4,2)와 (4,3)이 동시에 제거되는 것을 방지한다.

삭제조건 VII(가중치 6을 가지는 흑화소의 삭제조건) : 조건 VII. i) 만일 그 흑화소의 8-

중치 6을 가지는 흑화소들을 위한 템플리트들은 모두가 서로 대칭성을 가지고 있다. 그럼 10에 나열된 템플리트들 외에 나올 수 있는 경우는 연결성 유지를 위해서 삭제되어서는 안 되는 것들이다. 그럼 10-1은 가중치 6을 가지는 흑화소가 나타날 수 있는 경우 중 연결성 유지를 위해서 대칭 되는 화소중 하나씩만 삭제되어야하는 경우들이다. a와 e, b와 f, c와 g, d와 h는 서로 대칭이다. 이러한 경우 둘중 하나만을 선별하여 삭제하여야 한다. 앞의 두줄 화소 삭제 전략에 따라 이러한 경우 위쪽, 오



(그림 9) 가중치 5를 가진 외곽화소 제거를 위한 템플리트

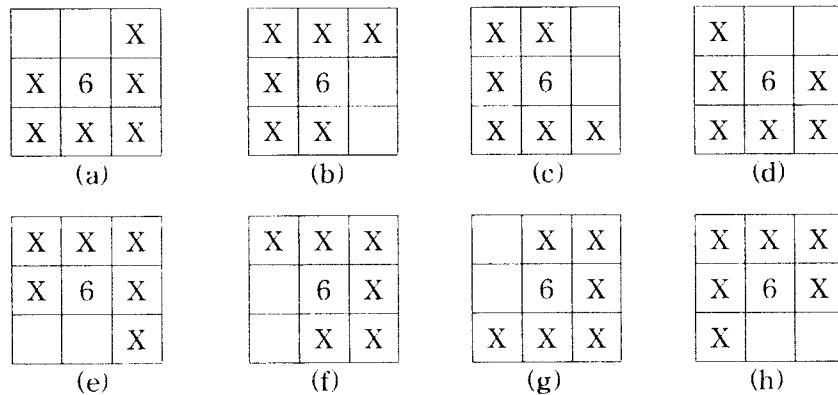
이웃이 그림 10의 (a)-(d)중의 어느 한 템플리트와 일치하면 이웃한 화소의 가중치에 무관하게 삭제한다. VII. ii) 템플리트 (e)-(h)중의 어느 한 템플리트와 일치하고 8-이웃 중 가중치 8인 화소를 하나 이상 가질 때 삭제한다. 가

른쪽을 삭제하기 위하여 조건을 VII. i)와 VII. ii)로 나누었다. 그럼 10-1의 화소 ④는 조건 VII. i)을 만족하여 그림 10의 템플리트 (a)에 의하여 삭제될 수 있으나 그와 대칭인 화소 ⑥는 조건 VII. ii)을 만족하지 못하므로 삭제되지

않는다.

삭제 조건 VII (가중치 7을 가지는 흑화소의 삭제조건) : 조건 VII.i) 가중치 7을 가진 흑화

진 이웃을 5개 가지고 있을 때 삭제한다(예 그림 11-2). 가중치 8인 이웃을 8개 가질 수 있는 경우는 그림 11-1의 템플리트와 같을 때뿐



(그림 10) 가중치 6을 가진 외곽화소 제거를 위한 템플리트

		X	X			
		g	(c)			
X	(a)	X	X	(d)	X	
X	e	X	X	h	X	
		f	(b)			
		X	X			

(그림 10-1) 가중치 6을 가진 화소의 대칭예

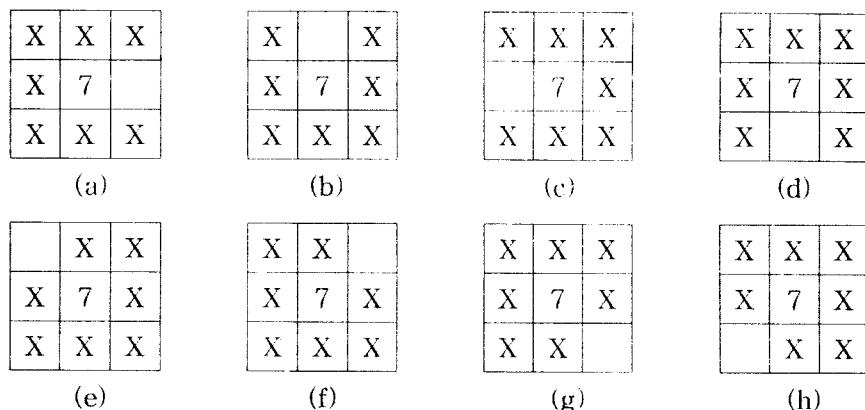
소가 그림 11의 템플리트 (a)나 (b)와 일치하면 이웃화소의 가중치와 관계없이 삭제한다. 조건 VII.ii) 템플리트 (c)나 (d)와 일치하며 8-이웃중 가중치 8을 가진 화소를 2개이상 가지고 있을 때 삭제한다. VII.iii) 가중치 8을 가

이다. 조건 VII.i)의 템플리트와 조건 VII.ii)의 템플리트는 서로 대칭이다. 예를 들어 그림 11-1의 ④는 ⑤와 동시에 삭제되어서는 안되고 ⑥는 ⑦와 동시에 삭제되어서는 안된다. 앞의 삭제전략에 따라 ④와 ⑥는 그림 11의 템플

리트 (a)나 (b)와 일치하면 삭제를 하고 ④와 ⑤는 조건 VII. ii)에 따라 템플리트 (c)나 (d)와 일치하고 8-이웃중 가중치 8인 화소를 2개 이상 가져야 한다. 이렇게 함으로써 그림 11-1과

같은 패턴이 있을 때 연결성을 유지하면서 세선화를 할 수 있다.

삭제 조건 VIII(가중치 9를 가진 흑화소의 삭제조건) : 가중치 8을 가진 흑화소는 다른 흑



(그림 11) 가중치 7을 가진 외곽화소 제거를 위한 템플리트

X	X	X	X
X	X	X	X
(c)	(a)		
X	X	X	X
X	X	X	X

X	X		X	X
X	X	(b)	X	X
X	X	(d)	X	X
X	X		X	X

(그림 11-1) 가중치 7을 가진 화소의 대칭예

		X	X	X			
		X	X	X			
X	X	(e)	X	(f)	X	X	
X	X	X	X	X	X	X	
X	X	(g)	X	(h)	X	X	
		X	X	X			
		X	X	X			

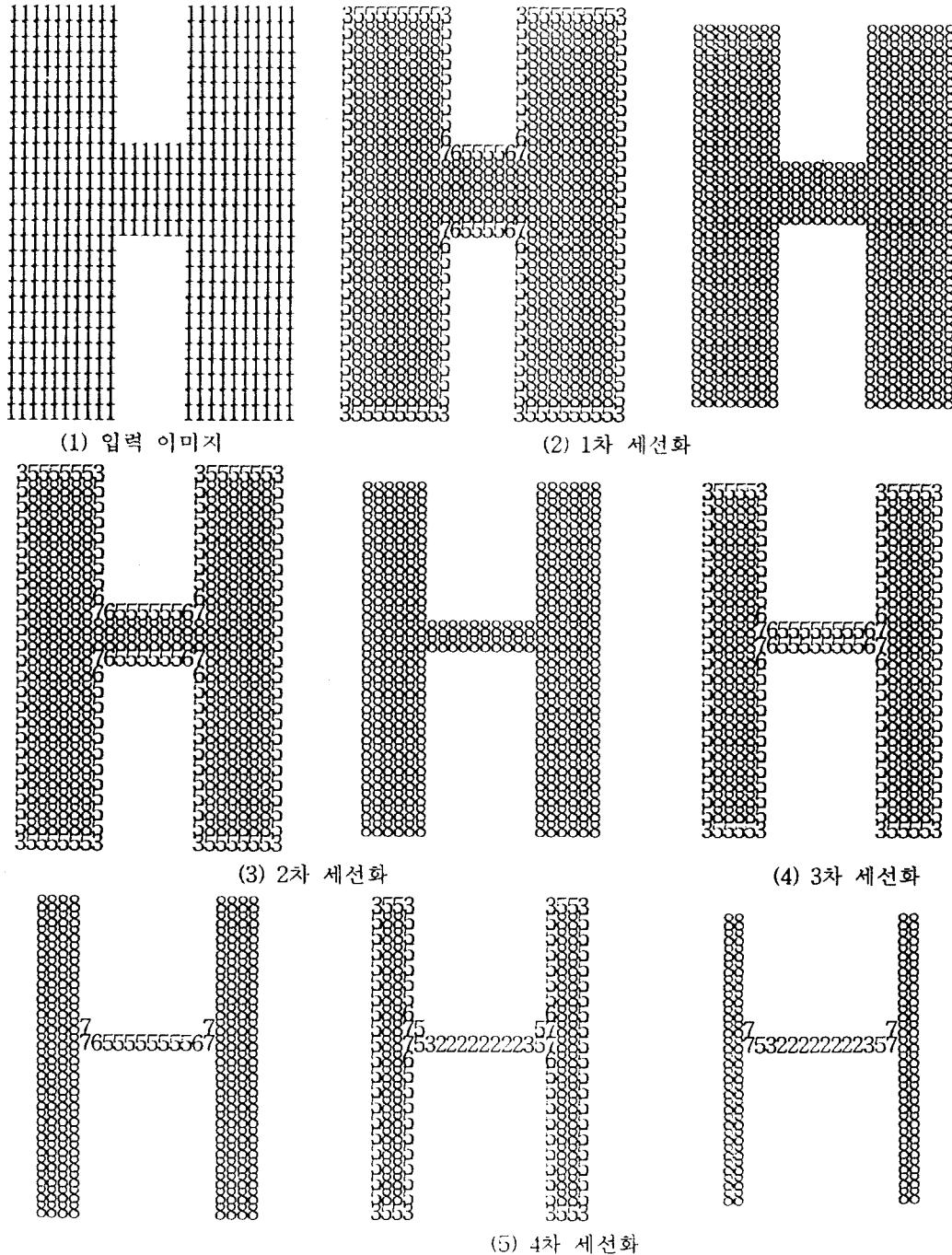
(그림 11-2) 가중치 7을 가진 화소중 이웃에 가중치 8을 5개 이상 가진 화소들의 예

화소로 완전히 둘러싸인 화소이므로 골격선의 후보로 삭제대상에서 제외된다. 반면에 가중치 9를 가진 흑화소는 다른 흑화소로부터 완전히 독립된 흑화소로 이웃 8화소중 흑화소가 하나도 없는 경우에 해당한다. 이웃하는 흑화소가 하나도 없는 경우 실제의 가중치는 0이 되나 0은 백화소를 표기하기 위해서 이미 쓰여졌으므로 0대신 9로 표기하는 것이다. 세선화를 하기 전에 다른 흑화소로부터 완전히 독립된 단일 흑화소는 유용한 데이터라고 할 수 없다. 제안된 알고리즘에서는 이러한 단일 흑화소들은 세선화 과정중 제거한다. 하지만 그림 18과 같은 경우 여러 단계의 세선화를 거쳐 단일 화소로 남을 경우도 있으므로 첫 번째 세선화의 경우만 가중치 9를 가진 화소를 제거하고 다음

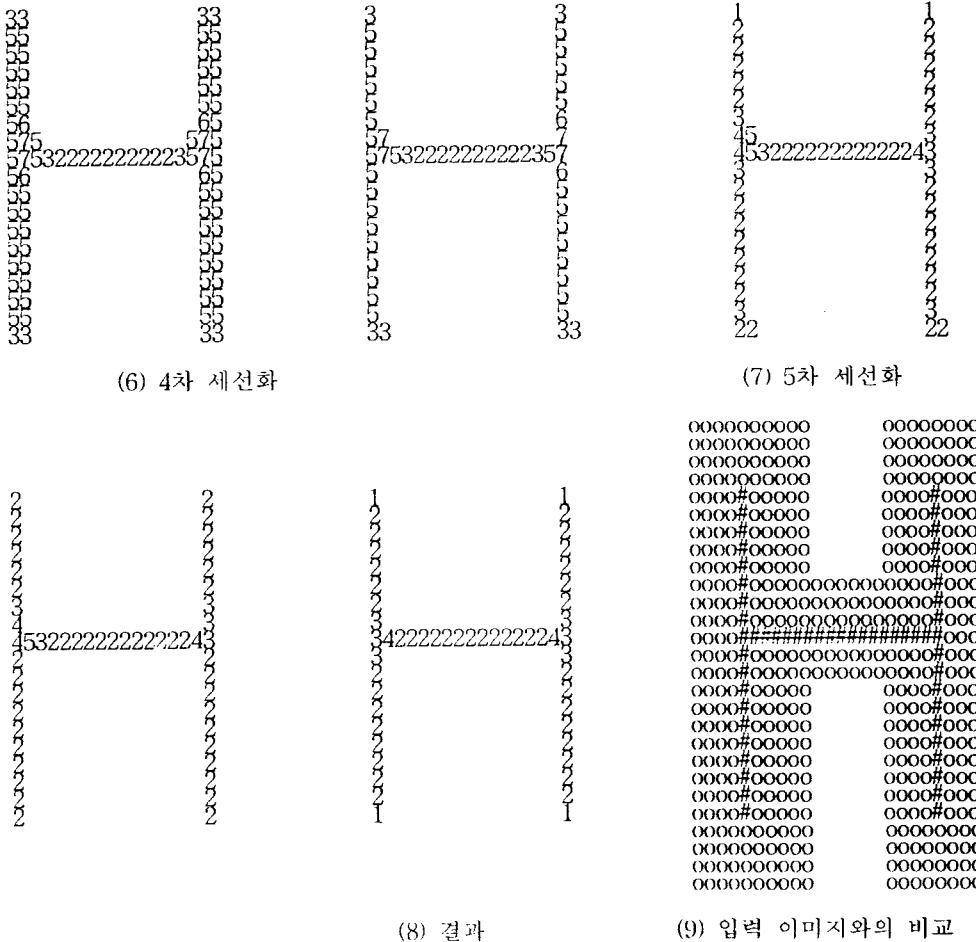
단계부터는 제거하지 않는다. 다음 그림12는 10화소 두께의 H자 이미지를 제안된 알고리즘으로 세선화하는 과정이다.

제안된 알고리즘을 이용한 세선화의 예

가중치를 이용한 병렬 세선화 알고리즘 17



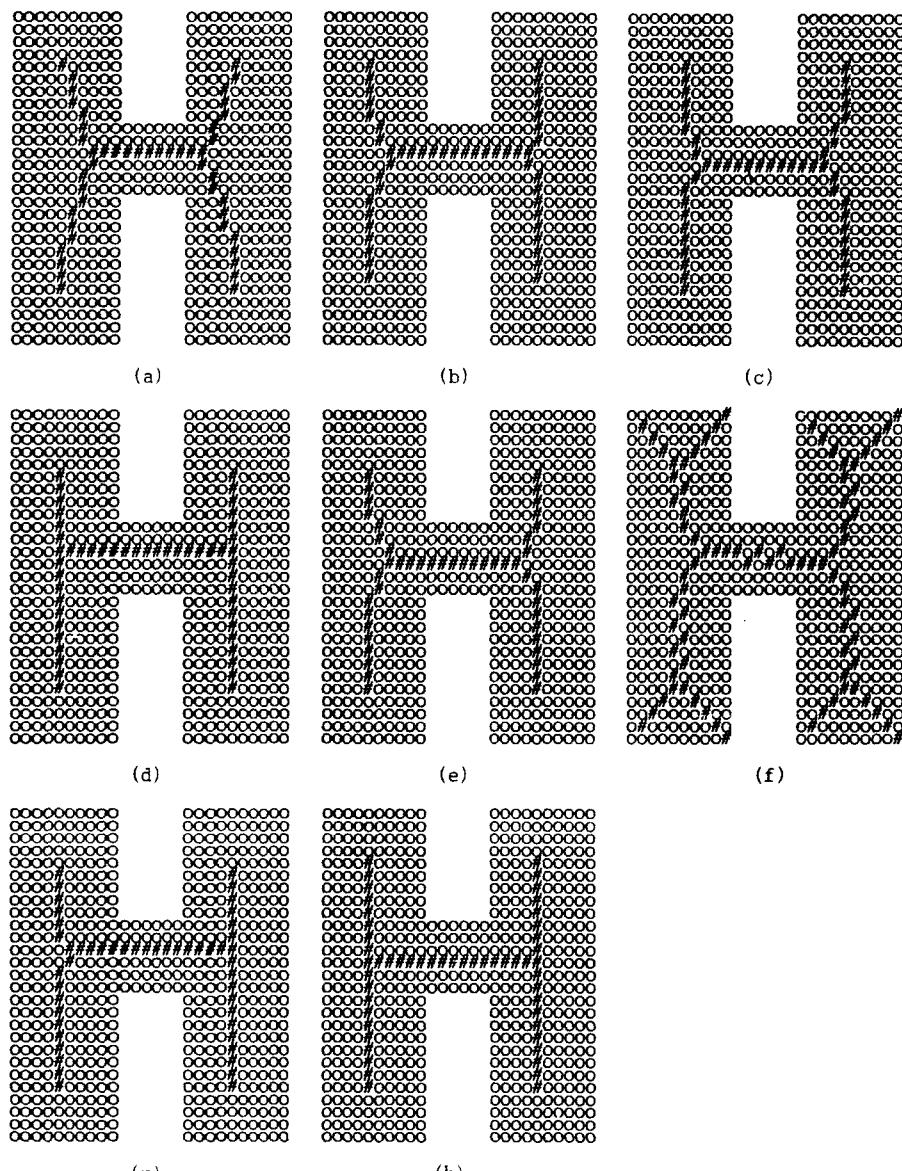
(그림 12) 10학소 두께의 H자 이미지의 세선화 단계



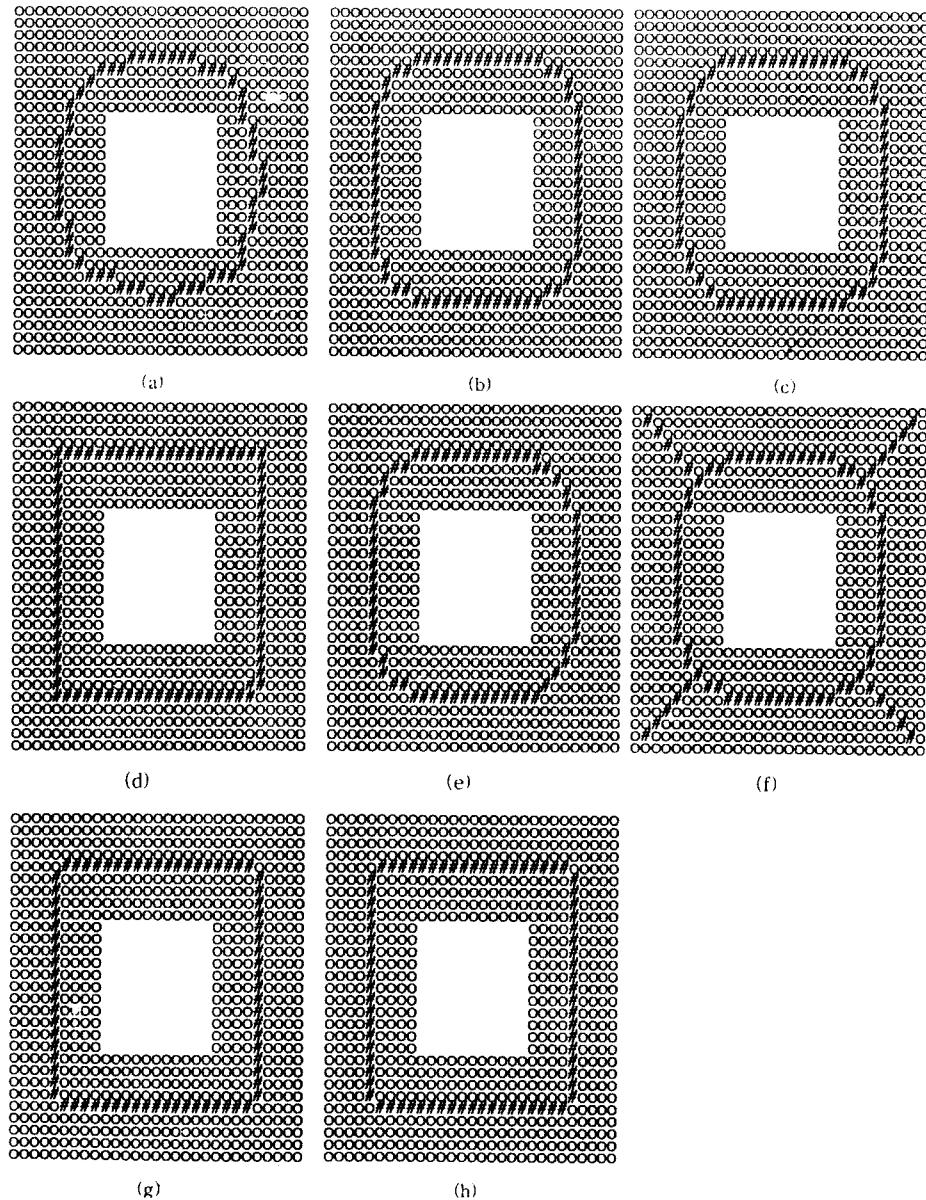
4. 성능 평가

성능 평가는 여러 개의 알고리즘들과 하되 그것들이 세선화하기 어려운 데이터를 중심으로 할 것이다. 그림 17에서 대부분의 알고리즘이 외곽선의 단일 화소 잡음으로 인하여

전체적인 모양이 왜곡되어 골격선에서 바람직하지 못한 결과를 나타내고 있다. 제안된 알고리즘은 이러한 문제점을 아무런 어려움 없이 처리하며 연결점을 제외하고는 모든 부분에서 단일 화소 넓이로 세선화가 된다. 그림 14부터 그림 17은 타 알고리즘과의 비교평가이다.

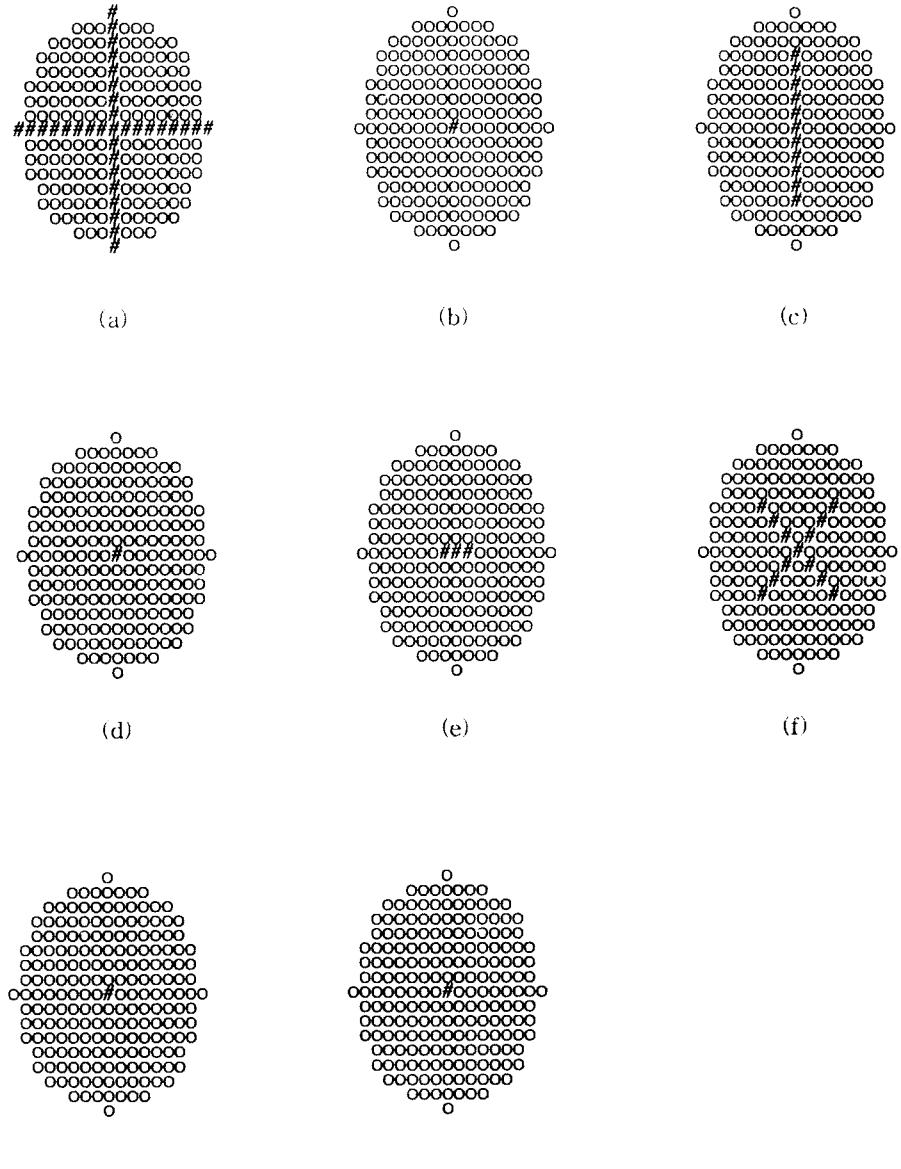


(그림 13) (a) Chin et al's 알고리즘	(b) Holt et al's 알고리
(c) Naccache and Shinghal's 알고리즘	(d) Lu and Wang's 알고리즘
(e) Guo and Hall's 알고리즘(A1)	(f) Guo and Hall's 알고리즘(A2)
(g) Jang and Chin's 알고리즘	(h) 제안된 알고리즘

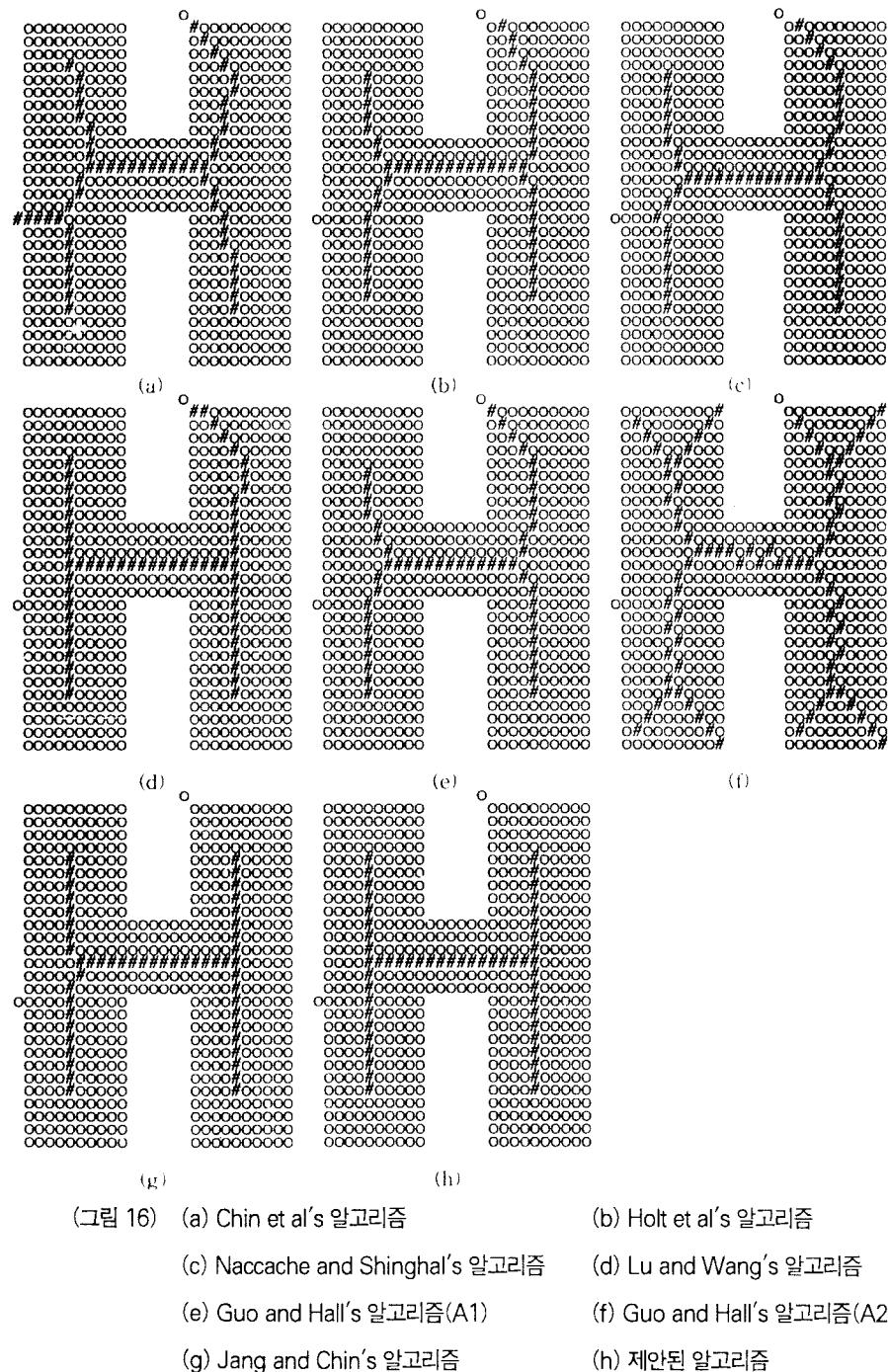


(그림 14) (a) Chin et al's 알고리즘
 (b) Holt et al's 알고리즘
 (c) Naccache and Shinghal's 알고리즘
 (d) Lu and Wang's 알고리즘
 (e) Guo and Hall's 알고리즘(A1)
 (f) Guo and Hall's 알고리즘(A2)
 (g) Jang and Chin's 알고리즘
 (h) 제안된 알고리즘

(b) Holt et al's 알고리즘
 (d) Lu and Wang's 알고리즘
 (f) Guo and Hall's 알고리즘(A2)
 (h) 제안된 알고리즘



(그림 15) (a) Chin et al's 알고리즘 (b) Holt et al's 알고리즘
 (c) Naccache and Shinghal's 알고리즘 (d) Lu and Wang's 알고리즘
 (e) Guo and Hall's 알고리즘(A1) (f) Guo and Hall's 알고리즘(A2)
 (g) Jang and Chin's 알고리즘 (h) 제안된 알고리즘.



7개의 타 알고리즘과 비교 결과 H자 이미지를 제외한 부분에서 알고리즘 [Jang92]와 제안된 알고리즘이 가장 우수함을 알 수 있다. 알고리즘 [Jang92]는 단일 반복 병렬 세선화 알고리즘이라는 점에서 제안된 알고리즘과 공통점이 있다. 제안된 알고리즘은 이러한 알고리즘 [Jang92]과 몇 가지 면에서 추가적인 비교평가를 할 것이다.

다음 표 1은 스캐너로 입력된 필기체 숫자中最 두꺼운 부분의 평균이 10인 750개의 데이터를 가지고 실험한 결과이다

사용하므로 검색하는 이웃화소의 수가 최대 24이고 제안된 알고리즘은 3×3 템플리트만 사용하므로 최대 8-이웃만을 검사한다. 검색해야하는 최대 템플리트의 수는 어떤 흑화소를 백화소로 바꾸기 위하여 비교해야하는 템플리트의 수자로 알고리즘 [Jang92]의 경우 삭제를 위한 비교 후 복원을 위한 비교를 다시 해야만 하므로 최악의 경우 30이 되고, 제안된 알고리즘의 경우 각 가중치에 따라 해당 템플리트만 비교하므로 최대가 12이다. 잡음처리 능력은 적은 수 외곽 화소로 인하여 전체 골격

(표 1) 알고리즘 [Jang92]와 제안된 알고리즘의 성능 비교

비교 기준	제안된 알고리즘	알고리즘 [Jang92]
검색하는 최대이웃 화소의 수	8	24
검색해야하는 최대 템플리트	12	30
잡음 처리능력	강	약
검색해야하는 평균 화소의 수	130.68	968.40
연결성	완전 8연결	완전 8연결

검색하는 최대이웃 화소의 수는 어떤 하나의 외곽 흑화소를 백화소로 바꾸기 위해서 검색해야하는 이웃 화소의 수이다. 알고리즘 [Jang92]의 경우 비교 템플리트로 5×5 까지

선의 모양이 왜곡되어 추출되는 것으로 그림 20에 몇 개의 필기체 숫자의 세선화결과로 비교하였다. 검색해야하는 평균 화소의 수는 하나의 이미지를 세선화 하여 최종적으로 골격

24 한낙희, 이필규

선을 추출하기 위하여 검색해야만 하는 화소의 수를 평균한 것이다(예. $(8 / 2) \times (12 / 2) \times 5,445 = 130.68$). 여기서 5,445는 750개의 데이터를 세선화 할 때 평균 반복 횟수이다. 마지막으로 연결성은 다음과 같은 정의에 따른 것이다(Rose66).

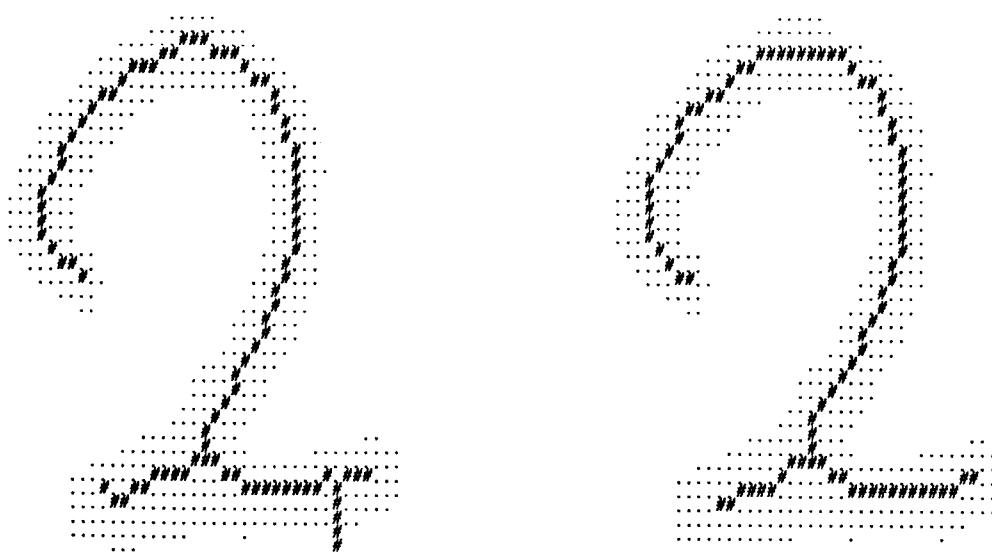
4 연결성 : 모든 화소가 대각선 방향으로는

연결되지 않고 상하 좌우로만 연결된 형태

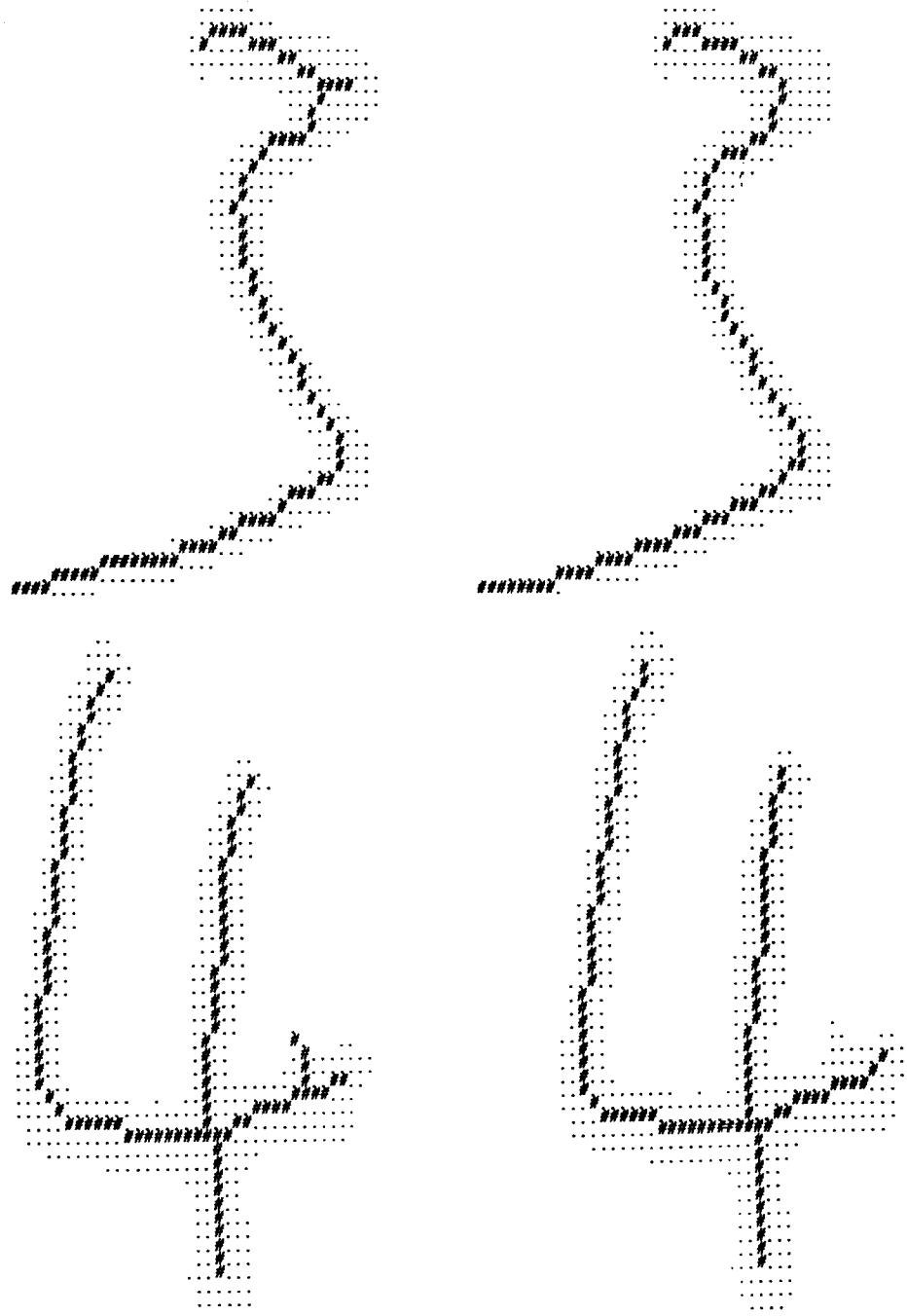
완전-8 연결성 : T 교차점을 제외하고는 모든 화소가 대각선 방향 또는 상하 좌우로 연결된 형태

불완전-8 연결성 : 4연결성과 완전-8 연결성이 공존하는 형태

다음 그림 17은 필기체 숫자들로서 잡음처

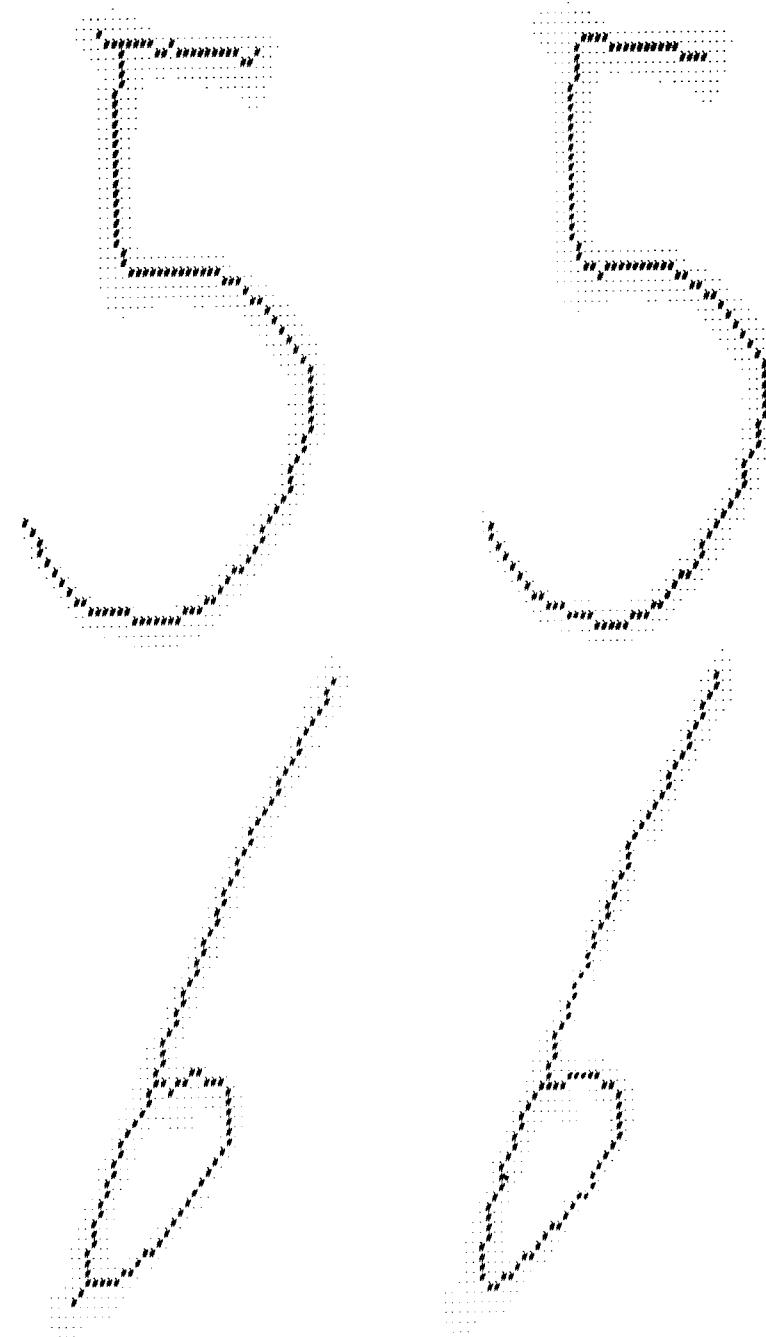


(그림 17) 세선화시 잡음처리 능력 비교

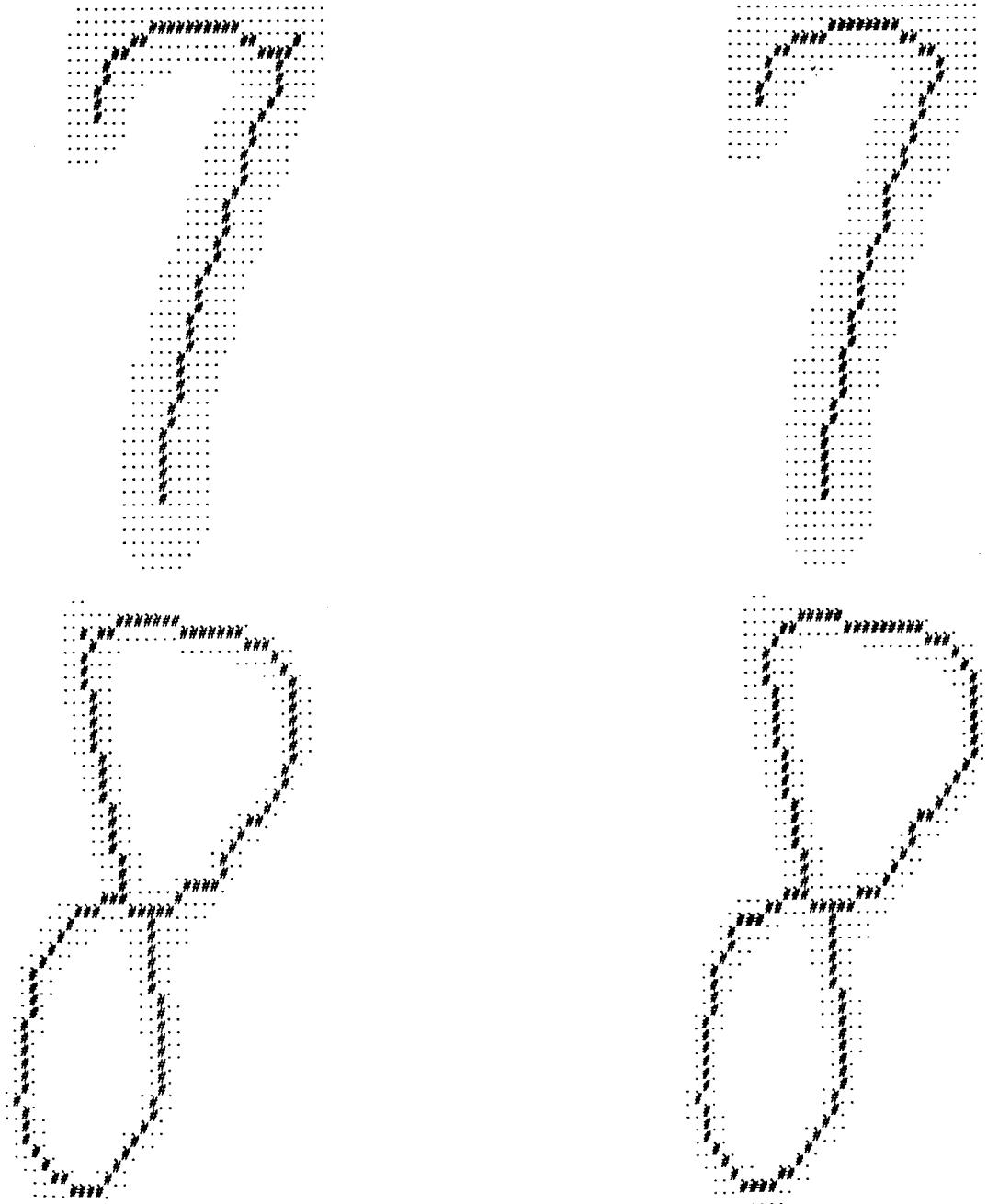


(그림 17) 계속

26 한나희, 이필규

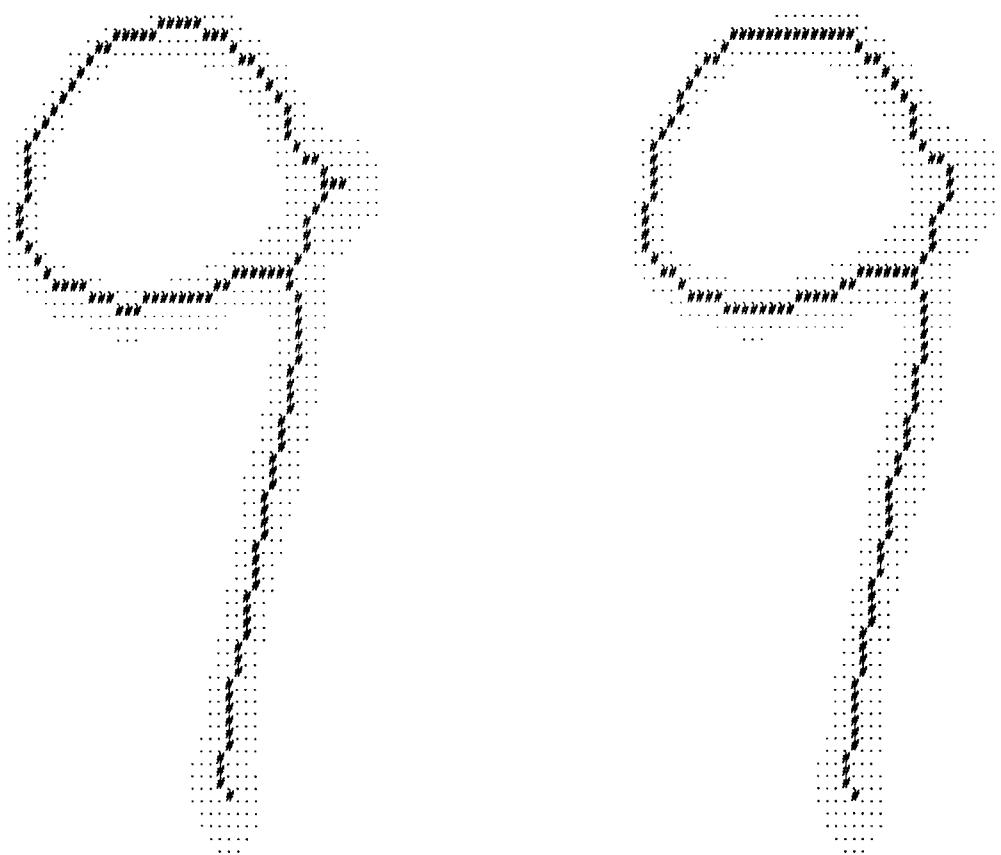


(그림 17) 계속



(그림 17) 계속

28 한나희, 이필규



(그림 17) 계속

리에 있어서 알고리즘 [Jang92]와 제안된 알고리즘의 비교이다. 왼쪽에 있는 그림이 알고리즘 [Jang92]로 세선화한 것이고 오른쪽에 있는 그림은 제안된 알고리즘으로 세선화한 결과이다. 알고리즘 [Jang92]의 경우 많은 경우에 하나나 두화소의 잡음으로 인하여 불필요한 잔가지가 형성된 것을 볼 수 있다.

5. 결론

본 논문은 삭제를 위해서 검색하는 화소의 수를 최소화하는 단일 반복 병렬 세선화 알고리즘을 개발하기 위한 것이다. 가중치는 다른 세선화 알고리즘과 구분되는 특징이다. 이 방법으로 본 논문은 타 병렬 알고리즘이 개선하지 못한 삭제를 위한 이웃화소의 검색범위를 현저히 줄일 수 있었다. 또한 잡음이 많은 필기체 문자나 숫자 인식에 전 처리를 최소화하고도 이상적인 골격선 추출이 가능하게 하였다. 성능 평가는 타 세선화 알고리즘들이 처리하기 어려웠던 부분들을 중심으로 하였으며 일반화된 실험은 스캐너로 입력된 필기체 숫자 데이터 750개를 가지고 행하였다. 실험 결과 제안된 알고리즘은 세선화 알고리즘이 갖추어야 할 평가기준들을 만족하며 일반화된 데이터의 처리에 적당한 알고리즘으로 판명되었다. 특히 제안된 알고리즘은 외곽면의 잡음에 강한 면을 보임으로써 필기체 문자인식이나

숫자인식의 전처리기로서 성능향상에 크게 기여하리라 여겨진다.

참 고 문 헌

- [Arce89] C. Arcelli and G. S. D. Baja, "A width-independent fast thinning algorithm." *IEEE Trans. Patt. Anal. Machine Intell*, vol. PAMI-11, no. 4, pp.411-414, Apr. 1989.
- [Blu64] B. Blum. "A transformation for extracting new descriptors of shape," *Proc. Symp. Modules Perception Speech Visual Form*(Cambridge, MA), 1964.
- [Borg84] G. Borgeforw. "Distance transformations in arbitrary dimensions," *Comput. vision Graphics Image Processing*, vol. 27, pp.321-345, 1984.
- [Chen88] Y. S Chen and W.H. Hsu, "A modified fast parallel algorithm for thinning digital patterns," *Patt. Recog. Lett.* vol. 7, pp.99-106, 1988.
- [Chin87] R. T. Chin, H. K. Wan, D. L. Stover, and R. D. Iverson, "A one-pass thinning algorithm and its

- parallel implementation," *Comput. Vision Graphics Image Processing.* vol. 40, no. 1, pp.30-40, Oct. 1987.
- [Dani80] P. E. Danielsson, "Euclidean distance mapping." *Comp. Graphics Image Processing*, vol. 14, pp.227-248, 1980.
- [DeCe89] A.L. DeCegame, *Parallel Processing Architectures and CLSI Hardware*. Englewood Cliffs. NJ: Prentice-Hall, 1989.
- [Guo89] Z. Guo and R. Q. Hall, "Parallel thinning with two-subiteration algorithms," *Commun. Assoc. Comput. Mach.* vol. 32, no. 3, pp.359-373, Mar 1989.
- [Hall89] R. W. Hall, "Fast parallel thinning algorithms: Parallel speed and connectivity preservation," *Commun. Asoc. Comput. Mach.* vol. 32, no. 1, pp.124-131, Jan. 1989.
- [Holt87] C. M. holt, A. Stewart, M. Clint, and R. H. Pettott, "An improved parallel thinning algorithm," *Comm. of the ACM*, vol. 30, no. 2, 1987, pp.156-160.
- [Hung83] S.H.Y. hung and T.kasvand, "Critical points on a perfectly 8- or 6-connected thin binary line," *Pattern Recogn.*, vol. 16, no. 3, pp.297-306, 1983.
- [Jang92] Ben k. Jang and Roland T. Chin, "One-Pass Parallel Thinning: Analysis, Properties, and Quantitative Evaluation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-14, no. 11, pp.1129-1149, Nov. 1992.
- [Loui92] Louisa Lam and Seong-Whan Lee, "Thinning Methodologies-A Comprehensive Survey," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-14, no. 9, pp.869-885, Sep. 1992.
- [Rose68] A. Rosendfeld and F. Pfaltz, "Distance functions on digital pictures," *Recogn.*, vol. 1, pp.33-61, 1968.
- [Rock82] ___, *Algorithms for Graphics and Image Processing*. Rockville, MD, Comput. Sci., pp.195-214, 1982.
- [Suen86] Y.K. Chu and C.Y. Suen, "An Alternative Smoothing and Stripping Algorithm for Thinning Digital Binary Patterns," *Signal Processing*, vol. 11, no. 3, pp.207-222, 1986.

[Tori81] J. I. Toriwaki and S. Yokoi,
“Distance transformation and
skeleton digitalized pictures with
applications.” *Progress in Pattern
Recognition*, (L. N. Kandel and A.
Rosenfeld, Eds) Amsterdam. The
Netherlands North-Holland, pp.187-
265, 1981.

[Wang86] H. E. Lu and P. S. Wang, “A
comment on a fast parallel algorithm
using thinning digital pattern.”
Commun. ACM, vol. 29, no. 3,
pp.239-243, Mar. 1986.