

최대 블록화 방법을 이용한 문자획 특징 추출에 관한 연구

김 의 정[†] · 김 태 균^{††}

요 약

본 논문에서는 오프라인 한자의 특징 추출을 위한 최대 블록화 방법(Maximum Block Method:MBM)을 제안한다. 최대 블록화란 처음 찾아진 화소로부터 블록을 점차 키워 나아감으로서 입력문자의 골격선과 특징점을 추출하는 방법이다.

기존의 세선화 방법에서는 국소적인 잡음의 영향과 굴곡점, 누락점 등에서 발생하는 왜곡 현상이 특징 추출을 어렵게 하는 단점이 있었다. 본 논문에서는 최대 블록 생성으로 인하여 잡음블록의 제거와 합성으로 직선과 사선획을 찾음으로서, 정확한 특징점 추출하는 방법임을 입증하였다. 실험 데이터로는 프린트 결과물, 중,고등학교 한자 교과서와 기타 문서 등을 사용하였다. 한자 이외에도 한글 및 영, 숫자를 입력하여 실험한 결과, 인식의 전처리 과정인 골격선 추출과 획의 특징 추출에서 매우 효과적인 방법임을 확인하였다.

A Study on the Feature Extraction of Strokes using the Maximum Block Method

Eui-Jeong Kim[†] · Tae-Kyun Kim^{††}

ABSTRACT

In this paper the Maximum Block Method is suggested for the Feature Extraction of Strokes of off-line Chinese characters. The Maximum Block Method is a technique which enlarges the block from the first found pixel that extracts the skeleton and features of the input characters.

The Maximum Block Method is an adequate technique for the correct extraction of the features since the existing thinning methods have shortcomings of making the feature extraction difficult from the distortions generated from the effects of the partial noises, inflection points and blemishes.

The printed outputs and Chinese books of the middle and high school students, and other materials are used for the test. It was found that the Maximum Block Method is also an effective technique for the extraction of skeleton line and features, which is the preprocessing of the pattern recognition, for the Korean characters and English as well as Chinese characters.

1. 서 론

정보화 시대로 변해가는 현대사회에서 문서의 양과 종류가 증가함에 따라, 자동으로 입력할 수 있는 방법으로서, OCR(optical character recognition)의 필요성이 증대되어 수십년전 부터 이에 대한 연구가 수행되어왔다.

국내의 경우에도 문자 인식이 시판되고 있고, 한

† 정 회 원: 시스템공학연구소 컴퓨터비전연구실

†† 정 회 원:충남대학교 컴퓨터공학과

논문접수:1996년 5월 30일, 심사완료:1996년 8월 9일

자 및 여러 문자를 같이 인식하는 방법들이 많이 제안되고 있다. 특히, 한자의 경우는 대부분 직선과 사선으로 복잡한 획(stroke)의 구조로 조합되어져 있고, 한글과 달리 문자의 구조상 교차점(cross point)이 많이 형성되어 있는 것이 특징 추출을 어렵게 하였다.

기존에 제안된 골격선 추출 방법중 세선화(thinning) [1][2][6]의 경우에는 패턴의 외곽(edge)에 있는 화소들을 골격이 남을 때 까지 제거하는 방법으로서, 국소적인 잡음(noise)에 영향을 받기 쉽고, 특징점이 불안정하여 획을 정확하게 찾기 위해서 특별한 처리를 요하게 된다. 그 밖에도 문자의 구조적 특징을 찾는 방법으로서, LAG(line adjacency graph)[3]는 획의 변화가 심한 곡선 문자에서는 골격선을 추출하기가 어렵다. 외곽선 매칭에 의한 골격화 방법[4]은 속도가 빠르게 골격선을 추출할 수 있으나 문자의 선폭이 일정해야 한다는 단점이 있으므로 명조체, 궁서체와 같은 경우에는 적용하기 어렵다. 그 외에도 문자의 특징을 추출하는 방법으로서는 여러 가지가 있다[5][7][8].

본 논문에서는 최대 블록화[9]의 장점을 최대한 이용하여 국소적인 잡음을 제거하는 방법과 직선획과 사선을 분리 추출하는데 중점을 두었다. 그리고 추출된 수평(H), 수직(V), 좌사(L), 우사(R) 블록의 방향 특징 성분을 추출한 후 잡음 제거 및 합성을 통해 획을 추출하게 된다. 또한 추출된 획 끝점의 연결 관계를 보아서 4가지 형태별로 특징점을 추출하고, 그밖의 특징들로서는 획의 추출순서, 획의 길이와 위치 특징들을 이용하여 인식에 필요한 특징을 얻을 수 있었다.

본 논문의 구성은 다음과 같다. 2장에서는 최대 블록화 방법과 블록들의 방향 추적방법, 3장에서는 획 분류를 위해 수직, 수평 성분과 사선 성분의 분리 및 합성법과 잡음 성분의 제거 조건을 통한 정규화된 사선획 추출법에 대하여 논하였다. 4장에서는 인식의 중요한 정보인 특징점 추출에 대하여 논하고, 5장에서는 실험결과를, 6장에서는 본 논문의 결론을 기술한다.

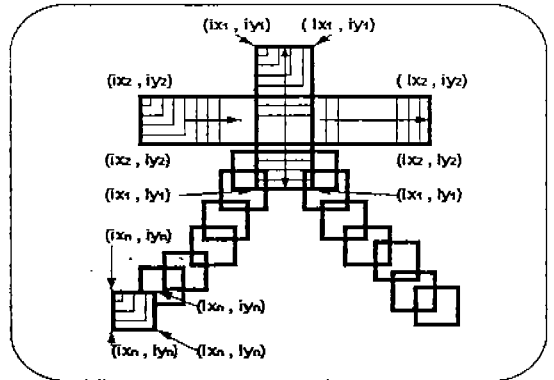
2. 최대 블록 생성법

대부분의 한자는 획의 구성이 직선과 사선으로 조합되어진 문자이다. 이러한 한자의 특징구조들을 정

확하게 추출하기 위해서 최대 블록화 방법을 사용한다. 특히 현재 사용되고 있는 문서체들이 다양한 폰트와 크기를 가지고 있어서 각기 다른 특징들이 추출되기 때문에 이러한 영상을 양자화 시킬 수 있는 알고리즘이다. 본 논문의 장점은 정확한 직선획의 추출과 획 주변의 국소적 잡음을 제거하며, 원영상 보다 블록의 좌표값 만으로 처리되기 때문에 특징 추출시 빠른 속도를 낼 수 있다.

최대 블록 생성은 획의 시작점에서 부터 블록을 점차 키워 나가는 방식이다. 이 과정에서 초기점 ix (좌측 초기점)와 iy(위쪽 초기점)에서 부터 블록을 증가시켜 최종점 lx(우측 끝점)과 ly(아래 끝점)까지 이르게 되면 블록이 생성된다.

(그림 1)은 한자 “大”자의 세로 방향(ix1, iy1), (lx1, ly1)과 가로 방향(ix2, iy2), (lx2, ly2) 그리고 사선 방향(ixn, iyn), (lxn, lyn)의 블록의 위치를 표현하고 있다.



(그림 1) 한자 “大”의 최대 블록 생성에
(Fig. 1) A maximum block method generation of the chinese character “大”.

이렇게 생성된 최대 블록의 정보는 각 블록의 꼭지점, 블록의 겹친 정도와 블록의 크기 및 진행 방향의 특징을 가지고 문자의 골격선 및 획의 특징을 추출하는데 사용한다.

3. 획의 분리를 위한 방법

최대 블록화는 수직, 수평성분의 획추출에 매우 효

과적인 방법이고, 사선획의 경우에는 사선성분이 수직, 수평성분을 가진 여러개의 블록으로 구성되어 겹쳐져 나타난다. 그러므로 사선의 경우는 겹쳐진 블록들 중에서 사선성분을 추적해 나가야한다. 이러한 블록들을 사선과 수직, 수평성분으로 분리하기 위해서는 여러 관점에서의 접근이 필요하다. 단순한 길이나 면적의 특징으로는 서로의 경계가 불분명해져서 오인식될 경우가 생기기 때문이다. 또한 잡음의 영향도 결과를 좋지 않게 만들게 된다. 따라서 다음의 여러 단계를 거치면서 원하는 획성분들을 분리한다.

3.1 수직, 수평 성분 블록의 합성 및 잡음 제거

본 절에서는 블록중에서 수직, 수평성분이 될 가능성이 있는 블록들을 합성 또는 제거함으로써 잡음을 없애고 간략화하여 단순화된 블록들로 만든다. 이때 수직, 수평성분의 블록과 그외의 블록을 구분 하는 것은 수직, 수평획 외에 사선성분 획을 정확하게 구분하기 위해서이다. 이러한 블록의 구분 방법은 블록의 가로, 세로의 비율에 의하여 쉽게 알 수 있다.

수직, 수평성분의 처리에서 제거 또는 합성될 수 있는 블록들의 대상은, 큰 블록과 75%이상 겹쳐진 블록은 큰 블록에 합성시킨다. 그리고 가로나 세로의 길이 중 1화소(pixel) 이하인 작은 블록은 제거한다. 그외의 블록들은 3.2과정을 거치게되어 사선으로 구분된다. 나머지 정확히 구분하기 어려운 블록은 3.2.3 절에서 다시 수정 과정을 거치게 된다.

이러한 처리 만으로도 많은 잡음이 제거되고 블록의 수가 줄어들게 되어, 1차적인 잡음제거와 획추출을 위한 작업이라 할 수 있다.

3.2 사선성분 블록의 추출

3.1과정을 거치면서 간단해진 수직, 수평성분의 블록과 합성에서 제외된 나머지 블록들을 합치면 최대 블록화한 최초의 블록들 보다 잡음이 제거된 필요한 블록들 만이 남는다. 이 블록들로부터 사선성분을 분리하는데는 여러 방법이 있을 수 있으나, 실제의 경우에는 여러 제약에 의해서 사용할 수 있는 방법이 한정된다.

3.2.1 사선의 판단

-블록이 사선 성분인지 알 수 있는 방법은 그 형태

가 수직, 수평의 획들과는 다른 형태를 가지고 있기 때문에, 사선성분은 대부분의 경우에 정방형에 가까운 형태를 하고 있으며 여러개의 블록들로 연결되어 있다. 그러므로 가로, 세로의 비율과 블록들의 연결된 형태로서 사선임을 판단 할 수가 있다.

그러나 사선이 수직이나 수평에 가까운 기울기 즉, 완만한 경사의 각도를 가지면 그 구성 블록들이 길어져서 사선에서 제외되어 직선으로 착각하는 경우가 생긴다.

그러므로 본 논문에서 접근한 방법은 블록의 외곽 꼭지점을 이용한 것으로서, 이러한 방법은 수학적으로 정확하게 외곽선을 추적하여 중심 골격선을 찾는 것이다. 그러나 이방법은 실제 데이터에서 발생할 수 있는 잡음과 획의 변화에서 오는 영향을 구분해야 하는 어려움이 있다.

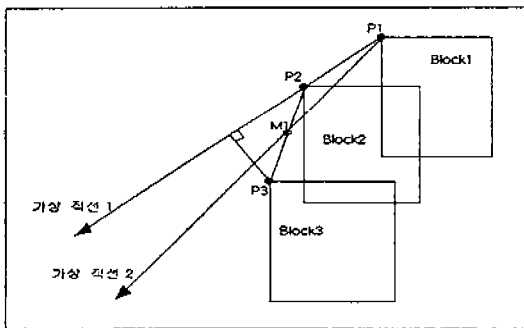
본 알고리즘은 블록화된 형태가 다음과 같으면 사선 성분이라 판단하여 사선획을 얻을 수 있다.

- 원래의 입력 영상의 외곽선에 최대블록화 하여 각 블록의 꼭지점이 존재한다.
- 찾고자 하는 사선은 직선획의 형태이다.
- 어떤 사선은 여러개의 직선으로 근사될 수도 있다.
- 근사되지 않은 점들은 잡음 또는 수직, 수평성분의 블록에 포함되므로 사선 성분에는 아무런 영향을 주지 않는다.

위의 특징들을 이용하여 시작점과 두번째점을 정의하고 간단한 산술식에 의해서 일정한 편차를 허용하면서 점들을 포함시키면 근사된 직선이 만들어진다. 이 근사된 직선은 포함된 점들의 산술적인 평균점을 지나간다. 이러한 조건을 만족하는 단순한 형태의 블록일 경우에는 이 직선이 바로 사선이다. 하지만 실제의 경우에는 직선이 아닌 곡선에 가까운 획도 많이 있다. 또한 획의 변화는 서서히 일어나기도 하고 급하게 일어나기도 한다. 즉, 일정한 편차 만으로는 허용 한계에 따라서 상이한 결과를 보인다. 그러므로 이러한 방법 만으로는 한계가 있다. 또한, 단순히 주어진 점들로 근사선을 찾는 것이 아니라 근사 가능한 점들을 찾아가야 한다. 본 논문에서는 이러한 이유로 다음과 같은 직선으로 근사하여 사선을 판단하는 방법을 사용했다.

- a. 시작되는 블록의 꼭지점을 P1점으로 하고 P1점에 연결된 다음 블록의 꼭지점인 P2 점을 찾는다.
 - b. P1점과 P2점을 연결하여 가상 직선1을 만들고, P2점을 현재의 점으로 한다.
 - c. 가상 직선1과 P3점을 직각으로 거리를 계산하여 최단거리를 계산하여 기준치 거리 이내에 있는 점이면서 사선 성분에 포함시킨다. 가상 직선1은 기준치 거리에 포함된 점에 의해서 가상직선이 수정된다.
 - d. P2점과 연결된 블록의 P3점을 찾아서 중간점인 M1점을 찾아 가상 직선2를 생성한다.
 - e. d의 과정에서 더 이상 포함될 점이 없거나 연결된 블록이 없을 때까지 반복한다.
- 획의 방향이 갑자기 변하거나 조건을 만족하지 못하면 추적을 멈추고 다음블럭 부터 다시 시작점인 P1점으로 부터 시작한다.

이 방법에서는 e의 과정에서 수직, 수평의 성분이 접합되어 사선이 단절이 되는 경우 3.2.2의 방법을 사용하여 단절을 막는다. (그림 2)는 가상 직선 근사를 이용하여 사선 블록을 추적하는 방법을 설명한 것이다.

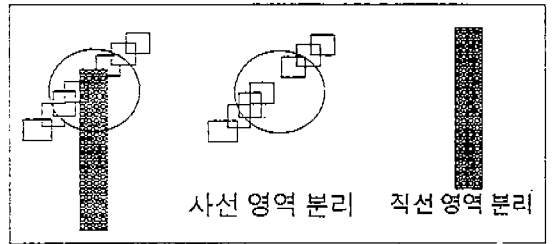


(그림 2) 블록의 꼭지점 P1을 찾은후 P2, P3의 외곽 꼭지점을 추적하여 가상의 사선으로 방향성분 결정
(Fig. 2) The direction character as a virtual orbique line by searching vertices p2 and p3 after finding vertex p1 of the block.

3.2.2 수직, 수평성분과 접합된 사선성분의 보완법
수직, 수평획과 사선획이 교차하거나 인접되어 있는 경우에 블록의 크기를 제한하여 분리하면 사선획의 일부가 수직, 수평획의 성분으로 포함되어 사선획

이 불완전해 지거나 끊어지는 경우가 있다. 이러한 원인을 분석한 결과 수직, 수평획과 사선획이 교차되면 사선획 성분의 일부가 수직, 수평성분획에 포함되기 때문으로 나타났다.

다음 (그림 3)과 같이 직선과 사선의 블록의 분할시 사선획이 연결이 되지않는 것을 볼 수 있다.

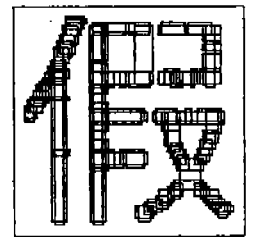


(그림 3) 사선과 직선의 교차에서 획의 분할형태
(Fig. 3) Segmentation of the stroke at the cross-points of oblique line and straight line.

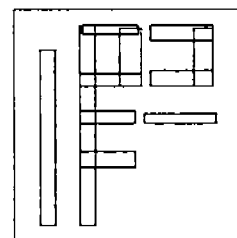
또한 단순한 사선 추출 방법을 사용하여 실제 데이터를 입력 후 획추출을 한 결과 (그림 4(e))의 1, 2의 경우와 같이 직선과 사선이 교차되는 부분에서 직선블럭의 분리때문에 사선획의 추출이 잘 되지 않는 경우가 생김을 볼 수 있다.



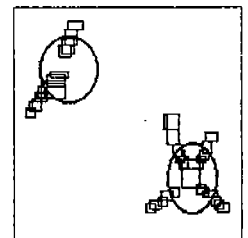
(a)입력data
(a)Input data



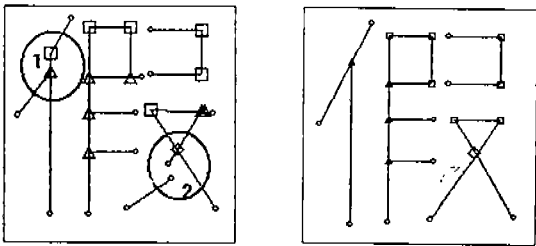
(b)최대블록화
(b)Maximum Block Method



(c)직선 추출
(c)Straight line extraction



(d)사선 추출
(d)Oblique line extraction



(e) 골격선 추출결과 (f) 사선 보완법의 적용결과
(e) Skeleton extraction (f) An Application result for the oblique interpolation

(그림 4) 교차영역에서 잘못추출된 stroke 1, 2

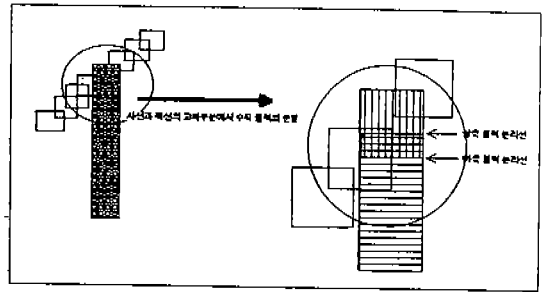
(Fig. 4) Erroneously extracted strokes 1 and 2 in the crossed region

이러한 문제를 해결하는 방법으로, 획 추출시 단절되는 획의 보완법으로 다음과 같은 두가지 알고리즘을 고안하였는데, 다음과 같이 사용한 a, b 방법중 b의 방법을 사용하여 획의 단절을 막을 수 있었다.

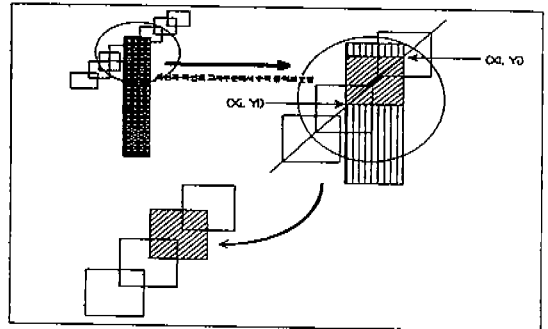
방법 a. 직선과 사선이 교차되는 경우 직선 부분을 절단하여야 하는데, 우선 직선 블록의 하측 분리선을 상위 블록의 세로 길이 만큼 절단하여 사선을 연결 하였다. 그리고 직선의 정확한 추출을 위하여 하측 분리선 보다는 조금 위인 상측 분리선을 정하여 절단한다.

방법 b. 직선에서 사선의 점과 교차되는 (X1, Y1) 점과 (X2, Y2) 점을 추출하여 블록을 생성 후 직선의 블록은 그대로 직선추출을 하는 방법으로서 우선 첫번째 블록에서 임의의 사선을 만들어 생성된 블록을 연결 블록으로 하고 직선블록은 그대로 두어 직선, 사선을 구한다.

본 논문에서는 (그림 5)와 같이 a, b 두가지 방법을 사용하여 보았으나 최종적으로 b의 방법을 사용하여 획의 연결을 강하게 유지했으며, 블록이 작아져서 잡음으로 나타날 수도 있으므로 만들어진 블록의 크기보다 1.5배 이상일 때만 블록을 유지하고 그 이하는 제거한다. 이러한 처리로 교차점이나 분기점 등의 문제를 해결한 결과로서, (그림 4(f))와 같이 교차되거나 사선과 직선의 점합 부분에서 향상된 획추출 결과를 얻을 수 있었다.



(a) 방법 a를 이용한 사선 추출
(a) Oblique line extraction using method "a".



(그림 5) (b) 방법 b를 이용한 사선 추출
(Fig. 5) (b) Oblique line extraction using Method "b".

3.2.3 사선성분에 포함된 잡음 제거

앞절에서 만들어진 사선처리 중에서 잡음이라고 판단되어 제외된 블록이 있는데 이러한 블록을 모두 제거하는 것이 아니라, 사선블록들 중에 끊어진 블록을 연결할 수 있는 블록은 다시 첨가한다. 그리고 잡음 제거 대상으로 판단하는 블록은 블록의 크기가 작은 것과 방향이 같고 블록이 중복되어있는 것이다.

3.3 획의 골격선 추출

이 과정에서는 획을 하나의 골격선으로 만드는데 우선 블록의 합성을 최종적으로 마치면 완성된 직선과 사선의 문자 블록을 이룬다. 이 두 성분을 분리하는 과정 동안 방향 및 특징은 블록이 이미 직선과 사선을 분리하는 동안 이미 가지고 있기 때문에 이러한 정보를 이용하여 골격선을 추출한다. 골격선 추출 방법은 다음과 같다.

우선, 직선의 경우에는 블록의 길이방향 즉, 수평, 수직회심을 찾아서 획의 양쪽 중심점을 연결하여 골격선으로 만든다. 사선의 경우에는 사선임을 판단하

는 가상직선에 연결된 시작과 끝 블록의 대각점을 연결하여 생성 되어진다. 이때 만든 골격선들은 획들의 연결 부위에서 항상 교차하게 되는데 추후 특징점 추출시에 끝점을 수정하여 원래의 정확한 특징을 얻기 위해서는 4.2절에서 자세히 설명하기로 한다.

4. 인식을 위한 특징 추출

정규화된 블록들의 방향 성분이 결정되면 인식을 위하여 획들의 연관 관계를 특징코드로 저장하여야 한다. 이러한 특징 코드 생성 규칙들의 순서는 우선 획의 4방향 특징을 찾는다. 획의 연관 관계 특징인 굴곡점, 분기점, 교차점을, 그리고 처음 찾아진 획에서 연결된 획의 좌에서 우로, 위에서 아래로 획의 추출 순서 특징을 결정한다. 또한 길이의 특징을 3단계로 구분하고, 위치특징으로써 연결된 획들의 무게 중심을 찾아 구한다.

4.1 방향 특징

본 논문에서는 오프라인 인식에서 일반적으로 사용하는 4방향 특징을 사용한다.

방향 특징 코드로는 ① “-” ② “/” ③ “|” ④ “\”와 같이 사용한다.

4.2 획의 연관 관계 특징

일반적으로 한자 및 한글 등은 문자 구조상 직선과 사선의 조합으로 표현이 가능하므로 골격화된 문자

의 특징점은 (그림 6)과 같이 단점(end point), 굴곡점(inflection point), 분기점(branch point), 교차점(cross point)들로 추출하였다. 다음과 같이 특징점은 4종류의 코드로서 표현된다.

본 방법은 획의 특징점을 정확히 추출 하였는가를 시각적으로 표현하기 위하여 다음과 같은 기호로도 표현하였다. 획의 연관 관계특징 [e: 단점(O), t: 굴곡점(□), b: 분기점(△), c: 교차점(◇)].

특징점인 교차점을 찾는 방법으로서, 두 획의 중간 지점이 연결된 점을 찾는다. 여러가지 찾는 방법[6]이 기술 되었지만, 본 논문에서는 다음과 같은 산술식에 의하여 찾는다.

두 직선 A: (x_{a1}, y_{a1})~(x_{a2}, y_{a2}), B: (x_{b1}, y_{b1})~(x_{b2}, y_{b2})에 대해서 각각 다음과 같은 직선의 방정식을 얻을 수 있다.

$$X_a = \frac{(b_b - b_a)}{(m_a - m_b)}$$

$$Y_a = m_a * \frac{(b_b - b_a)}{(m_a - m_b)} + Y_{a1} - m_a * x_{a1}$$

단, $Y_a = m_a * X_a + b_a$, $Y_b = m_b * X_b + b_b$

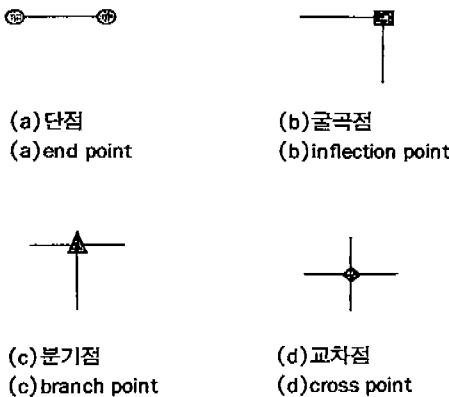
$$m_a = \frac{(y_{a2} - y_{a1})}{(x_{a2} - x_{a1})}, m_b = \frac{(y_{b2} - y_{b1})}{(x_{b2} - x_{b1})}$$

$$b_a = y_{a1} - m_a * x_{a1} = y_{a2} - m_a * x_{a2}$$

$$b_b = y_{b1} - m_b * x_{b1} = y_{b2} - m_b * x_{b2}$$

이렇게 해서 얻은 X_a, Y_a가 두 직선 A, B에 포함되는지에 따라서 교차점의 존재와 위치를 알 수 있다. 이러한 처리를 해야하는 필요성은 정규화된 블록을 가지고 골격선 추출시 획과 획의 접합부분에서 항상 교차하는 결과를 볼 수 있다. 이러한 결과는 획의 골격선 추출시 획의 중심선을 연결하여 추출 하였기 때문에 획의 연결점이 모두 교차되는 것이다. 그러므로 모든 교차점을 각 특징별로 구분하기 위해서 위의 직선의 방정식을 사용한 결과, 각 특징점은 굴곡점, 분기점, 교차점의 관계로 추출되어 진다.

- a. 단 점 : 획의 끝에 다른획이 연결되어 있지 않다.
- b. 굴곡점 : 두 획의 끝점과 교점이 같을때

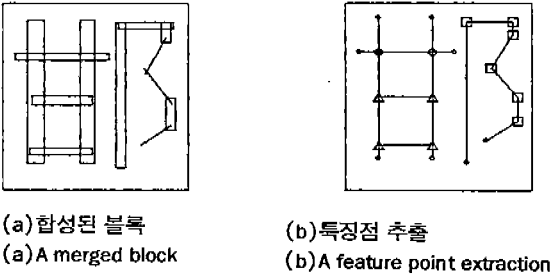


(그림 6) 4가지 특징점 구분

(Fig. 6) A classification of the 4 feature points.

- c. 분기점: 한 획의 끝점과 교점이 같고 다른 획은 갈지 않을때
- d. 교차점: 교점이 두획의 끝과 거리가 멀때

등의 특징으로 획간의 연관 관계인 특징점이 찾아진다. (그림 7)의 (a)에서는 최종적인 합성된 문자 블록을 (b)에서는 특징점 추출 결과를 보인다. 다음과 같은 특징점과 방향 성분은 인식 단계에서 중요한 정보로 사용된다.



(그림 7) 합성된 블록의 골격선 및 특징점 추출
(Fig. 7) An extraction of the skeleton and feature of the merged block.

4.3 획 추출시 순서 특징

연결된 골격획들의 정확한 특징 추출순서를 부여하기 위하여 본 방법을 추가하였다. 이 순서는 사람들이 손으로 문자를 쓰는 순서와 비슷하도록 순서를 주었다. 추출 순서는 처음 찾아진 획에서 연결된 획의 좌에서 우로, 위에서 아래로의 순서로 정한다. 단, 처음 찾아진 획에서 가장 가까운 획 부터 찾기 시작한다.

4.4 획의 길이 정보 특징

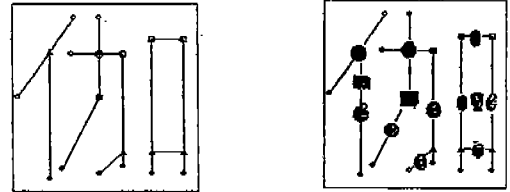
문자획의 길이는 필요에 따라 조금씩 다르다. 예를 들어 모양은 같지만 길이가 긴 획이나, 길이가 짧은 획은 구조상 많은 차이가 있다. 그러므로 획의 길이에 특징을 두어서 추출 하였다. 다음은 추출된 문자를 X, Y축의 일반적인 문자폭을 최대로 하여 최대길이면 "3/3", 제일 짧으면 "1/3" 등으로 길이 구분 하였다.

예를들어 "·"와 "।"과 같이 골격선 추출 결과가 크기는 다르지만 같은 모양으로 추출되는 경우가 생

기므로 이러한 문제점을 해결하기 위해서 다음과 같은 길이 특징 코드를 사용하였다.
길이 특징은 ① "1/3" ② "2/3" ③ "3/3"와 같다.

4.5 위치 특징

추출된 연결 획들의 무게 중심을 구하여 3×3의 블록 형태의 마스크에 각각 연결된 획의 무게 중심점을 표시하여 형태별로 위치 특징을 구분한다. 이러한 특징으로 정확한 문자의 위치를 기억할 수 있고, 인식할 때 대분류 및 소분류 과정에서 분류 특징을 구하는데 적합하다.



(a)골격화 및 특징추출 (b)위치결정(M)
(a)The Skeletonization and feature extraction (b)The Location determination

(그림 8) 연결된 각획의 무게 중심점 결정 예
(Fig. 8) An example of decision for the centers of the each crossed lines.

(그림 8)에서 처럼 획의 무게 중심(M)을 찾는 예를 볼 수 있고, 다음과 같은 순서로 연결된 획의 무게 중심을 구한다.

- a. 연결된 각 획 마다 중심점(1, 2, 3...)을 구한다.
- b. 구해진 중심점들 중 M값을 찾는다.
- c. 3×3의 각 블록 위치에 기억하여 CODE로 저장한다.

5. 실험결과

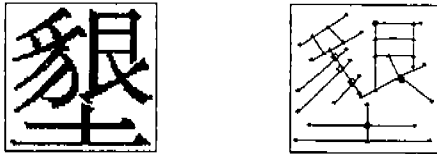
본 논문에서는 특히 오프라인 한자 및 모든 문자를 대상으로 하여 획의 특징점을 추출하는 알고리즘이다. 특징을 찾기 위해 프린터 인쇄물과 중학교, 고등학교 한자 교과서 등에서 얻어진 한자 및 한글, 영, 숫자를 사용하였고, MS-Windows 환경에서 본 방법을 구현하였다. 문자 영상이 입력되면 우선, 최대 블록화

방법을 이용하여 불룩을 확장시키고 각 문자의 획을 블록으로 구성한다. 본 논문에서는 타 논문과 비교하여 다른점으로서 직선 획을 정확히 추출하면서 잡음 성분도 제거할 수 있다. 또한, 사선을 정확히 추출하

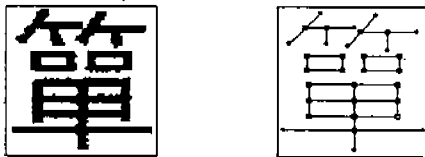
기 위해서 직선과 사선을 분리시킨 후 잡음 제거와 합성 과정을 통하여 골격선을 추출하였다.

추출된 골격선의 특징을 단점, 굴곡점, 분기점, 교차점들을 이용하여 인식에 필요한 구조 정보를 추출하였다. 한자 뿐 만 아니라 한글 및 영, 숫자 등을 실험한 결과 유용함을 입증하였다.

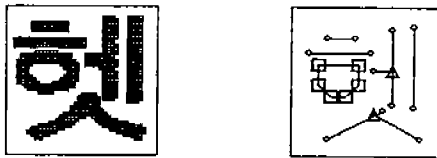
(그림 9)는 한자 및 한글과 영, 숫자의 골격선 및 특징점 추출 결과이다.



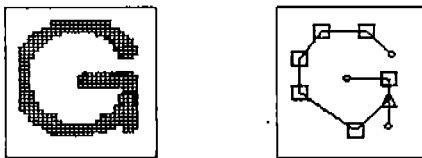
(a) 명조체 한자의 특징점 추출 결과
(a) An Extraction of the Ming Dynasty Style Chinese character



(b) 고딕체 한자의 특징점 추출 결과
(b) An Extraction of the Gothic Chinese character



(c) 한글의 특징점 추출 결과
(c) An Extraction of the Korean character

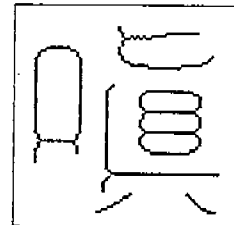


(d) 영, 숫자의 특징점 추출 결과
(d) An Extraction of the Alphanumeric character

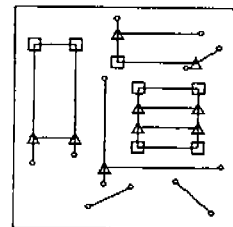
(그림 9) 여러 문자의 특징점 추출 결과
(Fig. 9) The extractions of different characters



(a) "헛"자의 입력 영상
(a) Input image of the "헛"



(b) Hilditch의 세션화 결과
(b) A thinning by Hilditch method



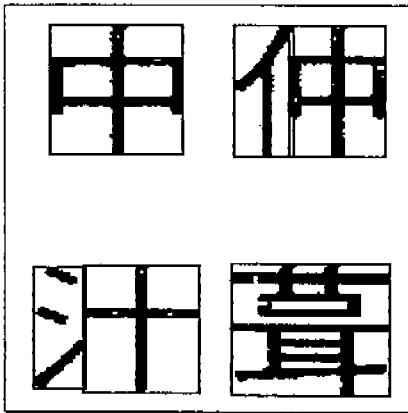
(c) 본 방법을 이용한 골격선 및 특징점 추출 결과
(c) The MBM extraction for the skeleton and feature

(그림 10) Hilditch의 세션화 결과와 최대 블록화 방법의 비교

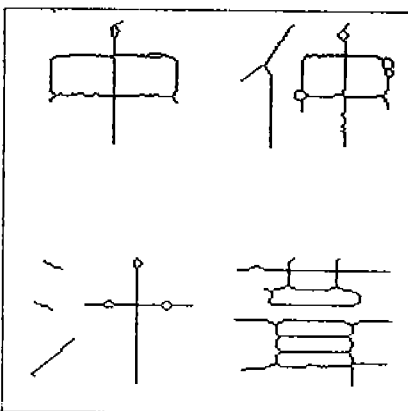
(Fig. 10) A comparison of a Hilditch's thinning result and an MBM result

기존의 세선화 방법과 비교하여 실험 해본 결과 세선화에서는 획의 특징점이 되는 골극점 등에서 발생하는 왜곡 현상 등으로 특징 추출 결과가 만족하지 않았다. 그러나 본 방법에서는 직선과 사선획을 분리함과 동시에 잡음 제거와 정확한 직선획을 얻을 수 있었고, 특징 추출 시간도 상당히 빠른 결과를 얻을 수 있었다.

(그림 10)에서는 동일한 입력 영상을 입력한 결과 Hilditch방법[1], [2]을 이용한 세선화 결과와 본 최대 블록화 방법을 이용한 골극선 및 특징점 추출의 결과를 보인다.



(a) 원래 입력영상에 잡음을 섞인 영상
(a) An original image with noises

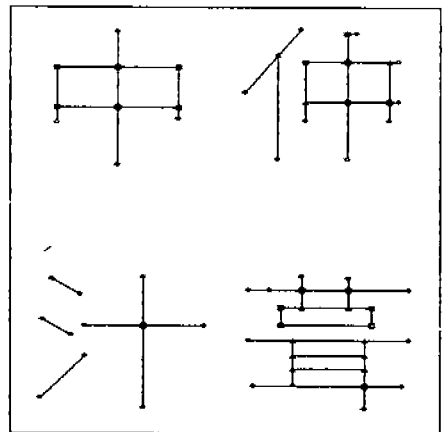


(b) 누락점이 생길 경우 취약점으로 나타난 세선화의 경우
(b) A thinning with missing point appeared as weakpoint

(그림 11) Hilditch의 방법을 이용한 세선화 결과
(Fig. 11) A thinning result of the Hilditch Method

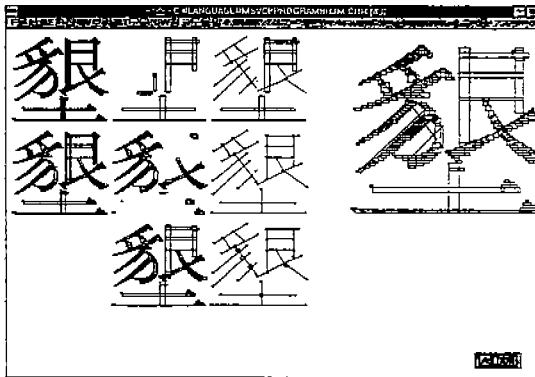
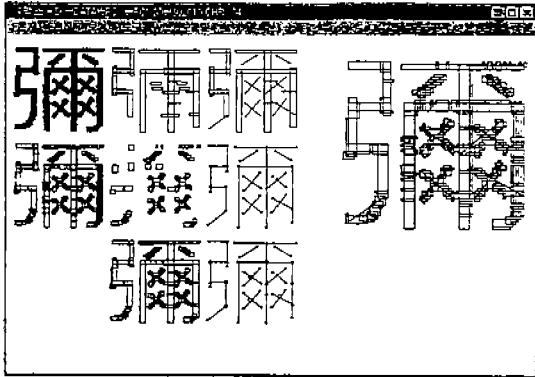
(그림 11) 처럼 세선화의 결과는 획의 누락점이나 골극점 등에 상당히 취약함을 볼 수 있다. 이러한 경우 인식을 못하거나 또 다른 방법을 취하기 때문에 원영상 보전과 빠른 시간 등을 기대하기 어려운 단점이 있다. 또한 특징 결과를 인식에 사용하기 위해서는 또 다른 직선화 작업을 하는 등 번거로움이 있는데, 이러한 작업이 오히려 오인식을 일으키는 요인이 될 수도 있다.

(그림 12)에서는 (그림 11(a))의 입력영상을 최대 블록화에서는 잡음이나 누락점에서 정확하게 획 추출한 것을 볼 수 있다. 이러한 인위적인 누락점을 생성하여 실험한 이유는 특히 칼라 영상에서 이진 영상으로의 변환시 이러한 경우가 많이 발생하기 때문이다. 따라서 본 방법은 이러한 변환된 이진 영상에도 강한 획 추출법이라고 볼 수 있다.



(그림 12) 인위적으로 원영상에 잡음을 섞어 입력한 최대 블록화 결과
(Fig. 12) The MBM result of the original image with artificial noises

(그림 13)에서는 고딕체, 명조체의 한자 데이터를 입력하여 최대 블록 생성과 직선, 사선블록의 분리, 골극선 추출 후 최종적인 직선의 방정식을 이용한 획 특징의 순서로 결과를 보인다.



(그림 13) 한자 고딕 및 명조체를 입력 후 골격선 추출 순서와 특징 결과 까지의 모든 처리 과정

(Fig. 13) The process of the skeleton extraction and feature extraction after the input of the ming dynastic style and gothic chinese characters

6. 결 론

본 논문에서는 최대 블록 생성을 이용하여 획의 직선 성분과 사선 성분을 분리하였고, 굵음 블록은 별도로 처리하여 제거 및 합성의 조건에 포함시켜 처리함으로써 골격선 및 특징점을 추출할 수 있었다. 사선 블록의 추적 방법으로서 블록의 외부 꼭지점을 이용하여 임의의 직선근사를 통하여 포함된 사선을 추출 할 수 있었다. 그리고 사선과 직선이 교차하는 부분에서는 단절된 사선블록을 보완하여 주는 알고리즘을 사용하여 획의 단절을 막을 수 있었다.

한자의 경우는 대부분 직선과 사선으로 복잡한 획(stroke)의 구조로 조합되어져 있고, 한글과 달리 문자

의 구조상 교차점(cross point)이 형성되어 있는데, 이러한 점들이 본 알고리즘에 적합함을 실험을 통하여 확인하였다. 그 외에도 여러 문자를 실험한 결과 오프라인 인식에 필요한 문자획의 정확한 골격선과 특징점을 추출할 수 있었다. 이러한 특징 결과를 구조적 문자인식에 적용하면 기존의 방법 보다 향상된 인식율과 빠른 속도를 기대할 수 있다. 또한 기존의 세선화 방법과 본 방법을 비교해 본 결과 본 알고리즘이 향상된 방법임을 실험을 통해 알 수 있었다. 앞으로 본 알고리즘을 기반으로 다양한 문서체들과 필기체 문서 까지도 인식할 수 있는 연구가 계속 되어져야 할 것이다.

참 고 문 헌

- [1] C. J. Hildith, "Linear Skeletons from Square Cupboards," in Machine Intelligence 4(Eds. B. Meltzer and D. Michie), American Elsevier, New York, pp. 403-420, 1969.
- [2] N. J. Naccache and R. Shinghal, "An Investigation into the Skeletonization Approach of Hilditch," Pattern Recognition, Vol. 17, No. 3, pp. 279-284, 1984.
- [3] S. Kahan, T. Pavlidis and H. S Baird, "On the recognition of printed characters of any font and size," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, Mar. pp. 274-288, 1987.
- [4] C. C. Han and K. C. Fan, "Skeleton Generation of Engineering Drawings Via Contour Matching," Pattern Recognition, Vol. 27, pp. 261-275, 1994.
- [5] H. Iwamoto, N. Hosokawa and S. Ozawa, "An Expansion of Pen Movement Stroke Extraction Method to Handwritten Chinese Character Recognition," 日本 電子通信學會論文誌, Vol. J69-D, No. 2, pp. 214-220, 1986.
- [6] H. Ogawa and K. Taniguchi, "Thinning and Stroke Segmentation for Handwritten Chinese Character Recognition," Pattern Recognition, Vol. 15, No. 4, pp. 299-308, 1982.
- [7] W. S. Hsu, Kunitake. Takahashi, Shinji Ozawa,

Hirochi Fujita, "Ordered Stroke Extraction Method for Printed Chinese Character Recognition," 電子通信學會論文誌, Vol. J65-D No. 2, pp. 266-273, 1982.

[8] Hiroaki. Iwamoto, Nobuo Hosokawa, Shinji Oza-wa, 電子通信學會論文誌, Vol. J69-D, No. 2, 1986.

[9] 김의정, 김태군 "오프라인 한자 특징을 위한 최대 블록화 방법에 관한 연구," 한국정보과학회 봄 학술발표 논문집, pp. 225-228, 1995.



김 태 군

1971년 서울대학교 공업교육학과(학사)

1980년 일본동경공업대학 대학원 물리정보공학과(공학석사)

1984년 일본동경공업대학 대학원 물리정보공학과(공학박사)

1974년~현재 충남대학교 컴퓨터공학과 교수
관심분야: 문자인식, 영상처리, 멀티미디어



김 의 정

1993년 충남대학교 대학원 컴퓨터공학과(공학석사)

1996년 충남대학교 대학원 컴퓨터공학과(박사과정수료)

1993년~현재 대전산업대학교 전자공학과 시간강사

1997년~현재 시스템공학연구소 컴퓨터비전연구실 연구원

관심분야: 문자인식, 영상처리, 멀티미디어