

# 필기 한글 패턴을 위한 두 단계 모양 분해 알고리즘

박정선\* 오일석  
전북대학교 컴퓨터과학과

## Two-pass Shape Decomposition Algorithm for Handwritten Hangul Patterns

Jeoung-Sun Park and Il-Seok Oh

Department of Computer Science, Chonbuk National University, Korea

요약

필기 한글 문자 인식을 위해서는 패턴을 구성하는 획 성분을 분석하는 작업이 필수적이다. 기존 인식 방법들은 세선화와 직선 근사에 기반한 방법을 사용하였다. 하지만 세선화는 필기 패턴을 크게 왜곡하는 단점을 안고 있기 때문에 새로운 방법론의 필요성이 대두되고 있다. 본 논문에서는 필기 한글 패턴의 영역-기반 모양 분해 알고리즘을 제안한다. 외곽선 분석을 이용한 기존의 한 단계 알고리즘의 한계를 지적하고, 이 한계를 극복할 수 있는 두 단계 알고리즘을 기술한다. 첫 번째 단계에서는 우세점을 찾아 B점점과 T점점을 탐지한다. 두 번째 단계에서는 볼록 껍(convex hull) 연산을 적용하여 미 분할된 부품에 대해 두 번째 분할 작업을 수행한다. PE92데이터베이스에 대해 실험한 결과는 세선화 방법보다 우수함을 보인다.

### 1. 서론

필기 한글 패턴을 분석하고 인식하기 위해서는 접촉 글자 분할, 자모 분할, 획 분할 등의 작업이 매우 중요하다.[1] 기존 논문은 이들 작업을 위하여 고전적인 방법인 세선화와 직선 근사에 의존한다. 하지만 세선화에 기반한 알고리즘을 필기체 문자 패턴에 적용하였을 경우 발생하는 정보 손실 및 패턴 왜곡은 문자 인식을 매우 어렵게 만드는 것은 확실하다[10]. 세선화 방법의 대안으로서 모양 분해 알고리즘은 대부분 임의의 2차원 패턴에 대해 동작하도록 개발되었다 [2,3,5,6,7,8,9]. 이러한 알고리즘의 장점은 모양은 분해하되, 분해된 요소들을 다시 조합하면 원래 모양을 그대로 재생할 수 있다는 것이다. 모양 분해는 두 가지 다른 응용에 대하여 생각할 수 있다. 한글 단어를 문자 단위로 분할하는 경우의 목적은, 다른 글자에 속하는 획이 같은 요소에 포함되지 않도록 하는 최소 개수의 요소 집합으로 분해하는 것이다. 한 문자를 세 개 자모(초성, 중성, 종성)로 분할하는 응용에서는, 다른 자모에 속하는 획이 같은 요소에 포함되지 않도록 하는 최소 개수의 요소 집합으로 분해하는 것이다. 두 경우 모두 최선은, 각 문자(자모)가 하나의 요소가 되도록 분해하는 것인데, 이것이 불가능하기 때문에 보다 작은 요소 집합으로 분해 한 후, 여러 요소를 조합하여 한 개의 자음이나 모음을 만드는 방식으로 문자 분할을 수행해야 한다.

본 논문은 한글 단어 분할과 자소 분할 등의 작업에 응용할 수 있는 두 단계 모양 분해 알고리즘을 제안한다. 두 개의 획이 T 형태로 만나는 T 점점, 획이 굵은 B점점을 우세점(dominant point)에 의존하여 분할하는 방법은 이전에 제안되었다[2,7]. 대용량 한글 패턴에 대한 면밀한 실험결과, 우세점에 의존하는 방법만으로는 모든 패턴에 대한 모양 분해를 할 수 없다는 결론을 얻었다. 그 이유는 첫 번째 원하는 위치에 있는 완벽한 우세점을 찾는다는 것은 불가능하고 두 번째 모든 패턴이 T점점과 B점점의 모양으로 이루어지지 않는다는 것이다. 이를 본 논문에서는 첫 번째 단계라 한다. 이런 이유로 이를 보완한 두 단계 모양분해 알고리즘을 제안한다. 본 논문에서 제안한 알고리즘은 새로운 패턴인 획이 Y 형태로 만나는 Y점점을 탐지하며, 첫 번째 단계의 결과의 모양 특징을 찾아 분해하는 알고리즘이다.

본 논문에서는 이러한 모양을 두 부분(첫 번째 단계, 두 번째 단계)로 나누어 특징을 탐지한다. 첫 번째 단계에서는 패턴 외곽선으로부터 우세점을 탐지하고 그 우세점에 의존하여 T점점과 B점점을 탐지[2,7]하고, 두 번째 단계에서는 우세점에 의존하지 않고 첫 번째 단계에서 찾지 못한 T점점과 B점점을 원 이미지 모양을 가지고 다시 한번 탐지하여 분할하며, Y점점을 탐지한다. 두 번째 단계에서는 패턴모양을 탐지하기 위하여 새로운 방법인 볼록 껍(convex hull) 알고리즘을 적용하여 분할 경로를 탐지하였다.

본 논문에서 제안한 알고리즘의 실험결과 B점점과 T점점에 대해서 놓치지 않고 잘 분할하였다. 하지만 한글 구조에 대한 지식 없이 분할하기 때문에 자모가 붙어 하나의 획처럼 인식하는 경우가 발생하였다. 이런 문제점을 해결하기 위하여 휴리스틱 정보가 들어가야만 한다. 본 논문에서 제안한 알고리즘의 결과를 분석한 결과 세선화 방법보다 우수함을 알 수 있었다.

### 2. 자모를 이루고 있는 특성

한글 모양은 자음 14개(ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㆁ ㆅ)와 모음 10개(ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ) 그리고 그들의 합성으로 이루어져 있다. 한글의 자모를 분석하여 보면 하나의 획이 굵은부분(B점점)과 두 획이 만나는 부분(T점점)의 조합으로 이루어짐을 볼 수 있다.

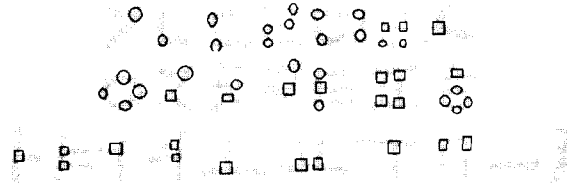


그림 1. 자모 모양의 특성 (O: B점점 □: T점점)

그림 2. 자모가 붙어 있는 패턴

그림 1에서 보듯이 기본 자모 24자는 B점점과 T점점만으로 이루어져 있다. 자모 단위가 붙은 패턴에서도 대부분은 B점점과 T점점으로 이루어진다. 하지만 그림 2에서와 같이 새로운 Y점점이 생성됨을 볼 수 있다. 이러한 한글 패턴을 분석해 보았을 때 한글을 이루고 있는 패턴을 3가지로 축약할 수 있다.

#### 한글을 이루고 있는 3가지 패턴

- 1) T점점: 두 개의 획이 T형태로 만남 (오목점과 오목점의 쌍)
- 2) B점점: 한 획이 구부러진 부분(오목점과 볼록점의 쌍)
- 3) Y점점: 두 개의 획이 Y 형태로 만남(세개의 오목점의 쌍)

T점점에서 변형된 패턴

한글을 자모 단위로 분할하기 위해서는 한글 특성상 2가지 패턴(T점점과 B점점)만으로 해결 할 수 있다. 그러나 필기체 한글 문자에서는 획의 기울기가 인쇄체 문자처럼 곧지 않고 굴곡이 많이 발생한다. 이러한 문제점을 보완 할 수 있는 Y점점 탐지가 필요하다. T점점이 되기 위해서는 반드시 오목 점(concave point)과 오목 점의 쌍으로 이루어져야 한다. B점점이 되기 위해서는 오목점과 마주보는 외곽선 상에 볼록점(convex point)의 한 쌍으로 이루어져야 한다. Y점점은 필기체 한글 문자에 자주 발생하는 패턴으로 T점점의 형태는 비슷하나 T점점이 되기 위한 조건이 만족하지 않는 모양을 가지는 형태이며 세 개의 오목점의 쌍으로 이루어진다. 그리고 한글에서 많이 나타나는 모양(O, ㅁ)은 B점점 모양의 합성이라 생각할 수 있다. 한글에서는 기본적으로 영어와 한자와 같이 두 획이 접침모양(X, f, t, 大, 土, etc)으로 이루어지지 않는다. 그러므로 제시한 3가지 패턴으로 한글을 분해하는 것이 가능하다. 하지만, 필기체 한글에서는 접침 모양(ㅃ, ㅆ, ㅈ, etc)으로 필기한 문자가 일반적인 사람의 필기 습관에 따라 여러 형태로 발생할 수 있다. 이곳에서 연결부품(joint part)이라는 개념을 도입한다. 연결부품이란 접침모양으로 이루어지는 패턴으로 T점점이나 Y점점에서 해결되는 분할라인으로 둘러싸인 영역을 말한다. 이 연결 영역은 두 부분에서 나타난다. 첫 번째 단계의 T점점에서 주로 'ㅁ'이 겹쳐 쓰여지는 부분('x'형태)이나 중성 'ㅏ' 'ㅑ'와 함께

\*이 연구는 정보통신부 대학기초과제의 지원에 의한 것입니다.

초성이 집친 부분(+'형태)등에서 나타난다. 이것은 T점점 조건에서 하나의 우세점이 다른 여러 개의 우세 점과 쌍이 되는 것을 허락하게 한다. 이 개념을 도입함으로써 가로로 분할해야 할지 세로로 분할해야 할지 하는 애매성을 없앨 수 있다. 두 번째 단계의 Y점점에서 나타날 수 있다. 이 곳도 역시 어느 한 방향으로 분할하기에는 애매한 부분이다.

3. 두 단계 모양 분해 알고리즘 소개

본 논문에서 제안한 알고리즘은 2장에서 설명한 바와 같이 두 단계에 걸쳐 한글을 모양 분해하였다. 두 단계는 아래와 같으며, 각 단계의 자세한 동작에 대한 설명이 뒤따른다.

한글 패턴을 위한 모양 분해

첫번째 단계:

- 절차 1: 잡음 제거 및 외곽선 추출
- 절차 2: 외곽선에서의 우세점탐지 및 불록점/오목점 구분
- 절차 3: 오목점을 중심으로 영역 설정(3-4거리변환 알고리즘 적용)
- 절차 4: T점점 추출 (T점점 알고리즘 적용)
- 절차 5: B점점 추출 (B점점 알고리즘 적용)

두번째 단계:

- 절차 1: 첫 번째 단계에서 나온 패턴에 대해서 고유값 설정
- 절차 2: 외곽선 추출 및 불록 할을 이루고 있는 점들 추출
- 절차 3: 홀에 대한 영역 검사
- 절차 4: 홀 모양 탐지

3.1 첫번째 단계 알고리즘

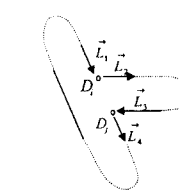
절차 1에서는 원본 영상(그림3(a))에서 외곽선을 따라 시계방향으로 체인 코드를 구한다. 또한 이 체인 코드의 방향값을 조사하여 미세한 돌출 부분을 제거하는 전처리 과정을 거친다.(그림3(b)) 절차 2에서는 Teh-Chin 알고리즘을 사용하여 우세점을 추출한다[4]. 이 알고리즘은 입력 매개 변수가 필요 없고 입력 영상의 크기와 회전에도 민감하여 강건한 탐지가 가능하지만 불필요한 우세점을 생성하기 때문에 거기에 맞는 제거 방법이 필요하다. 그림 3(d)는 전처리 과정이 끝나고 남은 우세점들을 보여준다. 이렇게 제거하고 남은 우세점들을 오목점(이하 CDP)과 불록점(이하 VDP)으로 분류한다.(그림3(d)) 이제 아래 연산을 적용하는데, 이 연산의 목적은 하나의 CDP를 중심으로 주위의 작은 영역을 생성하는데 있으며, 이 영역은 우리가 분할하려고 하는 인접한 CDP를 찾는 데 효과적으로 이용 될 수 있다[9]. 이 연산을 위하여 자신 CDP(D<sub>i</sub>)점과 바로 이웃한 몇 개의 외곽선상의 점(⋯,D<sub>i-2</sub>,D<sub>i-1</sub>,D<sub>i+1</sub>,D<sub>i+2</sub>⋯)을 포함한 점들을 concave interval이라고 정의한다. concave interval의 크기는 1/(D<sub>i</sub>),3(p<sub>i-1</sub>,D<sub>i</sub>,p<sub>i+1</sub>),5((D<sub>i-2</sub>,D<sub>i-1</sub>,D<sub>i</sub>,p<sub>i+1</sub>,D<sub>i+2</sub>) 또는 그 이상의 크기를 가질 수 있다. 이 하나의 CDP 중심으로 이루어진 concave interval 안에 있는 점들을 유일한 값으로 설정하고, 설정된 값을 (3,4)-거리변환 알고리즘을 수행하여 패턴 내부에 있는 화소값을 같은 값으로 설정한다(그림 3(e)). 이것은 각기 다른 CDP점에서 생성된 영역이 만나게 되어 큰 영역이 형성되는데, 이 CDP들은 우리가 분할하려고 하는 쌍들의 후보점들이 된다. 이 concave interval을 도입한 이유는 한글 필기체 패턴에서는 획의 두께와 바르기가 다르기 때문이다. 이는 CDP의 영역을 확장함으로써 복잡한 패턴에서도 적용 될 수 있게 한다[9].



(a)원본 영상 (b)외곽선 추출 (c)Teh-Chin 알고리즘 적용 (d)불필요한 우세점제거(검은점: 오목점 회색점: 불록점) (e)CDP 영역 설정(concave interval = 3)

그림 3. 전처리과정

T점점 알고리즘을 적용하여 T점점을 추출한다. 절차 3에서 추출된 CDP들중 두점을 선택할 수 있는 모든 경우의 수를 고려하여 각각에 대하여 그림 4의 조건이 맞는지를 검사한다.

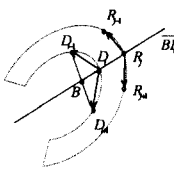


- 1) 두점의 근접성 : Distance(D<sub>1</sub>,D<sub>2</sub>) < ε<sub>2</sub>
- 2) 동일 직선상에 위치 : Abs(θ( $\vec{L}_1$ )-θ( $\vec{L}_2$ )) ≤ ε<sub>1</sub>
- Distance(L<sub>1</sub>, D<sub>1</sub>) ≤ ε<sub>2</sub> and Distance(L<sub>2</sub>, D<sub>2</sub>) ≤ ε<sub>2</sub>
- 3) 반평행성 : 180°-ε<sub>3</sub> ≤ Abs(θ( $\vec{L}_2$ )-θ( $\vec{L}_3$ )) ≤ 180°+ε<sub>3</sub>

그림 4. T점점이기 위한 조건

여기에서는 파라미터 값을 ε<sub>1</sub> = 35°, ε<sub>2</sub> = 10, ε<sub>3</sub> = 35°로 설정하였다. 절차 3에서 추출된 CDP들이 네 개이고 각각의 경우가 위의 조건에 모두 만족한다면 이 부분은 두 획이 교차하는 모양이라 볼 수 있으며, 네 개의 점이 한 쌍이 되는 연결영역이 된다. 절차 5에서는 B점점 알고리즘을 적용하여 B점점을 추출한다. 절차 4까지 하여 T점점을 형성하는 모든 쌍을 구했다. 여기에서는 T자 모양을 가진 모든 영역을 분할하였다고 가정하고, 남은 우세점에 대해서 B점점이기 위한 조건을 검사한다. T점점을 이루고 있는 모든 CDP를 제거하고 남은 CDP를 가지고 아래 조건을 검사한다.

- 1) D<sub>i</sub>에서 회전된 정도 D<sub>i-1</sub>D<sub>i</sub>D<sub>i+1</sub> < θ
- 2) 두점의 근접성 Distance(D<sub>i</sub>,D<sub>j</sub>) < α
- B = D<sub>i-1</sub>D<sub>i</sub>D<sub>i+1</sub>을 이동분 하여 지나는 점 R<sub>i</sub>는 직선 BD<sub>i</sub>을 지나 이웃한 외곽선에 만나
- 3) R<sub>i</sub>에서 회전된 정도 ∠R<sub>i-1</sub>R<sub>i</sub>R<sub>i+1</sub> = ∠D<sub>i-1</sub>D<sub>i</sub>D<sub>i+1</sub> ± β



파라미터 : α β θ  
그림 5. B점점이기 위한 조건

그림 5의 조건이 모두 만족하면  $\overline{D_i R_i}$ 는 굴곡부분이라고 판단하고 쌍으로 묶는다. 여기에서는 파라미터 값을 θ=145°, α=10, β=25°로 설정하였다. 그림 11(a)은 첫 번째 단계에서 나온 최종 분할된 모습이다. 자모 단위로 잘 쓰여진 패턴에서는 위의 두 알고리즘(B점점과 T점점)만을 가지고 분할 할 수 있다. 하지만 그림 11(a)에서 보는 바와 같이 자모가 붙어 있는 패턴에 대해서는 첫 번째 단계 알고리즘만을 가지고는 중성과 중성이 연결된 부분이 분할되지 않는다. 이렇게 분할하지 못한 이유는 중성과 중성이 연결된 부분의 굴곡이 완만하여 우세점을 생성시키기 못 하기 때문이다. 이러한 문제점을 보완하여 두 번째 단계에서는 불록 할 알고리즘을 적용시켜 홀의 모양에 따라 분할한다.

3.2 두번째 단계 알고리즘

절차 1에서는 첫번째 단계에서 추출된 쌍의 점들을 Bresenham 선분 알고리즘을 가지고 원 패턴에 선을 긋는다. 이렇게 그려진 선을 중심으로 해서 영역을 분리하기 위하여 4-connected component 알고리즘을 사용하여 각 영역을 서로 다른 값으로 설정한다(그림 6). 절차 2에서는 단계 1에서 추출된 각각의 영역에서 가장 오른쪽 위 부분을 시작으로 해서 시계방향으로 외곽선을 따라 체인코드를 추출한다. 체인코드를 구하는 것은 다음 단계에서 행해질 불록 할 알고리즘의 수행 시간을 줄이는 것과 함께 홀의 위치를 판별하는데 유용하기 때문이다. 불록 할을 이루고 있는 점들을 추출하기 위해서는 생성된 체인코드에서 가장 오른쪽 점(P<sub>1</sub>)과 가장 왼쪽 점(P<sub>2</sub>)을 구한다. 선분  $\overline{P_1 P_2}$ 을 기준으로 하여 오른쪽과 왼쪽에 있는 점들을 두 그룹으로 분리한다. 그런 다음 왼쪽 그룹은 선분  $\overline{P_1 P_2}$ 에서 왼쪽으로 가장 멀리 떨어진 점(P<sub>3</sub>) 그리고 오른쪽 그룹에서는 선분  $\overline{P_1 P_2}$ 에서 오른쪽으로 가장 멀리 떨어진 점(P<sub>4</sub>)을 구한다. 이렇게 구해진 네 선분  $\overline{P_1 P_2}$ ,  $\overline{P_3 P_4}$ 에 대해서 위의 방법과 마찬가지로 되풀이하며, 그룹에 점들이 두 점만이 존재 할 때까지 반복 처리한다. 두 점이 되면 이 점들은 불록 부분만을 이루는 최소의 점들이 된다. 그림 7에서 타원 안에 있는 점들이 불록 부분을 이룰 수 있는 최소의 점들이 된다

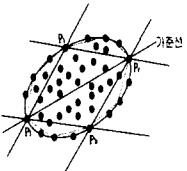
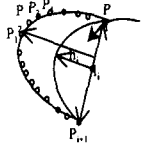


그림 6. 고유값 설정      그림 7. 불록 할 알고리즘

불록 할 알고리즘을 적용하여 나온 모양 중 오목하게 들어가는 부분을 홀이라 정의한다. 절차 3에서는 이 홀의 영역을 조사하여 분할 쌍을 찾는다. 홀은 첫 번째 단계에서 탐지를 못해서 생길 수도 있고 Y점점에서 생길 수도 있지만 잡음에 의해 생기는 작은 홀일 수도 있

다. 그러므로 여러가지 형태의 홀이 생기는 것들 중 우리가 원하는 홀을 판별하는 조건이 필요하다.

**홀이기 위한 조건**



- 1) 연결된 직선의 길이  $l_i > \theta_1$  ( $\theta_1 >= 10$ )  
 $(l_i = \overline{P_i P_{i+1}}, 0 <= i <= n, \text{ if } i=n+1 \text{ then } i=0)$
- 2)  $l_i$ 의 안쪽으로 지나가는 외곽선의 점들에 대해서  
 그 점과  $l_i$ 과 직교로 만나는 점과의 거리가 가장  
 먼 거리 값  $h_i > \theta_2$  ( $\theta_2 >= 7$ )

그림 8. 홀이기 위한 판별 조건

그림 8의 두 조건, 직선의 길이와 직선에서의 높이 값 모두 만족한다면 우리가 원하는 홀이라고 판단 할 수 있다. 절차 4에서는 홀 모양을 탐지한다. 절차 3에서 나올 수 있는 모양을 세 가지로 압축 할 수 있다. 첫 번째, B점점에서 분할하지 못한 부분 두 번째, T점점에서 분할하지 못해서 발생한 부분(그림 9), 세 번째 Y점점 분할이(그림 10) 있다. 이 세 가지 패턴은 첫 번째 단계의 전처리 과정인 우세점 탐지할 때 우리가 원하는 위치에 존재하지 않거나 처음부터 탐지하지 못해서 발생된 경우들이다.



그림 9. 첫번째 단계에서 놓친 부분 (홀 개수 = 1 & 홀 개수 = 2)

그림 10. Y점점 (홀 개수 = 3)

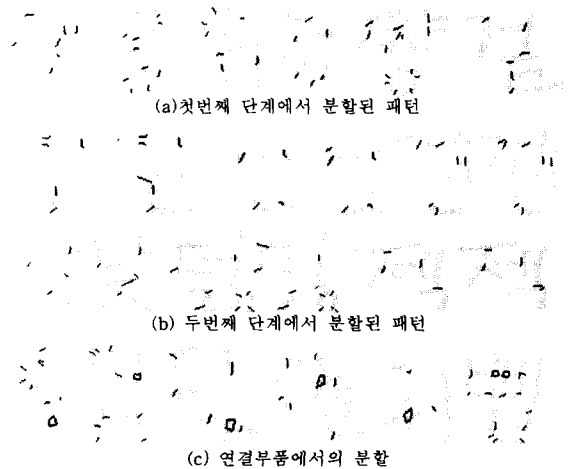
홀이 세 개일 때는 첫 번째 단계에서 T자 모양보다는 Y자 모양에 가까운 것으로 세 부분을 연결하는 역할을 한다. 또한 이것은 필기체 한글 패턴에서 많이 발생되는 패턴들 중의 하나이다. 그림 11(a)와(c)는 첫 번째 단계 결과 그리고 두 번째 단계를 통한 최종적인 분할 결과를 보여준다. 첫 번째 단계에서 증성과 종성사이를 분할하지 못한 부분을 두 번째 단계에서 완벽하게 분할하여 준다. 이것은 첫 번째 단계의 B점점 알고리즘에서 놓친 부분을 탐지하여 분할한 경우이다.



그림 11. 분할 과정 (a) 첫 번째 단계 결과 (b) 블록 형 알고리즘 (c) 두 번째 단계 결과

**4. 실험**

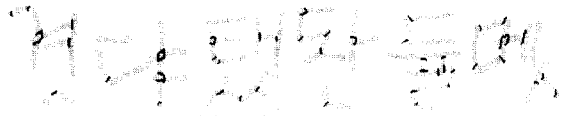
실험은 Pentium II에서 Visual C++로 구현하였으며, 테스트는 PE92 데이터베이스로 수행하였다. 실험 결과 정서체로 쓰여진 데이터에서는 첫 번째 단계 알고리즘만 가지고도 잘 분할되었다. 하지만 자모 단위가 붙은 경우에는 두 번째 단계 알고리즘을 적용해야 분할된 경우를 볼 수 있다. 그림 12는 PE92 데이터 중 잘 분할된 모양을 수집해 놓은 것이다.



(a) 첫번째 단계에서 분할된 패턴

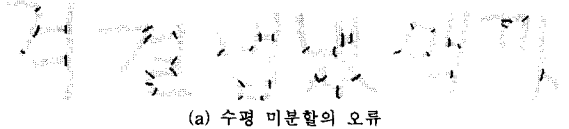
(b) 두번째 단계에서 분할된 패턴

(c) 연결부품에서의 분할



(d) 겹침에 의한 불완전한 분할

그림 12. 잘 분할된 패턴



(a) 수평 미분할의 오류



(b) 수직 미분할의 오류

그림 13. 잘 분할되지 않는 패턴

그림 13에서 나타나는바 같이 증성과 종성이 연결된 부분 또는 두 자음이 교차하는 부분에서 하나의 획으로 붙어 분할되지 못하는 문제가 발생했다. 이를 해결하기 위해서는 한글 모양에 대한 지식에 관한 휴리스틱을 이용한 방법을 병행해야 한다.

**5. 결론**

분할된 패턴영역을 살펴보면 B점점과 T점점에 대해서 놓치지 않고 잘 분할한 것을 볼 수 있다. 하지만 본 논문에서 제안한 방법은 한글모양에 따라 분할하기 때문에 수평 미분할의 오류, 수직 미분할의 오류는 제안한 방법 측면에서는 만족하지만, 최소 자모 단위로 분할한다는 기본 취지에는 벗어남이 때문에 실패로 간주하였다. 한글의 패턴을 완벽하게 분할하기 위해서는 특수한 경우에 대한 휴리스틱 정보가 들어가야만 하겠다. 세선화에서는 획이 두꺼워지는 부분에서 생기는 잔간지를 교정해주는 과정이 필요하고 또한 그 부분에서 오류가 발생될 소지가 있으나 제안한 방법에서는 이러한 문제점을 해결할 수 있다.

**참고 문헌**

1. 김수형, 정선화, 오일석, "필기 한글 문자의 오프라인 인식에 관한 사례 연구," 한국 정보과학회 춘계 학술발표논문집, pp. 396-398, 1998.
2. 오일석, 이진선, 유태웅, 박정선, "한글 문자 패턴을 위한 영역-기반 모양 분해 알고리즘," 정보과학회 춘계 학술발표논문집, 713-715, 1998.
3. C. Arcelli and L. Serino, "Finding near-convex shapes in complex patterns," *Proceedings of The International Workshop on Structural and Syntactic Pattern Recognition*, pp. 217-226, 1992.
4. C. H. Teh and R. T. Chin, "On the detection of dominant points on digital curves," *IEEE PAMI*, Vol. 11, No. 8, August 1989.
5. D. S. Doermann and Aziel Rosenfeld, "The interpretation and reconstruction of interfering strokes," *Third International Workshop on Frontiers in Handwriting Recognition* pp.41-50, 1993.
6. H. Nishida and S. Mori, "Structural analysis and description of curves by quasi-topological features and singular points," *Structured Document Image Analysis*, pp. 139-187, 1991.
7. Il-Seok Oh, Jin-Seon Lee, Tae-Woong Yoo, Jeoung-Sun Park, Ki-Chun Hong "Region-based Shape Decomposition Algorithm for Hangul Chatacter Patterns," *Proceeding of IWFHR'98*, pp. 557-566, 1998.
8. J. Xu, "Morphological decomposition of 2-D binary shapes into conditionally maximal convex polygons," *Pattern Recognition*, Vol. 29, No. 7, pp. 1075-1104, 1996.
9. K. Abe, C. Arcelli, T. Hisajima, and T. Ibaraki, "Parts of planar shapes," *Pattern Recognition*, Vol. 29, No. 10, pp. 1703-1711, 1996.
10. Louisa Lam, Seong-Whan Lee, Ching Y.Suen, "Thinning Methodologies-A Comprehensive Survey" *IEEE PAMI*, Vol. 14, No. 9, pp. 869-885, 1992.