오프라인 문서에서 개별 문자 추출과 한자 인식에 관한 연구

김 의 정[†] · 김 태 균[††]

요 약

본 논문에서는 인쇄체 문서 인식을 위한 전처리 과정인 개별 문자 추출 방법과 인식 방법에 대하여 논한다. 전처리에서는 접촉 문자(touching character) 또는 겹친 문자(overlapped character)등과 같이 추출이 곤란한 문자를 개별 문자로 추출하는 것이다. 기존의 문자 분리 방법에서는 투영(projection)에 의한 방법과 외곽선(edge) 추적에 의한 방법 등을 사용하여 왔으나, 제안된 방법은 문자열 추출 후 한번의 투영으로 연결 화소를 이용하여 개별 문자를 추출한다.

인식을 위해서는 최대블록화 방법(Maximum Block Method:MBM)을 이용하여 특징 추출을 한다. 최대블록화는 문자를 투영 중 처음 찾아진 점에서 부터 최대한 블록을 확장 시키는 방법이다. 문자를 이루는 최대블록들을 직선 블록과 사선 블록으로 분리후 골격화 시킨다. 특히 한자 인식에서 기존의 상용 문자 인식기와 비교하여 향상된 인식율을 얻을 수 있었다.

A Study on the Extraction of an Individual Character and Chinese Characters Recognition on the Off-line Documents

Eui-Jeong Kim[†] · Tae-Kyun Kim[††]

ABSTRACT

In this paper, the extraction method for individual characters and the recognition method for the printed documents are discussed. In preprocessing is a technique to extract characters that are difficult to manage such as touching characters or overlapped characters. Generally in the existing segmentation methods, projection and edge detection are applied. However, in this paper an individual character is extracted by using connected pixel with one projection after the string extraction.

The Maximum Block Method(MBM) is used for the recognition. The MBM is a method to enlarge the block to the last point from the pixel that was found during projection. The maximum blocks are skeletonized after the division into straight line block and oblique line block. Especially, in the recognition of chinese characters compared to the existing method it showed improved recognition rate.

† 정 회 원:시스템공학연구소 컴퓨터비젼연구실
†† 정 회 원:충남대학교 컴퓨터공학과
논문접수:1996년 9월 18일, 심사완료:1996년 10월 21일

1. 서 론

현대 사회는 문서의 홍수 시대라고 볼 수 있다. 정보화 시대에 따라, 기존의 문서를 수작업으로 입력하는 것은 많은 시간과 인력의 낭비를 초래하여 문서를 자동으로 입력하고자 하는 관심이 고조되고 있다. 따라서 문자 인식을 위한 많은 연구가 국내외적으로 활발하게 진행되어 왔다. 국내의 경우에도 문자 인식기가 시판되고 있고, 한글이나 한자 등 여러 문자를 같이 인식하는 방법들이 많이 제안되고 있다. 특히 오프라인 인식에서 정확한 개별 문자 추출이 인식율을 좌우하는 중요한 부분으로 대두되고 있다.

본 논문에서는 연결 화소를 이용한 개별 문자 추출 방법[7]과 최대블록화[13]를 이용한 인식 방법에 관하여 논한다. 개별 문자를 추출하는 기존의 방법으로는 투영에 의한 방법과[1-5] 외곽선 추적에 의한 방법[6]이 있다. 투영에 의한 방법은 빠른 속도와 문서 영상의 분리 결과 문자 인식에 영향을 줄 수 있는 많은 정보를 얻을 수 있지만, 겹친 문자를 분리해 낼 수 없고, 자소 분리를 할 수도 없다. 분리가 곤란한 경우는 강제 분리를 한다[1]. 외곽선 추적에 의한 방법은 겹친 문자를 분리할 수 있지만 속도면에서 늦고, 인식에 이용할 획득된 정보가 적다. 본 논문에서는 투영에 의한 방법과 외곽선 추적에 의한 방법의 장점을 이용하여 연결 화소를 이용한 문자 분할을 하였다. 분할 방법으로서 투영에 의해 문자행을 추출한 후 연결 화소 블록을 구하였고, 연결 화소를 문자 구조상 조합하여 개별 문자로 추출하였다. 접촉된 문자들은 기준 블록폭을 적용해서 분리하여 실험한 결과 신문이나 기타 잡지 등에서 접촉되거나 겹친 문자를 분리하는데 향상된 결과를 얻을 수 있었다.

이렇게 얻은 개별 문자를 인식하는데 있어서 특히, 한자의 경우는 대부분 직선과 사선으로 복잡한 획(stroke)의 구조로 조합되어져 있고, 한글과 달리 문자의 구조상 교차점(cross point)이 많이 형성되어 있어 특징 추출을 어렵게 하였다. 기존의 골격선 추출 방법들 중 여러 방법이 제안되었는데 세선화(thinning)[9][10]의 경우에는 패턴의 외곽(edge)에 있는 화소들을 골격이 남을 때 까지 제거하는 방법으로, 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)[11]에서는 획의 변화가 심한 곡선 문자의 경우에는 골격선을 추출하기가 어렵다. 외곽선 매칭에 의한 골격화 방법[12]은 속도가 빠르게 골격선을 추출할 수 있으나 문자의 선폭이 일정해야 한다는 단점이 있으므로 명조체, 궁서체와 같은 경우에는 적용하기 어렵다. 본 논문에서는 최대 블록화의 장점을 최대한 이용하여 국소적인 잡음을 제거하는 방법과 직선획과 사선을 분리 추출하는 데 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 추출 순서, 획의 길이와 위치 특징들을 코드로 생성하여 인식에 필요한 특징을 얻을 수 있었다. 인식 방법으로서 결정 트리(decision tree)를 구성하여 효율적인 트리를 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 연결 화소를 이용한 문자 분할 방법으로서 개별 문자 추출을 위한 방법을 제안하고, 3장에서는 최대블록화를 이용한 인식 방법에 관하여 논한다. 4장에서는 실험 결과를, 5장에서는 본 논문의 결론 순으로 기술한다.

2. 연결 화소를 이용한 문자 분할 [7]

2.1 투영에 의한 연결 화소 추출

투영에 의한 연결 화소를 추출하기 위해서는 우선 줄단위 문자행 추출을 하여야 한다. 추출된 문자행을 가지고 다시 열에 대한 처리를 하게 된다.

2.1.1 문자행 추출

문자행 추출 시 줄단위의 가로 방향으로 투영을 하면서 흑화소의 누적 분포를 판단하여 문자행의 시작점과 끝점을 결정한다.

문서 영상에서는 백화소일 때 0, 흑화소일 때 1을 나타내는 함수 f(x, y)로 표현한다.

$$f(x, y) = \begin{cases} 0 & \text{백화소} \\ 1 & \text{흑화소} \end{cases}$$

2.1.2 투영에 의한 연결 화소 추출

윗절에서 추출된 문자행을 가지고 연결 화소를 추출하기 위해서는 각 문자열을 세로로 투영하여 추적하는데, 우선 찾아진 점에서 부터 이웃화소들과 연결되었는지를 알아야 한다. 그 방법으로서 세로 투영시 백화소는 0으로 레이블링하고 흑화소에 대해서는 연속 흑화소(Black Run)를 구하여 레이블링 한다. 이러한 결과 각 레이블 값은 하나의 연결 화소 블록을 가지게 된다. 이렇게 추출된 연결 화소들을 합성과 분리 과정을 통하여 떨어진 문자는 물론 겹친 문자와 접합 문자 등을 분리 할 수 있다. 그림 3(b)에서는 연결 화소 추적을 이용하여 레이블링된 자소를 모두 개별적으로 추출한다.

아래 식 단계 2에서와 같이 세로 방향 연속 흑화소 구간 (P, Q)에서 왼쪽 이웃 화소들의 레이블 값을 참조한다. 레이블 값이 없으면 구간 (P, Q)에 새로운 값을 부여하고 레이블 값이 있으면 그 값을 구간 (P, Q)에 부여한다. 구간 (P, Q)에 레이블 값이 결정되면 그 값에 해당하는 연결 화소 블록을 구간 (P, Q)에 따라 수정한다. 만약 구간 (P, Q)의 왼쪽 이웃 화소들이 두 개 이상의 레이블 값을 갖는다면 가장 작은 값을 구간 (P, Q)에 부여한다. 나머지 값들의 블록은 구간(P, Q)의 블록에 합성한다.

단계 1. 문자행을 추출해 낸다.

1. 가로 방향 투영

$$P(y)=\sum_{x=1}^{M} f(x, y)$$

[$P(y)$는 가로 방향으로 투영한 흑화소의 합]

2. 문자행 추출

i번째 열의 시작점과 끝점은 $P(y) \geq l$일때

$LStart(i)=y$ if $P(y-1)<l$

$LEnd(i)=y$ if $P(y+1)<l$

[l은 잡영으로 추측될 수 있는 임계치]

단계 2. 연결 화소 추출

단계 1에서 추출된 행에 대해서 세로 방향으로 투영을 한다.

1. $f(x, y)=0$이면, $L(x, y)=0$

   $f(x, y)=1$이면, 연속 흑화소(Black Run)구간을 구한다.

   세로 방향 연속 흑화소 구간은 $P \leq y \leq Q$에서 $f(x, y)=1$이다.

2. 구해진 연결 흑화소 구간 (P, Q)의 왼쪽 이웃열의 흑화소의 레이블 값을 참조한다.

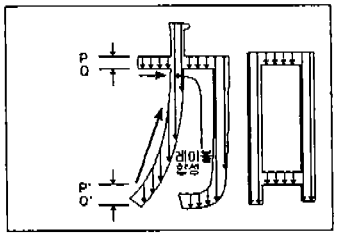
$$L(x, y)=\begin{cases} L(x-1, y) & \text{if } L(x-1, y)>0 \\ 0 & \text{if } L(x-1, y)=0 \end{cases}$$

$M=\{L(x, y)/P \leq y \leq Q$

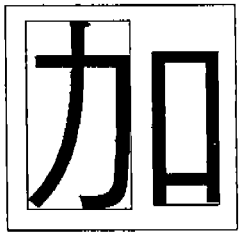
3. M의 원소가 0 하나이면 구간 (P, Q)에 $NewLable$을 부여한다.

   x=0일 때, $NewLable=0$이고 $L(x, y)=0$

(a) 입력 영상
(a) Input image



(b) 레이블화
(b) Labeling



(c) 연결화소 추출
(c) Connected pixel extraction

(그림 1) 레이블링 하는 과정
(Fig. 1) Labeling process

4. M이 0이외 두개 이상의 원소를 가지면 하나의 원소를 선택하고 나머지는 합성한다.

그림 1에서는 위 식에서 획득된 정보를 이용하여 연결 화소를 이용하여 문자 분리 과정의 예를 보인다. 그림 1(b)에서는 흑화소의 런값 P, Q와 P′, Q′ 값이 나중에 하나의 레이블로 합성 되는 것을 볼 수 있고 그림 1(c)와 같은 결과를 얻는다. 이러한 각 레이블 값은 하나의 연결 화소 블록을 가지게 된다. 이렇게 추출된 연결 화소들을 합성과 분리 과정을 통하여 떨어진 문자는 물론 겹친 문자와 접촉 문자 등을 분리할 수 있다.

2.1.3 연결 화소 블록의 수직 합성

투영에 의한 연결 화소 추적에 의해 추출된 문자 블록은 하나의 문자가 될 수도 있지만, 대부분의 문자는 조합되지 않은 자소들로 분리되어 있다. 따라서 개별 문자 단위로 추출하기 위해서는 이 자소들을 한 문자로 조합해야 한다.

한글 또는 한자의 경우 문자의 형태가 조합 문자로 이루어져 있기 때문에 이러한 문자들은 자소들의 수직적 조합에 의해 단일 문자를 이루는 경우와 수평적 조합에 의해서 단일 문자를 형성하는 경우가 있다. 전자의 경우는 "을", "를", "는" 등과 후자의 경우는 "이", "가", "에" 등을 예로 들 수 있다. 한자의 경우 역시 "示", "言", "聶" 등과 "心", "化", "小" 등으로 구분될 수 있다. 연결 화소가 하나의 문자 블록으로 추출되려면 수직적인 결합과 수평적인 조합이 필요하다. 수직 조합을 우선으로 하여 가능한 연결 화소의 문자블록을 추출한 후 여러개의 문자 블록을 하나의 문자로 합성하는 데에는 각 블록들의 겹친 정도를 사용한다. 즉 세로축으로 겹침이 심한 블록들을 하나의 블록으로 합성하게 된다. 이렇게 합성된 블록은 하나의 문자를 이루게 되고 분리된 문자나 겹친 문자들은 완전한 개별 문자로 추출하게 된다. 그림 3(b)에서 연결된 화소의 블록을 세로로 합성하여 그림 3(c)와 같이 수직축으로 합성을 한다.

수평적 결합은 한 문자가 수평적으로 떨어진 두개 이상의 자소로 분리되어 그들이 다른 자소들과 접촉되어 있는 경우가 있으므로 순서상 2.3절에서 자세히 설명하기로 한다.

2.2 접촉 문자의 분리

연결 화소 블록의 합성 후 분리되지 않은 접촉 문자들이 남게된다. 접촉 문자를 분리하기 위해서는 분리할 블록을 선택해야 한다.

2.2.1 분리할 블록의 선택

분리할 블록을 선택하는 것은 분리를 하는 방법보다 더욱 중요하다. 어떤 블록이 분리할 블록인가를 알아야 정확한 분리가 가능하다. 만일 분리하지 않아도 될 블록을 분리 한다든지, 접촉 문자를 분리하지 않아도 되는 블록으로 결정하게 되면 아무리 좋은 분리 알고리즘이라도 정확한 분리를 할 수 없기 때문이다.

본 논문에서 제안한 분리 블록을 선택하는 경우는 합성 단계에서 만들어진 블록들을 문서 영상의 한 행씩에 대해 평균 문자 블록 가로폭의 최빈수(mode)를 기준 블록 폭이라고 한다. 그 폭에 일정한 임계치를 주어서 그 이상이 되는 블록을 분리할 블록으로 판단하게 된다. 한글이나 한자의 문서는 한 행에서는 문자의 크기 변화가 심하지 않으므로 기호 문자와 같은 극단적인 값에 영향받지 않는 최빈수로 문자의 대체적인 크기를 알 수 있다. 최빈수 값이 두 개 일때, 즉 쌍둥형(bimodal)일 때는 문자행의 세로 높이에 가까운 최빈수를 선택하게 된다.

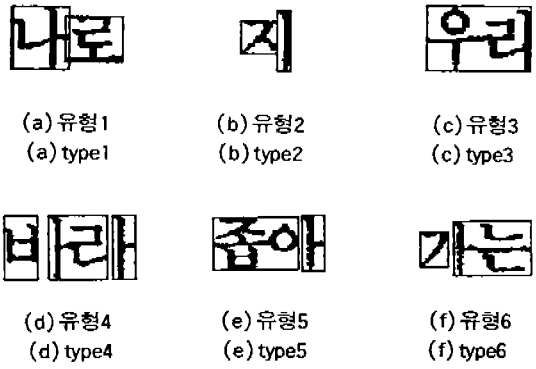
2.2.2 문자 분리

한글 문서에서의 문자 간 접촉 형태는 6가지의 유형[7]으로 대변할 수 있다. 그림 2에서는 일반적인 문서에서 문자 간의 6가지 접촉 유형을 설명한 것이다. 유형 1은 자소가 겹쳐 있으나 연결화소 추적에 의하여 분리 할 수 있다. 유형 2는 문자 단위로 추출하기 위해서는 자소 블록을 가로로 합성해야 한다. 그 나머지 3, 4, 5, 6 유형을 분리할 대상으로 삼았다.

3, 5, 6 유형은 최빈수 문자폭 보다 크기 때문에 분리 대상으로 선정되지만 유형 4는 정교한 규칙이 필요하다. 따라서 최빈수 문자폭에 가까운 문자폭을 가진 블록에 대해서는 그 특징에 맞는 블록을 선정한다.

유형 3은 대략 양쪽 끝에, 유형 5는 왼쪽 끝에, 그리고 유형 6은 오른쪽 끝에 최빈 블록폭의 분리 위치가 있다. 여러 문자가 붙은 경우 그 크기는 최빈 블록폭의 2배 이상이 된다. 이러한 경우는 유형 3으로 보고 그에 따른 분리를 한다.

각 유형별 특징에 맞는 범위에 대해서 세로로 투영하여 얻은 흑화소수들 중 최소 흑화소수를 문자간 분리 위치로 선정하고, 분리된 블록들에 대해서 다시 분리될 블록의 조건에 맞는지를 확인한다.



(a) 유형1 (a) type1 (b) 유형2 (b) type2 (c) 유형3 (c) type3

(d) 유형4 (d) type4 (e) 유형5 (e) type5 (f) 유형6 (f) type6

(그림 2) 6가지 접촉 유형
(Fig. 2) Six touching types.

2.3 분리된 문자 블록의 수평 합성

분리된 문자 블록들은 단일 문자를 구성하는 경우도 있지만 대부분의 문자는 수평적인 결합을 필요로 하게 된다. 2. 1. 3절에서 연결 화소 블록의 합성시 제외된 수평적 결합을 하여야 하나의 완전한 문자로 추출된다. 최빈 블록폭을 하나의 문자 폭으로 보고 두 블록의 크기가 최빈 블록폭과 비슷하면 하나의 블록으로 합성해서 개별 문자로 추출한다.

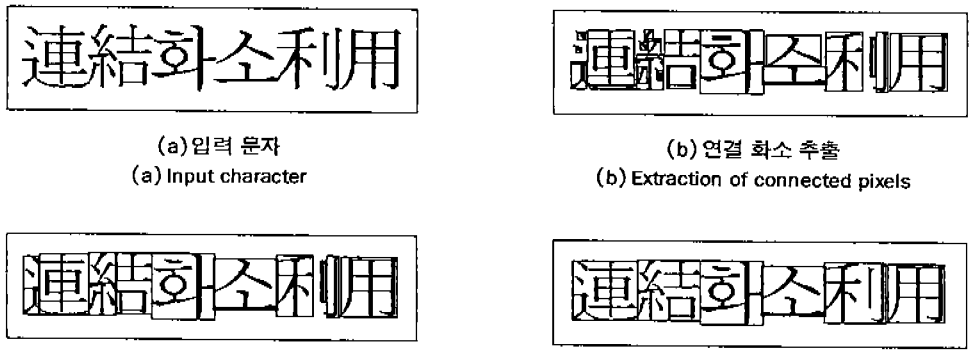
그림 3에서는 문자의 자간 간격 -15의 혼합 문자를 그림 3(b) 처럼 연결 화소 추출로 얻은 블록을 그림 3(c)와 같이 수직으로 합성 하였고, 그림 3(d)와 같이 수평 블록의 합성과 함께 개별 문자의 결과를 얻을 수 있다.

3. 특징 추출과 문자 인식[13]

3.1 최대 블록화 방법을 이용한 문자의 특징 추출

문자획의 특징 구조들을 정확하게 추출하기 위해서 최대 블록화 방법을 사용한다. 최대 블록 생성 방법은 획의 시작점에서 부터 블록을 점차 키워 나아가는 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 9(b)에서는 입력 영상을 최대블록화 한 중간 과정을 볼 수 있다.

최대블록화 하여 생성된 블록은 수직,수평획을 추출하는데 효과적 이다. 그리고 정보로 사용하게되는 블록들은 실제 원영상에서 볼 수 있는 영상과는 다르다. 원영상인 경우에는 문자의 외부 꼭지점 밖에는 없다. 이러한 외부 점들은 원영상의 외곽선에 포함되는 점들로서 이러한 점을 이용하여 최대블록화를 하면 원래의 외곽에서 볼 수 없었던 안쪽 블록의 정보를 이용 할 수 있다. 획의 내부 구조까지 블록들이 구성되어 있기 때문에 더 많은 특징 정보를 이용 할 수



(a) 입력 문자 (a) Input character

(b) 연결 화소 추출 (b) Extraction of connected pixels

(d) 개별 문자 추출 (d) Extraction of an idividual character

(c) 세로축 블록의 합성 (c) Merge of the vertical exes block

(그림 3) 개별 문자 추출 순서
(Fig. 3) The sequence for the extraction of an idividual character.

있다.

또한 원영상을 양자화하면서 오차가 발생한다는 것과 이러한 오차는 일정한 범위의 편차와 오차 범위를 가진다는 점, 그리고 이러한 정보들로 원래의 영상을 근사화 시켜서 재생이 가능하다는 점들을 이용했다. 이렇게 생성된 최대 블록의 정보는 각 블록의 꼭지점, 블록의 겹친 정도와 블록의 크기 및 진행 방향의 특징을 가지고 직선 블록과 사선 블록으로 구분하여 문자의 골격선 및 획의 특징을 찾아 인식에 사용된다.

3.2 획의 방향성 결정

3.2.1 수직, 수평 성분 블록의 추출

생성된 최대 블록들 중에서 수직, 수평 성분이 될 가능성이 있는 블록들을 합성 또는 제거함으로써 잡음을 없애고 간략화 하여 단순화된 블록들로 만든다. 이때 수직, 수평 성분의 블록과 그외의 블록을 구분하는 것은 수직, 수평획 외의 사선 성분 획을 정확하게 구분하기 위해서 이다. 이러한 블록의 구분 방법은 블록의 가로, 세로의 비율에 의하여 쉽게 알 수 있다.

수직, 수평성분의 처리에서 제거 또는 합성될 수 있는 블록들의 대상으로서, 큰 블록과 75%이상 겹쳐진 블록은 큰블록에 합성시킨다. 그리고 가로나 세로의 폭이 1화소(pixel)인 작은 블록은 잡음 블록으로 판단하여 제거한다. 그외의 블록들은 3.2.2 과정을 거치게되어 사선으로 구분된다. 이러한 처리 만으로도 많은 잡음이 제거되고 블록의 수가 줄어들게 되어, 1차적인 잡음 제거와 획추출을 위한 작업이라 할 수 있다. 그림 9(c)에서는 직선획 블록을 추출한 영상이다.

3.2.2 사선성분 블록의 추출

3.1절 과정을 거치면서 간단해진 수직, 수평 성분의 블록과 합성에서 제외된 나머지 블록들을 합치면 최대블록화한 최초의 블록들 보다 잡음이 제거된 필요한 블록들만 남는다. 이 블록들로 부터 사선 성분을 분리하는 데는 여러 방법이 있을 수 있으나, 실제의 경우에는 여러 제약에 의해서 사용 할 수 있는 방법이 한정된다.

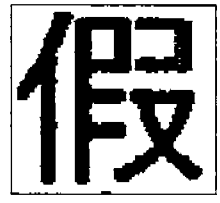
우선 블록이 사선 성분인지 판단 할 수 있는 방법은 그 형태가 수직, 수평의 획들과는 다른 형태를 가지고 있기 때문에, 사선 성분은 대부분의 경우에 정방형에 가까운 형태를 하고 있으며 여러개의 블록들로 연결되어 있다. 그러므로 가로, 세로의 비율과 블록들의 연결된 형태로써 사선임을 판단 할 수가 있다.

3.2.3 수직, 수평 성분과 접합된 사선 성분의 보완법

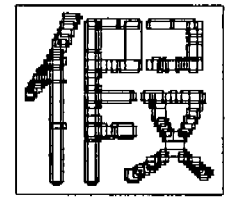
수직, 수평획과 사선획이 교차하거나 인접되어 있는 경우에 블록의 크기를 제한하여 분리하면 사선획의 일부가 수직, 수평획의 성분으로 포함되어 사선획이 불완전해 지거나 끊어지는 경우가 있다. 이러한 원인을 분석한 결과 수직, 수평획과 사선획이 교차되면 사선획 성분의 일부가 수직, 수평 성분획에 포함되기 때문으로 나타났다.

다음 그림 4(d)와 같이 직선과 사선의 블록의 분할시 사선획이 연결 되지 않는 것을 볼 수 있는데, 이것은 단순한 사선 추출 방법을 사용하여 실제 데이터를 입력 후 획추출을 한 결과 그림 4(e)의 1, 2의 경우와 같이 직선과 사선이 교차되는 부분에서 직선 블록의 분리 때문에 사선획의 추출이 잘 되지 않는 경우이다.

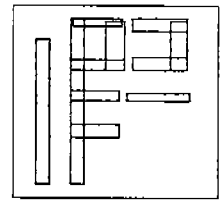
이러한 문제를 해결하는 방법으로서, 그림 4(g)와 같이 획 추출시 단절되는 획의 보완법으로 다음과 같은 알고리즘을 사용하여 획의 단절을 막을 수 있었다.



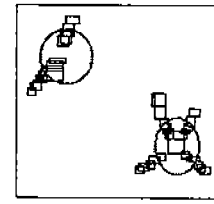
(a) 입력data
(a) Input data



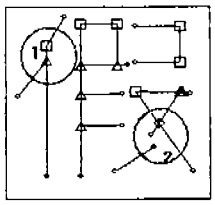
(b) 최대블록화
(b) Maximum Block Method



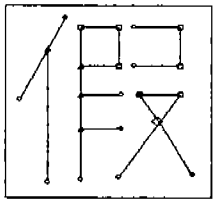
(c) 직선 추출
(c) Straight line extraction



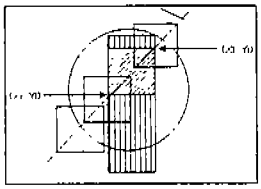
(d) 사선 추출
(d) Oblique line extraction



(e) 골격선 추출결과
(e) Skeleton extraction



(f) 사선 보완법의 적용결과
(f) An application result for the oblique interpolation



(g) 사선연결 알고리즘
(g) An algorithm for the connection of oblique line

(그림 4) 교차영역에서 誤추출된 stroke 1, 2의 해결
(Fig. 4) Erronously extracted strokes 1 and 2 in the crossed region.

직선에서 사선의 점과 교차되는 (Xi, Yi) 점과(Xi, Yi) 점을 추출하여 블록을 생성 후 직선의 블록은 그대로 직선 추출을 하는 방법으로서 우선 첫번째 블록에서 임의의 사선을 만들어 생성된 블록을 연결 블록으로 하고, 직선 블록은 그대로 두어 직선, 사선을 구한다. 이와 같은 방법을 사용하여 획의 연결을 강하게 유지하며, 블록이 작아져서 잡음으로 나타날 수 도 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 접합 부분에서 향상된 획추출 결과를 얻을 수 있었다.

3.2.4 획의 골격선 추출

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

우선, 직선의 경우에는 블록의 길이방향 즉, 수평, 수직획임을 찾아서 획의 양쪽끝 중심점을 연결하여 골격선으로 만든다. 사선의 경우에는 사선임을 판단하는 가상 직선에 연결된 시작과 끝 블록의 대각점을 연결하여 생성되어진다. 이때 만들어진 골격선들은 획들의 연결 부위에서 항상 교차점으로 되는데, 특징점 추출시에 교차된 끝점을 수정하여 정확한 획의 특징을 구한다.

3.3 획의 특징 추출

정규화된 블록들의 방향 성분이 결정되면 인식을 위하여 획들의 연관 관계를 특징 코드로 저장하여야 한다. 이러한 특징 코드 생성 규칙들의 순서는 우선 획의 4방향 특징을 찾는 것이다. 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, 위치 특징으로써 연결된 획들의 무게중심을 찾아 구한다.

(1) 방향 특징
- (수평 =〉 "−", 좌사 =〉 "/", 수직 =〉 "|", 우사 =〉 "\")와 같이 4방향 특징을 사용

(2) 획의 연관 관계 특징
- 단점 :(○)==〉 "e"
- 굴곡점 :(□)==〉 "t"
- 분기점 :(△)==〉 "b"
- 교차점 :(◇)==〉 "c"

(3) 획 추출시 순서 특징

추출 순서는 처음 찾아진 획에서 연결된 획의 좌에서 우로, 위에서 아래로의 순서를 정한다.

(단, 처음 찾아진 획에서 가장 가까운 획 부터 찾기 시작한다.)

(4) 획의 길이 정보 특징

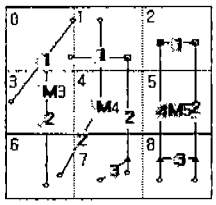
길이 특징은 추출된 문자를 X, Y축의 일반적인 문자폭을 최대로 하여
- 제일 짧으면(1/3 이하) =〉 "1"
- 중간 길이면(2/3 이하) =〉 "2"
- 최대 길이면(2/3 이상) =〉 "3" 등으로 길이 구분을 하였다.

(5) 위치 특징
- 연결된 각 획 마다 중심점(1, 2, 3...)을 구한다.

- 구해진 중심점들 중 M값을 찾는다.
- 3X3의 각 블록 위치에 기억하여 코드로 저장한다.(그림 5(a))

그림 5(b)에서는 본 절에서 제안한 획의 특징을 인식에 필요한 코드로 생성한 결과이다. 코드 생성 예를 보면 "伽"에서 "亻"자는 "/"획과 "|" 획으로 이루어져 있는데 여기에는 획의 길이 특징, 획의 연관 관계, 추출 순서, 위치 특징 모두를 코드에 포함하고 있다.

특징점은 한획이 여러개의 특징점을 가질 수 있으며, 그 획을 중심으로 정의한 것이다. 특징점들은 획의 시작점에서 가까운 순서로 배열하고, 특징점들이 너무 가까이 인접해 있으면 하나의 특징으로 간주한다. 그리고 획의 연관 관계 특징점 중 e(단점)와t(굴곡점)는 획의 양 끝에 서만 가능하다.



(a) 위치 결정(M)
(a) The location determination(M)

```
/ 2ebe
| 2be >3

| 2ect
-1ect
| 2tb
/ 1be
/ 2te >4

-1tt
| 3tbe
-1bb
| 3tbe >5
```

(b) 특징 코드
(b) Feature code

(그림 5) 연결된 각획의 무게 중심점과 특징 코드
(Fig. 5) The gravity center of each connectedstroke and feature code.

3.4 인식 트리 구성

생성된 코드를 가지고 트리를 구성하는데, 이 트리는 N개의 가지(branch)를 가질수 있는 트리로서 가지의 개수는 가변될 수 있다. 그림 6은 트리의 구성을 표현한 것인데 하나의 노드내에 특징 코드가 들어오는대로 br0, br1, br2,...brN 의 순서로 누적되어 효율성을 극대화 시킨 것이다. 또한 링크에서 다음 오는 트리의 인덱스를 가르키는 방식으로 구성된다. 또한 이 트리에서 개개의 노드는 하나의 R-code (Return-code)를 가지는데 이 코드는 노드에서 획열의 끝이 발견되면 인식이 끝난 것으로 보고 결과를 출력한다.

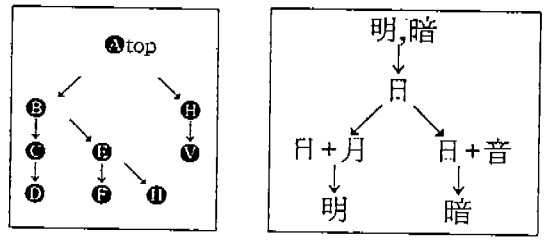
만일 코드가 비어있으면 새로운 문자이므로 학습에 들어간다. 가지의 수는 획의 다양성에 의존 하여 결정된다. 만약 획의 배열이 다양한 조합으로 이루어지면 이 수는 증가한다.

획의 열로 부터 트리의 노드를 찾으려면 이 획열을 하나의 숫자열로 바꾸어야 하는데 여기서는 테이블을 하나 만들고 새로운 획 코드가 들어오면 이 테이블에 등록하고 구성한다.

그림 7은 "明"자와 "暗"의 문자가 입력되면서 트리가 분류되는 모습을 예로써 표현한 것인데, 우선 "日"자의 트리를 사용후 트리가 분류 되는 것을 볼 수 있다. 이러한 방법을 사용함으로써 트리의 효율성을 높일 수 있었다.

Main	Index	1	2	·····	N
(인식결과)	Code	br0	br1	·····	brN
R-code	Link	·	·	·	·

(그림 6) 트리 노드 구조
(Fig. 6) Structure of a tree node



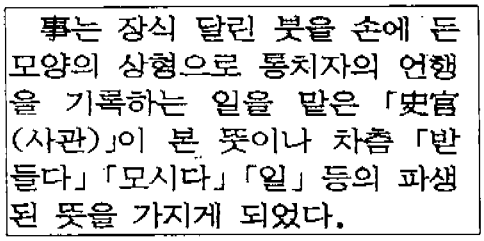
(그림 7) 트리 분류의 예
(Fig. 7) Branching of a tree

4. 실험결과

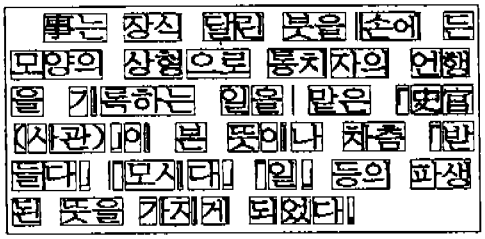
본 논문에서는 오프라인 한자 뿐만이 아닌 모든 문자를 대상으로 개별 문자를 추출하여 인식하는 알고리즘을 제안 하였다. 개별 문자를 추출하기 위하여 문서상에 6가지 종류[7]의 접촉 형태가 혼용된 문서를

사용하여 개별 문자를 추출 하였고, 대상 문자로는 신문 및 프린터 인쇄물과 중학교, 고등학교 한자 교과서 등에서 얻어진 한자와 한글, 영, 숫자를 대상으로 하였다.

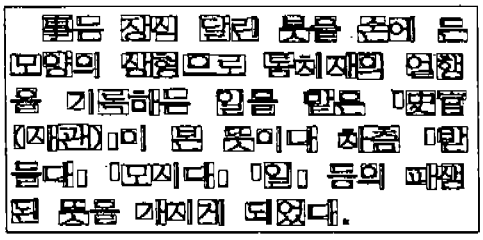
문자 추출 방법은 본문에서 처럼 우선 문자행을 추출하였다. 그리고 문자열 투영에 의한 연결 화소 정보를 이용하여 모든 연결 문자를 추출 하였으며, 문자 구조를 고려하여 상하, 좌우 블록을 하나의 개별 문자가 되도록 합성 하였다. 최종적으로 최빈 블록폭을 가지고 접촉 문자등을 분리하여 개별 문자 추출할 수 있었다.


事는 장식 달린 붓을 손에 든 모양의 상형으로 통치자의 언행을 기록하는 일을 맡은 「史官(사관)」이 본 뜻이나 차츰 「받들다」「모시다」「일」 등의 파생된 뜻을 가지게 되었다.

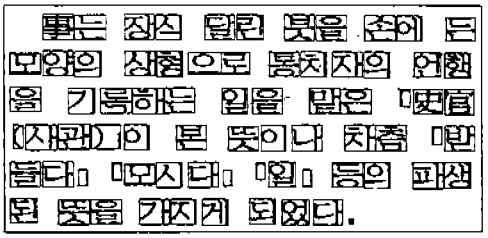
(a) 입력 데이타
(a) Input data



(b) 투영법을 사용하여 추출된 문자 블록
(b) The character blocks extracted using the projection method



(c) 연결 화소 블록
(c) The connected pixel blocks



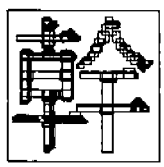
(d) 본 방법을 사용하여 추출된 문자 블록 결과
(d) The result of extracted characters using the MBM

(그림 8) 입력된 신문 영상에 대해 기존의 방법과 본 방법을 비교한 결과
(Fig. 8) The result of the comparison between the existing method and the MBM on the images of a newspaper.

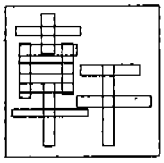
그림 8에서 기존의 방법과 본 방법의 개별 문자 추출 차이점을 볼 수 있다. 본 방법의 유효성을 확인하기 위하여 그림 8(a)에서는 입력 문서, 그림 8(b)에서는 투영법만 사용한 결과, 그림 8(c)에서는 연결 화소만을 이용한 결과를 보인다. 그림 8(d)는 본 방법을 사용하여 추출된 문자 블록들이며, 그림 8(b), (c)를 모두 충족시켜주는 결과이다.

추출된 개별 문자가 입력되면 우선 최대블록화 방법을 이용하여 각 문자의 획을 블록으로 구성한다. 본 논문에서는 타 논문과 비교하여 다른점으로서 직선 획을 정확히 추출하면서 잡음 성분도 제거할 수 있다. 또한, 그림 9에서와 같이 사선을 정확히 추출하기 위해서는 직선과 사선을 분리시킨 후 잡음 제거와 정규화된 블록의 합성 과정을 통하여 골격선을 추출하였다. 추출된 골격선의 특징을 단점, 굴곡점, 분기점, 교차점 등을 이용하여 인식에 필요한 구조 정보를 추출하는 전 과정을 순서대로 보인다. 본 논문에서는 주로 한자에 대한 특징을 추출하는 방법이지만 한글 및 영문자를 혼용하여 실험 한 결과 혼합 문서

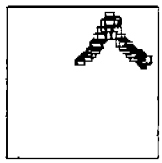
(a) 입력 문자
(a) Input character



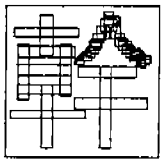
(b) 최대블록화
(b) Maximum block generation



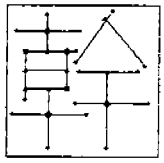
(c) 직선획 분리
(c) Straight line segregation



(d) 사선획 분리
(d) Oblique line block segregation

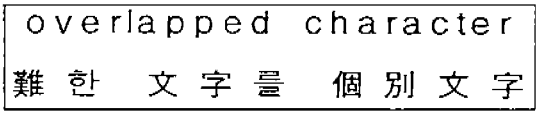


(e) 정규화된 블록
(e) Remerged block

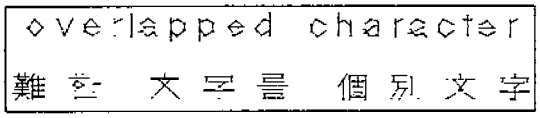


(f) 특징 추출
(f) Feature extraction

(그림 9) MBM방법을 이용한 특징 추출의 순서
(Fig. 9) The sequence for the feature extraction using the MBM.



(a) 혼합 문자의 입력 영상
(a) Input image of the mixed characters

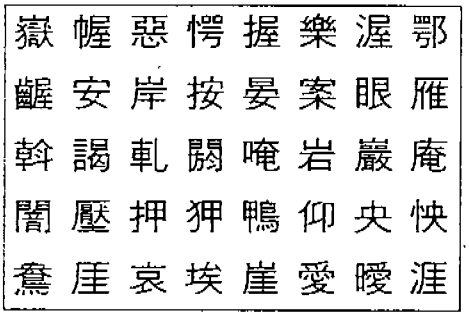


(b) 골격 및 특징 추출
(b) Skeleton and feature extraction

(그림 10) 여러 문자의 특징점 추출
(Fig. 10) The feature extractions of different characters.

에서도 정확한 특징 추출 및 인식을 할 수 있는 방법임을 그림 10에서와 같이 실험을 통해 입증 하였다.

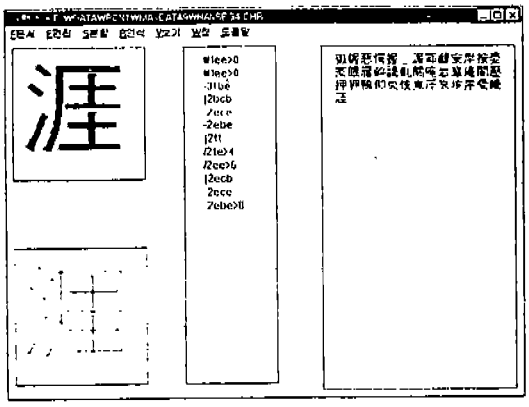
그림 11에서는 실험 데이터 중 무작위로 선정한 문자 영상을 입력하여 개별 문자를 추출한 결과와 인식에 필요한 특징을 추출하여 인식한 결과로서 그림 11(d)에서는 "涯"자를 마지막으로 입력하여 골격선 추출과 획의 구조적 특징을 추출하여 중앙의 특징 코드를 얻을 수 있다. 오른쪽 그림과 같이 인식 결과 중 "樂"자가 미인식을 하였으나 반복 학습 후 올바른 결과를 얻을 수 있었다.



(a) 입력 영상
(a) Input characters

(b) 개별 문자 추출
(b) The extracted individual characters

(c) 특징 추출
(c) The feature extraction



(d) 특징 코드와 인식 결과
(d) The result of the feature code and recognition

(그림 11) 임의의 한자 문서를 입력 후 인식된 결과
(Fig. 11) The recognition of the input of arbitrary Chinese characters.

5. 결 론

문자 인식을 위한 연구 결과가 국내외적으로 많이 발표되었고, 또한 이러한 연구의 결과로 상업용 인식기가 시판되고 있다. 그러나 인식에 앞서 전처리에서의 문자 추출의 결과가 인식율을 판가름 할 수 있는 중요한 부분임을 알 수 있고, 한자의 인식율이 저조한 것으로 되어있다.

따라서 본 논문에서는 개별 문자 추출 방법으로서 기존의 투영법과 외곽선 추적하는 방법을 모두 만족시킨 연결 화소를 이용한 투영 방법을 제안하였다. 개별 문자로 추출된 문자를 인식하는 알고리즘으로는 최대 블록화 방법을 사용하였다. 본 방법은 획의 직선 성분과 사선 성분을 분리하였고, 잡음 블록은 별도로 처리하여 제거 및 합성의 조건에 포함시켜 처리함으로써 인식에 필요한 골격선 및 특징 코드를 추출 할 수 있었다. 특히 한자의 경우는 대부분 직선과 사선의 복잡한 구조로 조합되어져 있고, 한글과 달리 문자의 구조상 교차점(cross point)이 형성되어 있는데, 이러한 점들이 본 알고리즘에 적합함을 실험을 통하여 확인하였다.

한자의 학습 데이터는 인쇄체 문자의 완성형 한자 4,888자를 대상으로 명조체, 고딕체 등을 학습 하였고, 기타 여러 문자의 학습 데이터도 가지고 실험을 하였다.

본 논문에서는 전처리 과정으로서 본 알고리즘을 이용한 개별 문자 추출로 향상된 인식율을 얻을 수 있었다. 현대에서는 모든 문자가 혼용되어 사용되고 있기 때문에 문자 분할 및 혼용 문자를 동시에 인식할 수 있는 방법을 개발하기 위하여 연구 중에 있다.

참 고 문 헌

[1] S. Tsujimoto and H. Asada, "Resolving Ambiguity in Segmenting Touching Characters," Proceeding 1st, International Conference on Document Analysis and Recognition, pp. 701-709, 1991.

[2] S. Liang, M. Ahmadi, M. Shridhar, "Segmentation of Characters in Printed Document Recognition," Proceeding 2nd, International Conference on Document Analysis and Recognition, pp. 569-572, 1992.

[3] Y. Lu, "On the Segmentation of Touching Characters," Proceeding 2nd, International Conference on Document Analysis and Recognition, pp. 440-572, 1992.

[4] D. Wang, S. N. Srihara, "Classfication of Newspaper Image Blocks Using Texture Analysis," Computer Vision, Graphics, and Image Processing 47, pp. 327-352, 1989.

[5] 이인동, 권오석, 김태균, "블록영상의 추출 알고리즘," 한국정보과학회 논문지, Vol. 18, No. 2, pp. 218-226, 1991.

[6] 장명욱, 천대녕, 양현승, "연결화소를 이용한 문서 영상의 분할 및 인식," 한국정보과학회논문지, Vol. 20, No. 12, pp. 1741-1751, 1993.

[7] 김의정, 김태균 "인쇄체 문서 인식을 위한 문자 추출에 관한 연구," 제2회 문자인식 워크샵 발표 논문집, pp. 171-179, 1994.

[8] 이균하, "한글 문자 인식에 관한연구," 정보과학회지, Vol 9, No 1, pp. 45-53, 1992.

[9] 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.

[10] N. J. Naccache and R. Shinghal, "An Investigation into the Skeletonization Approach of Hilditch," Pattern Recognition, Vol. 17, No. 3, pp. 279-284, 1984.

[11] S. Kahan, T. Pavilidis 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.

[12] C. C. Han and K. C. Fan, "Skeleton Generation of Engineering Drawings Via Contour Matching," Pattern Recognition, Vol. 27, pp. 261-275, 1994.

[13] 김의정, 김태균 "최대블록화 방법을 이용한 문자획 특징 추출에 관한 연구," 한국정보처리학회 논문지, Vol. 4, No. 3, pp. 1141-1151, 1997.

김 의 정

1993년 충남대학교 대학원 컴퓨터공학과(공학석사)
1997년 충남대학교 대학원 컴퓨터공학과(공학박사)
1993년~현재 대전산업대학교 전자공학과 시간강사
1997년~현재 시스템공학연구소 컴퓨터비젼연구실 연구원
관심분야: 문자인식, 영상처리, 멀티미디어

김 태 균

1971년 서울대학교 공업교육학과(학사)
1980년 일본동경공업대학 대학원 물리정보공학과(공학석사)
1984년 일본동경공업대학 대학원 물리정보공학과(공학박사)
1974년~현재 충남대학교 컴퓨터공학과 교수
관심분야: 문자인식, 영상처리, 멀티미디어