

필기 한글 문자의 모양 분해

(Shape Decomposition of Handwritten Hangul Characters)

박정선[†] 홍기천^{**} 오일석^{***}

(Jeong-Sun Park) (Ki-Chun Hong) (Il-Seok Oh)

요약 필기 한글의 문자나 단어 인식에 있어 패턴을 구성하는 획 성분을 추출하는 작업은 매우 중요하다. 세선화와 직선근사에 기반한 고전적인 방법은 원래 패턴을 크게 왜곡한다는 단점을 가지고 있다. 우리는 이러한 문제점을 해결하기 위하여 한글 패턴에 적합한 모양 분해 알고리즘을 제안한다. 한글 패턴은 T-접점과 B-접점이라는 두 가지 모양 특징을 중심으로 분할할 수 있다고 관찰에 근거하여 알고리즘을 설계하였다. 또한 세 개 이상의 획이 복잡한 형태로 만나는 결합 지점을 강건하게 처리하는 방법도 제시한다. 제안한 알고리즘을 PE92 데이터베이스에 적용한 결과를 제시한다.

Abstract In the recognition of handwritten Hangul characters or words, extraction of strokes is an important task. The classical approach based on the thinning and line approximation, have some drawbacks of distortion of the original patterns. In this paper, we propose a shape decomposition algorithm specifically designed for Hangul patterns to resolve such problems. The proposed algorithm is designed based on the observations that Hangul patterns can be decomposed with two shape features called T-junction and B-junction. Additionally we present a robust technique which deals with the situation where three or more strokes meet at a joint. We present experimental results obtained by applying the proposed method to PE92 database.

1. 서론

한글 문자에 대한 자동 인식은 1970년대부터 많은 연구가 되어왔다[1,2]. 필기 한글 패턴을 분석하고 인식하기 위해서는 접촉 글자 분할, 자모 분할, 획 분할 등의 작업이 매우 중요하다. 한글은 글자간 유사성과 부류수의 방대성 등으로 인해 획을 추출하고 이를 분석하여 인식하는 접근방법이 주류를 이루기 때문에, 한글 문자의 획 분석은 어느 문자 집합보다 중요하다.

기존 논문은 이들 작업을 위하여 세선화(thinning)와 직선근사(line approximation)에 의존한다[3,4,5,6,7]. 하지만 세선화에 기반한 획 추출 알고리즘을 필기 한글 패턴에 적용하였을 경우 발생하는 정보 손실 및 패턴 왜곡

은 문자 인식을 매우 어렵게 만드는 요소가 된다[8]. 이런 세선화의 단점을 보완하기 위한 방법론이 제시되어왔다. 미세한 잡음인 화소를 전혀리로 제거하고 나서 문자 영상의 평균 획 굵기, 한글의 구성 원리, 그리고 한글 자모의 특성을 이용하여 접촉 획의 분리와 Y형태의 왜곡을 해결하는 지식기반 세선화 방법[9], 이진 영상이 아닌 명암 영상을 입력으로 하여 이를 3차원 지형으로 간주하고 다양한 기하학적 특징점들을 찾아 골격선을 추출하는 방법[10,11] 등이 있다. 이러한 방법들 역시 입력 패턴에 대한 제약 조건들로 인한 한계점을 가지고 있다.

세선화 방법의 대안으로서 모양 분해(shape decomposition) 알고리즘이 있다[12,13,14,15,16]. 이들 알고리즘의 목적은 최소 개수의 유사 볼록 부품(near convex parts)으로 원래 패턴을 분해하는 것이다. 이러한 알고리즘은 모양은 분해하되 분해된 요소들을 다시 조합하면 원래 모양을 그대로 재생할 수 있다는 장점을 가지고 있다. 기존의 모양 분해 알고리즘은 대부분 임의의 2차원 이미지 패턴을 대상으로 개발되었으며, 문자에 적용된 예는 드물다.

필기 숫자나 영문에 적용된 경우, 획 두께의 변화된

[†] 비 회 원: 전북대학교 컴퓨터과학과
jspark@cs.chonbuk.ac.kr

^{**} 정 회 원: 전북대학교 컴퓨터과학과
kchong@cs.chonbuk.ac.kr

^{***} 통신회원: 전북대학교 컴퓨터과학과 교수
정보검색 시스템 연구센터
iso@moak.chonbuk.ac.kr

논문접수: 1999년 12월 30일

심사완료: 2001년 5월 14일

지점을 탐지하고 기하학 조건에 의해 이 지점이 다른 문자와 교차해서 생기는 부분인지를 판단하여 분할하는 방법[17], 수직으로 과분할(over segmentation)을 허용하여 분할된 각 부품들간의 매칭패스 조건을 검사하여 재조합을 통한 분할 방법이 있다[18]. 복잡한 모양을 가진 한자에 대한 연구는 비교적 많이 되어 왔다[19,20]. 한글에 적용된 경우로는, 우편물 주소영상에 나타나는 단어를 인식하기 위해 입력 패턴을 수직으로 과분할(over segmentation)하고 인식 과정에서 이들을 조합하여 사용하는 방법이 있다[21]. 이 방법은 우편물에 나타나는 단어에 국한된 기법으로서, 일반적인 경우에 적용할 수 있는 기법으로는 세션화 밖에 찾아 볼 수 없다[2]. 한글 인식은 획 분석에 의존하는 경우가 많기 때문에 이에 대한 연구를 심도 있게 수행하는 것은 매우 중요하다.

본 논문은 한글 패턴에 적합한 모양 분해 알고리즘을 제안한다. 입력은 한글 문자의 2차원 흑백 영상이며, 출력은 유사 볼록(near-convex)한 부품(parts)으로 분해된 패턴이다. 모양 분해는 두 가지 다른 응용을 생각할 수 있다. 첫째는 한글 단어를 문자 단위로 또는 한글 문자를 자모 단위로 분할하는 것이다. 한글 단어를 문자 단위로 분할하는 경우의 목적은, 다른 글자에 속하는 획이 같은 부품에 포함되지 않도록 하는 최소 개수의 부품 집합으로 분해하는 것이다.

한 개의 문자를 자모(초성, 중성, 종성)로 분할하는 응용에서는, 다른 자모에 속하는 부품이 같은 요소에 포함되지 않도록 하는 최소 개수의 부품 집합으로 분해하는 것이다. 두 경우 모두 최선은 각 문자나 자모가 하나의 부품이 되도록 분해하는 것인데 이것이 불가능하기 때문에, 제안한 알고리즘을 사용하여 유사 볼록한 부품 집합으로 분해한 후 여러 부품을 조합하여 한 개의 문자 또는 자모를 만들 수 있다.

두 번째 응용은 한글 인식이다. 모양 분해된 패턴으로부터 다양한 특징을 추출하여 사용할 수 있다. 예를 들어 수직 또는 수평 획 부품만을 쉽게 추출할 수 있다. 또한 모양 분해 결과로부터 골격선을 추출하여 인식기의 입력으로 사용할 수도 있다. 예를 들어 포항공대 인식기[9,22]는 필기 한글이 모양 분해되었다는 가정 하에서 작동하므로 이 인식기의 전 단계 모듈로 사용할 수 있다.

제안한 알고리즘은 필기 한글 패턴의 모양에 대한 주의 깊은 분석을 바탕으로 설계하였다. 한글 패턴은 두 개의 획이 T자 형태로 만나는 T-접점(T-junction), 하나의 획이 굽은 B-접점(B-junction), 그리고 세 개의 획을 연결해주는 결합 부품(joint part)을 중심으로 분할할 수 있다. 이들을 찾기 위하여 패턴 외곽선에 존

재하는 우세점(dominant point)을 탐지하고 이들을 특징점(feature point)으로 사용한다[23].

오목한 부분에 있는 특징점을 오목 특징점(concave feature point)이라 하고, 볼록한 부분에 있는 특징점을 볼록 특징점(convex feature point)이라 정의한다. T-접점은 오목 특징점과 오목 특징점의 쌍으로, B-접점은 오목 특징점과 볼록 특징점의 쌍으로 정의하고, 기하학적 조건을 검사하여 찾는다. 이를 첫 번째 단계라 한다. 위 과정은 정서체 한글 패턴에 대해서는 좋은 성능을 보이지만, 난필로 쓴 한글 패턴에서는 완벽한 분해가 되지 않는 경우가 많다. 이것은 심한 굴곡과 복잡한 획 연결 등으로 인하여 우세점 탐지 알고리즘만으로는 정확한 특징점 탐지가 불가능하기 때문이다. 이런 이유로 난필로 쓴 패턴에 대해서는 유사 볼록 부품으로 분할되지 못한 경우가 다수 발생한다.

우세점 탐지 알고리즘에서 특징점 탐지를 못한 것을 보완하기 위하여 볼록 핵(convex hull) 알고리즘을 적용하여 특징점을 탐지한다[24]. 첫 번째 단계에서 분할된 부품 각각에 대해 볼록 핵 알고리즘을 적용하여 영상에서 원 영상을 제거하고 남은 연결 요소(connected component)들을 오목 영역(concave region)이라 정의한다.

이 오목 영역에서 가장 내부에 있는 점을 탐지하여 오목 특징점으로 간주한다. 오목 특징점이 한 개일 때는 B-접점으로 분할이 이루어지며, 두 개일 때는 T-접점의 분할이 이루어진다. 세 개 이상일 때는 세 개 이상의 획이 복잡하게 만나는 상황으로서 다른 획들을 연결해주는 지점으로 간주하여 결합 부품(joint part)으로 분할한다. 이를 두 번째 단계라 한다.

제안한 알고리즘은 PE92 데이터베이스에 대해 실험하였다[25]. 실험 결과, 정서체 패턴 대부분에 대해서는 첫 번째 단계에서 T-접점과 B-접점을 찾아 성공적으로 분해가 완료되었다. 하지만 난필로 쓰인 패턴에 대해서는 두 번째 단계를 거쳐야만 분해가 완료됨을 볼 수 있었다. 2장은 한글이 갖는 고유한 모양을 분석하고 이를 토대로 알고리즘을 설계한다. 3장은 전체적인 알고리즘을 소개하고, 4장은 알고리즘의 각 단계를 자세히 기술한다. 5장은 실험환경, 결과, 그리고 토론을 기술하고, 6장에서 결론을 기술한다.

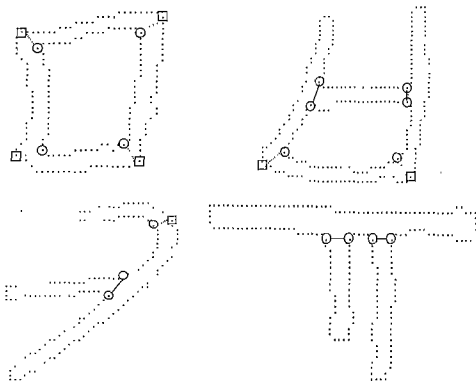
2. 한글 모양 분석

한글은 표 1에 보인 바와 같이 자음 14개와 모음 10개 그리고 쌍 자음 16개와 쌍 모음 11개로 구성되고, 이들을 2차원으로 배치하여 문자를 만든다.

표 1 한글 자모 집합

단일 자음(14)	ㄱ ㅋ ㆁ ㄷ ㅌ ㄴ ㅅ ㅈ ㅊ ㅍ ㅎ ㅇ ㅊ ㅊ ㅊ ㅊ ㅊ ㅊ
이중 자음(16)	ㄱ ㅈ ㅊ ㅊ ㅊ ㅊ ㅊ ㅊ ㅊ ㅊ ㄴ ㄴ ㄴ ㄴ ㄴ ㄴ ㄴ ㄴ
단일 모음(10)	ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ
이중 모음(11)	ㅘ ㅙ ㅚ ㅜ ㅝ ㅞ ㅟ ㅠ ㅡ ㅢ ㅣ

이들 패턴을 유사 블록 부품으로 분해하기 위해 두 가지 모양 특징(shape feature)에 주목한다. 하나의 획이 구부러진 지점을 B-접점(B-junction)이라고 하고, 두 획이 T자 모양으로 만나는 지점을 T-접점(T-junction)이라고 정의하고, 이 두 가지 모양 특징을 중심으로 분할을 시도한다.



○: 오목 특징점 □: 볼록 특징점

그림 1 T-접점과 B-접점

B-접점과 T-접점은 패턴의 외곽선(contour)을 따라 방향이 급격히 변하는 점을 중심으로 존재한다. 이러한 점을 특징점이라 하고, 오목성에 따라 오목 특징점(concave feature point)과 볼록 특징점(convex feature point)이라 정의한다. 그림1에서 보는 바와 같이 T-접점은 실선으로 표시된 오목 특징점과 오목 특징점의 쌍으로 이루어지며, B-접점은 점선으로 표시된 오목 특징점과 볼록 특징점의 쌍으로 이루어진다.

한글 패턴은 영어와 중국 언어와 같이 두 획이 교차한 모양(X, f, l, 大, 土, etc)으로 이루어지지 않으므로, T-접점과 B-접점만으로 한글을 분해할 수 있다. 그러나 무 제약으로 필기된 한글 문자에서는 획이 수직으로 만나지 않고 그림2와 같이 비스듬히 만나는 경우가 빈번하

다. 이런 경우는 T-접점과 B-접점만으로 분할하기가 자연스럽지 않고 모호성을 발생시킨다.

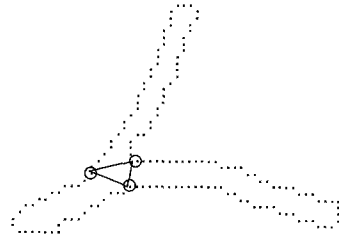


그림 2 결합부품(joint part)

모호성이 발생하는 지점을 자연스럽게 처리하기 위해 결합 부품(joint part)을 정의한다. 결합 부품은 인접한 획들이 교차하는 한 점을 제공함으로써 인식기의 입력 패턴으로 들어갈 골격선을 구할 때 유용하게 사용할 수 있다. 이런 곳의 대부분은 획의 굴곡이 완만하여 오목 특징점이 발생하지 않아 T-접점이나 B-접점으로 분할하지 못한 경우이다. 결합 부품 탐지는 알고리즘의 두 번째 단계에서 이루어진다.

3. 알고리즘 개요

제한한 모양 분해 알고리즘은 2장의 분석 결과에 따라 설계되었으며, 두 단계로 구성된다. 첫 번째 단계는 아래 알고리즘 *Hangul_shape_decomposition*의 steps 2-11로 구성되며, 두 번째 단계는 steps 12-19로 구성된다. 알고리즘의 입력은 한글 문자를 위한 2차원 흑백 영상이며, 출력은 이 패턴을 모양 분해한 결과로서 유사 블록 부품의 집합이다.

Procedure *Hangul_shape_decomposition*;

Input : a Hangul character pattern P /* P=2-D binary array */

Output : a set of near-convex parts S

begin

1. for each of connected components in P, insert it into S and do steps 2-19;
/* first pass of the algorithm */
2. for the part p in S, *detect_feature_point*(p, F⁺, F⁻);
/* detect T-junctions */
3. for each pair of two points f_i and f_j in F⁻ do
4. if the pair satisfies *is_T_junction*(f_i, f_j) begin
5. insert the pair $f_i - f_j$ into T-junction set T;
6. remove f_i and f_j from F⁻;
- end;
- /* detect B-junctions */
7. for each pair of a point f_i in F⁻ and a point f_j in F⁺ do
8. if the pair satisfies *is_B_junction*(f_i, f_j) begin

9. insert the pair $f_i - f_j$ into B-junction set B;
10. remove f_i from F^- and f_j from F^+ ;
- end;
- /* decompose the part into smaller parts */
11. divide the part p using the junctions in T and B, and insert new parts into S;
- /* second pass of the algorithm */
12. for each part p in S do
13. if p does not satisfy $is_near_convex(p)$ begin
14. compute a convex hull over p;
15. identify the concave regions and their innermost points;
- /* decompose the part into smaller parts */
16. case number of concave regions do begin
17. 1: make a T-junction, divide p using it, and insert new parts into S;
18. 2: make a B-junction, divide p using it, and insert new parts into S;
19. 3 or more: make a joint part, divide p using it, and insert new parts into S;
- end;
- end;
- end;

초기 입력 패턴은 흑백 영상을 0(흑)과 1(백)로 표시한 이진 패턴이며 이 패턴에 대하여 연결 요소 (connected component)를 추출한다. 연결 요소의 개수는 문자열의 길이 또는 자모나 문자 단위의 교차 정도에 따라 다를 수 있으며, 각 연결 요소로 S를 초기화한 후 알고리즘을(steps 2-19) 수행한다. 먼저 우세점 알고리즘 [23]을 사용하여 입력 패턴의 외곽선에 있는 우세점(dominant point)을 탐지한다. 이 우세점들 중에는 미세한 돌출 부분에서 탐지된 점도 포함되어 있기 때문에, 이런 우세점을 제거하는 과정이 필요하다. 이렇게 제거하고 남은 우세점을 특징점(feature point)이라 한다. T-접점과 B-접점의 후보 점들로 사용하기 위해서 추출된 특징점들을 오목 특징점과 볼록 특징점으로 구분하고 오목 특징점 집합을 F^- , 볼록 특징점 집합을 F^+ 로 표시한다. T-접점에 위치한 특징점을 찾기 위해서는 모든 오목 특징점 쌍 $f_i - f_j$ 에 대해, 기하학 조건을 검사한다(step 4의 $is_T_junction(f_i, f_j)$). 탐지된 T-접점을 집합 T에 넣고, 해당 특징점을 F^- 에서 제거한다. B-접점을 찾기 위해서는, 모든 오목 특징점 f_i -볼록 특징점 f_j 쌍에 대해 기하학 조건을 검사한다(step 8의 $is_B_junction(f_i, f_j)$). 탐지된 B-접점을 집합 B에 넣는다. T와 B에 있는 T-접점과 B-접점을 이용하여 원래 부품 p를 분할하고, 그 결과 부품들을 S에 삽입한다. 여기까지를 첫 번째 단계라 한다.

제한한 모양 분해 알고리즘의 최종 출력은 유사 볼록 부품으로 이루어진 집합이어야 한다. 정서체로 쓰여진 패턴에 대해서는 첫 번째 단계를 마치고 나면 대부분

완전한 분해가 이루어지지만, 난필로 쓰여진 패턴은 유사 볼록 부품으로 완벽한 분해가 이루어지지 않는다. 이런 경우를 처리하기 위하여 두 번째 단계(steps 12-19)를 수행한다. 우선 첫 번째 단계에서 얻어진 부품(즉, S에 속하는 부품) 각각에 대해 볼록 힐 알고리즘을 적용하여(step 13의 $is_near_convex(p)$) 유사 볼록 부품인지를 검사하고, 만약 유사 볼록 부품이 아니라면 step 14-19를 수행한다.

부품 p에 대한 볼록 힐 영상에서 원 영상을 제거하고 남은 영역을 오목 영역(concave region)이라 한다. 오목 영역에서 가장 내부에 있는 점을 오목 특징점으로 탐지한다. 오목 특징점은 입력 부품의 굴곡 정도에 따라 여러 개가 나올 수 있다. 오목 특징점이 한 개일 때는 확실히 반대편에 있는 볼록 특징점을 찾아 이들 두 특징점을 쌍으로 하는 B-접점으로 분할이 이루어지며, 두 개일 때는 이들을 쌍으로 하는 T-접점으로 분할한다. 세 개 이상일 때는 세 개 이상의 획이 복잡하게 만나는 지점으로 간주하여 이들 오목 특징점으로 구성되는 결합부품(step 19)으로 분할한다. 여기까지를 두 번째 단계라 한다.

4. 세부 알고리즘

4.1 특징점 탐지

이 절은 3장 알고리즘의 $detect_feature_point()$ 에 해당한다. 원 입력 패턴(그림3(a))은 스캔 상태에 따라 외곽선에 미세한 돌출부분이 많이 존재하는데, 전처리 과정을 통하여 이런 잡음을 제거한다.



(a) 입력 패턴



(b) 외곽선 추출(체인코드 형성)



(c) 우세점 추출(Teh-Chin 알고리즘)



(d) 특징점 추출(검은점: 오목 특징점, 회색점: 볼록 특징점)

그림 3 특징점 탐지

우선 입력 패턴으로부터 외곽선을 따라 시계방향으로 체인 코드를 구한다(그림 3(b)). 이 체인 코드에 우세점(dominant point) 탐지 알고리즘을 적용하여 방향값이 급격히 변화된 점을 추출한다(그림 3(c)). 우세점 탐지 알고리즘은 이전에 많이 제안되어져 왔는데, 그 중에서 Teh-Chin 알고리즘[23]은 패턴의 크기나 변화에 민감하지 않기 때문에 본 논문에서는 Teh-Chin 알고리즘을 사용한다. 그러나 Teh-Chin 알고리즘은 너무 많은 우세점을 탐지하는 문제점이 있다. 이렇게 많이 발생한 우세점들 중 미세한 돌출 부분에서 생기는 것들을 제거하는 과정이 필요하다. 이를 위하여 한 우세점에 대해 좌우 이웃한 우세점과의 각도를 계산한다. 즉, 우세점 d_i 에 대해 좌우 이웃한 우세점과의 각도 $\angle d_{i-1}d_i d_{i+1}$ 를 계산하고, 이 각도가 임계값 T 이상이면 잡음에 의한 우세점이라 간주하고 제거한다. 실험에서는 T 를 145° 로 하였다. 이렇게 추출된 우세점을 특징점(feature point)이라 정의하고, 이들을 다음 단계에서 T-접점과 B-접점 탐지에 사용한다.

T-접점은 두 개의 오목 특징점의 쌍으로, B-접점은 하나의 오목 특징점과 하나의 볼록 특징점 쌍으로 구성되기 때문에, 탐지된 특징점을 오목 특징점과 볼록 특징점으로 구분하는 과정이 필요하다(그림 3(d)). 이 연산을 위하여 하나의 특징점 d_i 가 좌우 이웃한 특징점 d_{i-1} 과 d_{i+1} 로 구성되는 선분($\overline{d_{i-1}d_{i+1}}$)의 오른쪽에 있다면 볼록 특징점이 되고, 왼쪽에 있다면 오목 특징점이 된다.

4.2 T-접점 검사

이 절은 3장 알고리즘의 $is_T_junction()$ 에 해당한다. T-접점을 위한 조건 검사는 F^- 에 속하는 두 개의 오목 특징점 쌍에 대해 수행한다. 오목 특징점 개수가 n 이라면, 총 nC_2 개의 조합에 대해 수행한다. 한 개의 쌍을 구성하는 오목 특징점을 각각 D_i 와 D_j 라 하자. D_i 바로 앞에 있는 특징점 D_{i-1} 과 이루는 방향 선분 L_1 , D_i 바로 다음 특징점 D_{i+1} 과 이루는 방향 선분 L_2 를 생각할 수

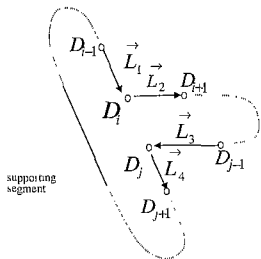
있다. 마찬가지로 D_j 에 대해서 L_3 과 L_4 를 구성한다. 이들이 그림 4에 있는 세 가지 조건을 모두 만족하면 T-접점으로 판정한다. T-접점으로 판정되면 D_i - D_j 쌍을 T-접점 집합 T 에 삽입하고, 다음 단계인 B-접점 검사에 참여하지 못하도록 D_i 와 D_j 를 오목 특징점 집합 F^- 에서 제거한다.

조건1은 D_i 와 D_j 간의 거리가 임계값 T_1 이내에 들어야 한다는 것이다. 이 조건은 거리 제한을 두어, 조건2와 조건3을 우연히 만족하여 틀린 T-접점을 생성하는 경우를 막기 위함이다. 조건2는 벡터 \vec{L}_1, \vec{L}_4 가 동일직선상에 위치 해야한다는 것이다. $Angle(\vec{L}_1, \vec{L}_4)$ 는 두 개의 벡터 \vec{L}_1 와 \vec{L}_4 가 이루는 각도이다. ϵ_1 으로 약간의 오차를 허용한다. 조건3은 벡터 \vec{L}_2, \vec{L}_3 가 서로 다른 방향을 가지고 평행 해야한다는 것이다. ϵ_2 를 두어 약간의 오차를 허용한다. 이 세 가지 조건은 최소 조건 집합이다. 즉, 세 가지 중 하나라도 만족하지 않으면 T 모양을 이루지 못하는 경우가 T-접점으로 생성되고, 또한 이들 세 가지면 충분하다. 실험에서는 $T_1=10, \epsilon_1=35^\circ$, 그리고 $\epsilon_2=2$ 로 설정하였다.

4.3 B-접점 검사

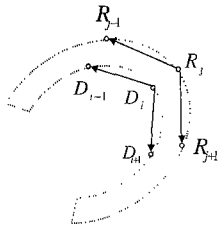
이 절은 3장 알고리즘의 $is_B_junction()$ 에 해당한다. B-접점 검사는 F^- 에 속하는 하나의 오목 특징점 D_i 와 F^+ 에 속하는 하나의 볼록 특징점 R_j 를 쌍으로 구성하고, 이들에 대해 그림 5의 조건을 적용하여 수행한다. 실제로는 R_j 는 D_i 에 대해 획 반대편에 있는 가장 가까운 볼록 특징점으로 선택한다. D_i 와 R_j 가 그림 5의 세 가지 조건 모두를 만족하면 B-접점으로 판정한다. B-접점으로 판정되면 D_i - R_j 쌍을 B-접점 집합 B 에 삽입하고, D_i 와 R_j 를 특징점 집합 F^- 와 F^+ 에서 제거한다.

조건1은 D_i 앞과 뒤에 있는 특징점과 이루는 각도가 임계값 안에 드는지 검사하여 획이 굽어져 있는지 판단한다. 조건2는 R_j 앞과 뒤에 있는 특징점과 이루는 각도가 조건1의 안쪽 각도에서 ϵ_3 의 범위를 넘는지를 판단



- 조건1: 근접성(proximity)
 $Distance(D_i, D_j) < T_1 \dots\dots\dots(1)$
- 조건2: 공선성(collinearity)
 $0^\circ - \epsilon_1 \leq Angle(\vec{L}_1, \vec{L}_4) \leq 0^\circ + \epsilon_1$ AND
 $Distance(\vec{L}_1, D_j) \leq \epsilon_2$ AND $Distance(\vec{L}_4, D_i) \leq \epsilon_2 \dots\dots(2)$
- 조건3: 반대 방향 평행성(antiparallelism)
 $180^\circ - \epsilon_1 \leq Angle(\vec{L}_2, \vec{L}_3) \leq 180^\circ + \epsilon_1 \dots\dots\dots(3)$

그림 4 T-접점 조건



조건1: 안쪽 각도(inner angle)
 $\angle D_{i-1}D_iD_{i+1} < T_2$ (4)

조건2: 바깥쪽 각도(outer angle)
 $\angle D_{i-1}D_iD_{i+1} - \epsilon_3 < \angle R_{j-1}R_jR_{j+1} < \angle D_{i-1}D_iD_{i+1} + \epsilon_3$(5)

조건3: 근접성(proximity)
 Distance(D_i, R_j) < T_3 (6)

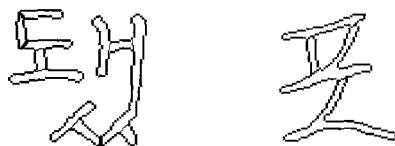
그림 5 B-점점 조건

한다. 조건3에서는 D_i 와 R_j 간의 거리가 임계값 안에 들어야 한다. 실험에서는 $T_2=145^\circ$, $T_3=10$, $\epsilon_3=10^\circ$ 로 설정하였다.

이제 T-점점 집합 T와 B-점점 집합 B 속에 있는 점점들을 가지고 분할 작업을 수행한다. 분할은 점점을 구성하는 두 개의 특징점 사이에 직선을 그어 수행한다. 분할 결과로 나오는 부품들을 S에 삽입한다. 여기까지가 전체 알고리즘의 첫 번째 단계이다. 그림6은 첫 번째 단계의 결과다. 비교적 곧은 획을 갖는 ‘됐’자는 T-점점과 B-점점을 제대로 찾아 유사 블록 부품으로 분해가 잘 이루어졌다. 단, 가장 오른쪽 획에서 분할이 덜 이루어졌다. 굴곡이 많은 ‘포’자에서는 두 곳에서 분할이 이루어지지 않았다. 미 분할이 일어난 이유는 특징점 추출 과정에서 굴곡 부분이 완만하여 오목 특징점을 탐지하지 못했기 때문이다. 이러한 현상 때문에 다음절부터 시작하는 두 번째 단계를 거치도록 하였다.

검사한다. 유사 블록으로 판정된 부품은 더 이상의 처리가 필요하지 않지만 유사 블록하지 않은 부품에 대해서는 또 한번의 분할을 시도한다. 유사 블록 검사를 위하여 Zimmer가 제안한 블록 형 알고리즘을 사용한다[24].

이 알고리즘은 흑은 물체, 백은 배경에 해당하는 이진 영상을 입력으로 받는다. 우리의 경우는 글자의 획에 해당하는 부품 영역은 흑, 나머지 영역은 배경으로 백이다(그림8(a)). 꼭지점 좌표로 정의되는 다각형을 입력으로 받아 불록한 다각형을 출력하는 전통적인 블록 형 알고리즘과 달리, Zimmer 알고리즘은 이진 영상을 입력으로 받아 이진 영상을 출력한다. 물론 연산은 마스크를 사용한 영상-기반으로 수행된다. 이 알고리즘은 그림7의 5x5 마스크를 사용하여 입력 영상을 2~3번 정도 스케닝함으로써 수행이 완료되고 마스크 연산이 단지 흑 화소 개수를 세면되기 때문에 고속으로 수행할 수 있다. 본 논문은 이러한 이유로 Zimmer 알고리즘을 선택하였다.



가는 선: T-점점 분할
 굵은 선: B-점점 분할

그림 6 알고리즘 첫 번째 단계의 결과

4.4 유사 블록 검사

이 절은 3장 알고리즘의 *is_near_convex()*에 해당한다. 한글 모양 특성상 첫 번째 단계만을 가지고 분할이 완료되는 경우도 있지만(그림6의 연결요소 ‘도’), 특징점의 불완전한 탐지에 의해 유사 블록 부품으로 분해되지 않는 경우도 종종 발생한다(그림6의 연결요소 ‘깃’과 ‘포’). 이러한 이유로 두 번째 단계의 시작에서는, 첫 번째 단계에서 얻은 모든 부품에 대해 유사 블록 여부를

p7	p6	p5	p4	p3
p8				p2
p9		C		p1
p10				p16
p11	p12	p13	p14	p15

그림 7 블록 형 영상 생성 5x5 마스크

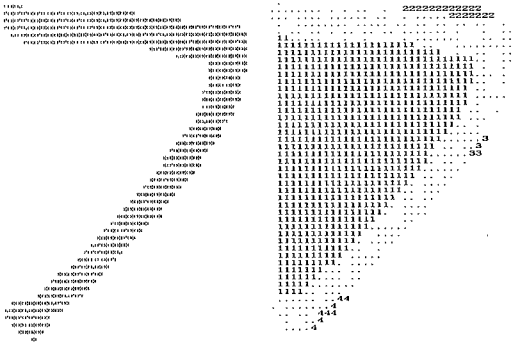
$$\text{sum} = \sum_{i=1}^{16} p_i + (p_2 \& !p_3 \& p_4) + (p_6 \& !p_7 \& p_8) + (p_{10} \& !p_{11} \& p_{12}) + (p_{14} \& !p_{15} \& p_{16}) \dots \dots \dots (7)$$

if sum > 7 then '1' else "no change".....(8)

알고리즘은 다음과 같이 수행된다. 입력 영상의 좌상단에서 출발하여 열-우선으로 영상을 스캔하여 값이 1인 화소는 제외하고 0을 갖는 화소 P에 대해 그림7의 마스크 연산을 수행한다. P에 C를 맞춤으로써 마스크를

씩은 다음 수식7로 sum을 계산한다. 이 식에서 P_i 는 영상의 해당 화소 값으로서 0 또는 1이다. !은 0은 1로, 1은 0으로 바꾸는 complement 연산자이고, &는 논리 연산자 AND이다. 이렇게 얻어진 sum 값이 7보다 크면 현재 화소 값을 1로 변경시키고 그렇지 않다면 그대로 둔다. 보다 자세한 알고리즘 설명은 [24]에 찾아보기 바란다.

하나의 부품에 대한 유사 블록 검사는 다음과 같이 이루어진다. 입력 부품 영상(그림8(a))에 대해 앞에서 설명한 알고리즘을 사용하여 블록 헐 영상(그림8(b))을 계산한다. 이렇게 얻은 블록 헐 영상을 구성하는 화소는, 원래 부품에 속하는 화소(그림8(b)에서 .로 표시)와 부품의 오목한 부분을 채우는 화소(그림8(b)에서 숫자로 표시)로 구분할 수 있다. 우리는 숫자로 표시된 화소로 구성되는 영역 각각을 오목 영역(concave region)이라 부른다. 그림8(b)에서는 네 개의 오목 영역이 있다. .로 표시된 화소의 개수를 세어 부품영역의 면적 R_p 를 계산하고, 숫자로 표시된 화소 개수를 세어 오목 영역의 면적 R_c 를 계산한다. 부품 영역과 오목 영역 면적의 비율을 R_c/R_p 로 계산하고, 이 비율이 임계값 T_4 를 넘으면 유사 블록하지 않다고 결정한다. 실험에서는 T_4 값으로 1/3을 사용하였다.

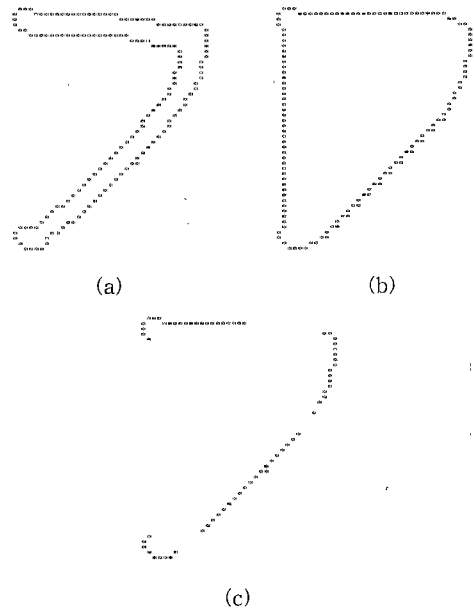


(a) 부품 영상 (b) 블록 헐 영상(오목 영역은 숫자로 표시)

그림 8 유사 블록 검사

4.5 오목 영역 중심으로 분할

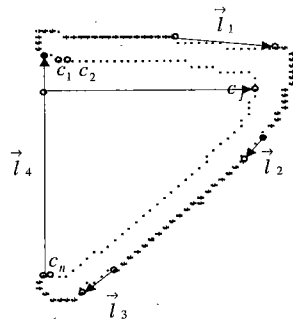
앞에서 유사 블록하지 않다고 판정된 부품에 대해 분할을 시도한다. 분할은 오목 영역의 가장 깊은 점을 중심으로 수행한다. 오목 영역은 부품의 외곽선을 따라 존재하므로, 이 외곽선을 이용하여 분할 지점을 찾는다.



a) 입력 영상의 외곽선, b) 블록 헐 영상의 외곽선, c) 겹친 점

그림 9 최종 블록 헐을 이루는 점 추출

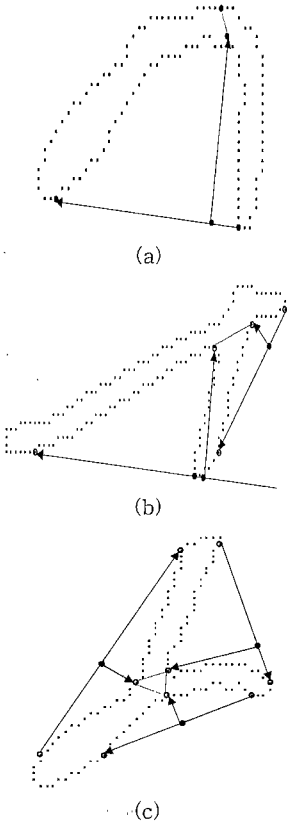
먼저 각 오목 영역에 대해 외곽선상에서 시작하는 점과 끝나는 점을 찾는다. 이를 위해서 부품 영상과 블록 헐 영상에서 외곽선을 구하고(그림9(a)와 (b)), 이 두 외곽선 모두에 존재하는 점을 구한다(그림9(c)). 이런 겹친 점으로 구성된 외곽선에는 여러 곳의 끊긴 지점이 있는데 이들 각각이 하나의 오목 영역에 해당하고 끊긴 지점의 양 끝점을 오목 영역의 시작점과 끝점으로 사용할 수 있다.



오목 특징점 조건:
 $length(\vec{l}_i) \geq T_5$ AND $maximum(\vec{l}_i, C_j) \geq T_6, \dots, (9)$

그림 10 오목 영역이기 위한 조건

그림10은 이렇게 얻어진 네 개의 벡터 \vec{l}_i 를 보여준다. 이제 벡터 \vec{l}_i 를 이용하여 각 오목 영역에서 분할 점을 생성할지 여부와 생성하는 경우 분할 점 좌표를 계산한다. 우선 \vec{l}_i 의 길이를 계산하고, 길이가 임계값 T_5 보다 작으면 분할에 관하여 의미 없는 오목 영역이라 간주하고 무시한다. 실험에서는 T_5 로 7을 사용하였다. 앞의 검사를 통과한 오목영역에 대해서(즉, \vec{l}_i 길이가 임계값보다 큰 오목 영역에 대하여), 벡터의 시작점과 끝점 사이에 있는 외곽선상의 점들에서 \vec{l}_i 까지의 거리를 계산한다. 이들 중 가장 큰 거리 값을 갖는 점이 오목 영역에서 가장 깊은 곳에 위치한 점이다. 이 거리 값이 임계값 T_6 보다 작으면 분할에 관하여 의미 없는 오목 영역이라 간주하고 무시한다. 실험에서는 T_6 로 5를 사용하였다. 그렇지 않으면 이 오목 영역에 대해 가장 깊은 점을 오목 특징점(concave feature point)으로 생성하고 이를 분할 기준 점으로 사용한다.



a) B-점점, b) T-점점, c) 결합 부품
그림 11 오목 특징점 개수에 따른 분할

그림10의 경우 $\vec{l}_1, \vec{l}_2, \vec{l}_3$ 는 무시되고, \vec{l}_4 에서 오목 특징점 C_j 가 생성된다. 오목 특징점 개수에 따라 세 가지 경우로 압축할 수 있는데 그림11이 이들을 예시하고 있다. 오목 특징점이 하나일 때는 B-점점으로 분할하고(그림 11(a)), 두 개일 때는 T-점점으로 분할하고(그림11(b)), 그리고 세 개 이상일 때는 여러 획들이 한 지점에서 갈라지는 지점으로 보고 결합 부품으로 분할한다(그림11(c)). 오목 특징점들이 두 개 또는 세 개일 때는 탐지한 오목 특징점들간의 연결로 분할이 이루어진다. 하지만 한 개일 때는 탐지한 오목 특징점과 쌍을 이룰 획 반대편에 있는 볼록 특징점을 찾는 과정이 필요하다. 이 볼록 특징점은 오목 특징점으로 들어오는 벡터를 연장하여 획 반대편의 점을 구한다.

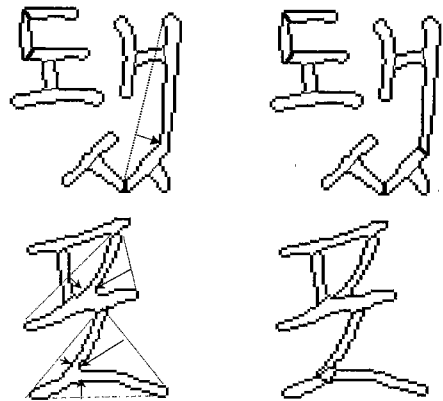


그림 12 오목 특징점 탐지 및 최종 분할

그림12는 모양 분해 알고리즘의 최종 분할 결과를 보여준다. 첫 번째 단계에서 분할하지 못한 부분을 두 번째 단계에서 완벽하게 분할했다. '됐'자의 중성과 중성 부분에서는 오목 특징점을 하나 탐지하여 B-점점으로 분할이 이루어졌고, '포'자의 초성에서는 두 개의 오목 특징점을 탐지하여 T-점점으로 분할이 이루어졌으며, '포'자의 중성에서는 세 개의 오목 특징점을 탐지하여 결합 부품으로 분할이 이루어짐을 볼 수 있다.

5. 실험

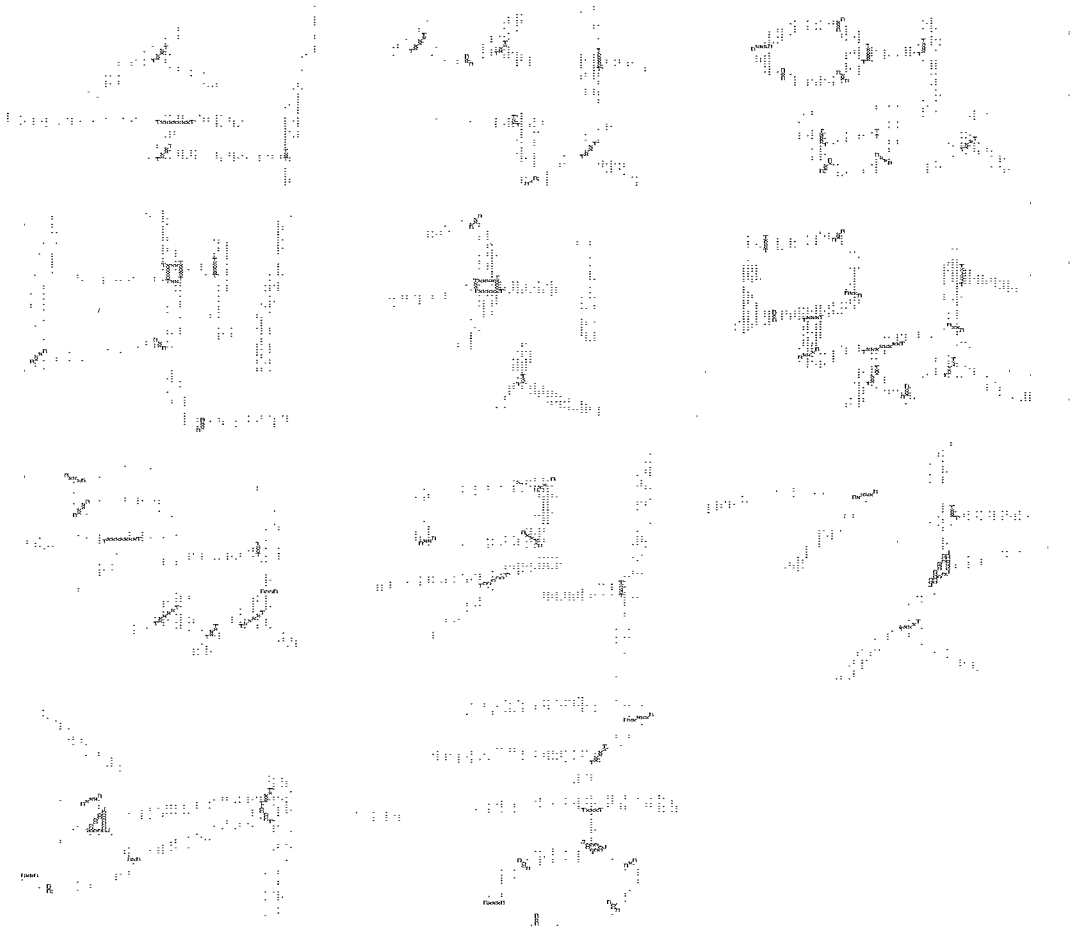
제안한 알고리즘은 Pentium II에서 Visual C++로 구현 하였으며, PE92 데이터 베이스로 실험하였다[25]. PE92 데이터 베이스는 2,350자 각각에 대해 100 샘플씩 총 235,000자를 수집하여 구축하였다. 그림 13은 제안한 알고리즘을 펠기 한글 패턴에 적용한 결과를 보여 주고 있다.

PE92 데이터 베이스에 있는 패턴에 대해 적용하여, 그 중 11개 패턴에 대한 결과를 제시하였다. 그림에서 T-접점을 구성하는 특징점은 T, B-접점을 구성하는 특징점은 B, 그리고 결합 부품을 구성하는 특징점은 J로 표시하였고, 첫 번째 단계에서 결정된 T-접점과 B-접점에 대해서는 분할 경로를 x로 표시하고, 두 번째 단계의 T-접점, B-접점, 결합부품에 대해서는 분할 경로를 o로 표시하였다.

11개 중 앞에서 제시한 6글자, ‘쉬’, ‘쌌’, ‘없’, ‘헛’, ‘곶’, ‘뿔’은 첫 번째 단계에서 분해가 완료되었다. 모두 유사 블록 부품으로 분해되었음을 볼 수 있다. 또한 모든 패턴에서 최소 집합(minimum set)으로, 즉 분해된 부품 두 개를 하나로 합치면 유사 블록 조건이 더 이상 만족되지

않게 분해되었음을 알 수 있다. ‘쉬’는 4개의 T-접점만으로, ‘쌌’은 5개의 T-접점과 2개의 B-접점, 그리고 ‘없’은 5개의 T-접점과 6개의 B-접점으로 분해되었다. ‘없’에서 ‘o’ 경우에는 B-접점 4개로 분해되었는데, 실험결과 ‘o’은 대부분 4~6개의 B-접점으로 분해되었다.

우리가 사용한 모양 분해 방법은 입력된 원래 패턴을 구성하는 획을 아무 손상 없이 명시적으로(explicitly) 분해하여 유사 블록 부품으로 만들어 놓기 때문에, 각 부품에서 특징을 추출하는 것이 용이하다. 예를 들어, ‘o’의 경우 각 부품의 안쪽 외곽선과 바깥쪽 외곽선의 곡률을 계산하고, B-접점으로 분할되어 연결되어 있는 부품들에 대해 곡률을 이용한 조건을 검사하여 ‘o’을 인식할 수 있다. 이



x : 알고리즘의 첫 번째 단계에서 분할된 접점 o : 알고리즘의 두 번째 단계에서 분할된 접점
 (T: T-접점의 특징점, B: B-접점의 특징점, J: 결합 부품의 특징점)

그림 13 모양 분해 결과

러한 점이 제안한 알고리즘의 장점 중의 하나이다. ‘벤’과 ‘긱’자에서 획이 교차한 지점은 네 개의 T-접점으로 분할되었다. 이러한 부분은 쉽게 결합 부품으로 찾아 낼 수 있어, 이웃한 획들을 연결하는 지점으로 사용할 수 있다.

그림13의 뒤에 제시된 5개 글자. ‘뿔’, ‘뿔’, ‘갓’, ‘혀’, ‘콩’은 두 번째 단계까지 수행하여 분해를 완료하였다. 모든 패턴이 최소 집합의 유사 블록으로 분해되었다. ‘뿔’ 패턴은 첫 번째 단계를 마친 후 오른쪽 수직 획과 받침이 만나는 지점에서 미 분할이 발생하였는데, 두 번째 단계에서 하나의 오목 특징점이 탐지되어 B-접점으로 분할되었다. ‘뿔’ 패턴 경우는 ‘ㄱ’에서 미 분할이 발생하였는데, 이 현상은 수평 획과 수직 획이 비스듬이 만나 한쪽에서 특징점 추출이 실패하였기 때문이다. 두 번째 단계에서 두 개의 오목 특징점이 탐지되어 T-접점으로 분할되었다. ‘갓’ 패턴 경우는 수직획과 받침이 만나는 지점에서 미 분할이 발생하였다. 두 번째 단계에서 세 개의 오목 특징점이 탐지되어 결합 부품으로 분할되었다. ‘혀’ 경우는 결합 부품 1개와 T-접점 1개로 분할되었고, ‘콩’은 결합 부품 1개로 분할되어 유사 블록 부품으로의 분해가 완료되었다.

PE92의 패턴들에 대해 실험한 결과 약 47%의 샘플은 첫 번째 단계에서 분해가 완료되었고, 나머지 53%는 두 번째 단계까지 수행하여 분해가 완료되었다. 표2는 실험결과로 얻은 몇 가지 통계값을 보여주고 있다.

한글 6형식 각각에 대해 세자씩 선택하고, 이들 18개 글자 각각에 대해 PE92 데이터베이스의 100개 샘플을 사용하여 총 1,800자에 대해 임계값(T_5, T_6)의 조절에 따른 부품 추출 개수를 관찰하여 표3에 제시하였다. 표3에서 최적의 부품 개수는, 정자체 글자에서 크게 뺀 직선 성분을 하나의 부품으로 하여 그 글자를 나타내는데 필요한 최소 부품 개수를 의미한다. 예를 들어 ‘가’는 ‘ㄱ’ 두개 부품과 ‘丨’ 두개 부품이 필요하기 때문에 총 네 개의 부품으로 이

루어진다. ‘ㅇ’과 같이 굴곡 성분으로 이루어진 패턴은 4개의 직선 성분으로 이루어진다고 가정하였다.

표 2 분해에 관련한 평균값

	첫 번째 단계에서 탐지(개수)		두 번째 단계에서 탐지(개수)			최종(개수)				
	T-접점	B-접점	T-접점	B-접점	결합 부품	T-접점	B-접점	결합 부품	부품 개수	
첫 번째 단계에서 분해 완료된 패턴	47%	3.50	2.60	0	0	0	3.50	2.60	0	8.63
두 번째 단계에서 분해 완료된 패턴	53%	2.95	2.65	0.43	0.82	0.13	3.38	3.47	0.13	9.14

표 3 임계값 조절에 따른 분해 결과

형식	날자	최적의 부품 개수	임계값 조절에 따른 평균 부품 개수		
			$T_5=3, T_6=1$	$T_5=5, T_6=3$	$T_5=7, T_6=5$
1	‘가’ ‘메’ ‘터’	4	9.13	5.18	4.40
		7	14.05	8.11	6.94
		6	12.77	7.13	6.42
2	‘노’ ‘쑤’ ‘푼’	4	8.92	4.83	4.26
		6	13.51	7.24	6.34
		7	16.19	8.48	7.61
3	‘뿔’ ‘화’ ‘퀴’	7	14.30	8.43	7.37
		8	16.14	9.96	7.74
		7	16.00	8.92	7.57
4	‘랏’ ‘뿔’ ‘갓’	9	16.11	9.37	8.13
		11	21.34	13.3	10.93
		8	16.89	9.44	8.51
5	‘옥’ ‘뿔’ ‘을’	8	15.17	9.21	7.26
		11	21.06	12.7	10.48
		10	17.07	10.01	8.05
6	‘쉽’ ‘괘’ ‘뿔’	9	19.92	11.4	9.53
		9	17.99	11.05	9.61
		16	26.98	15.8	13.02

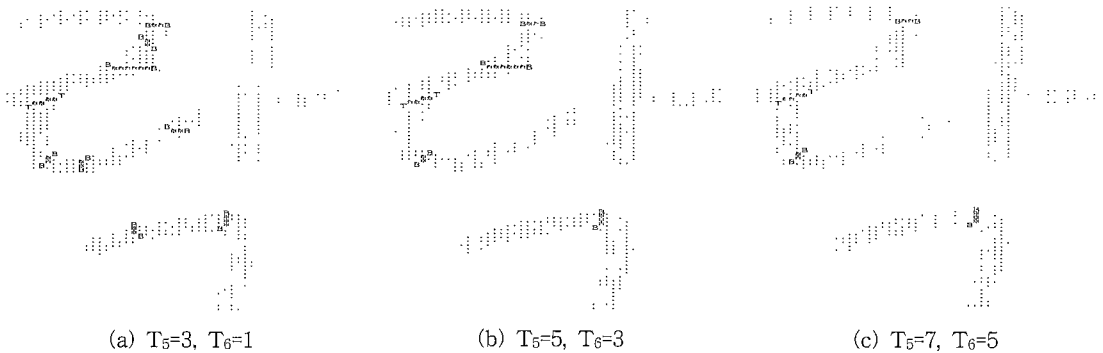


그림 14 임계값 조절에 따른 분할

실험결과 임계값 $T_5=5$, $T_6=3$ 으로 설정했을 때 최적의 부품 개수를 출력함을 알 수 있었다. 임계값을 작게하면 그림14(a)와 같은 과분할(over segmentation)이 발생하고, 임계값을 크게하면 그림14(c)와 같은 미분할(under segmentation)이 발생한다.

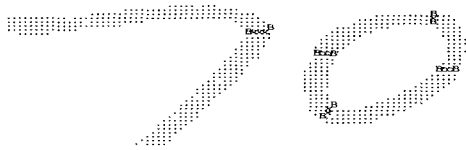


그림 15 인간의 지각적 판단에서 유사 블록에 어긋나는 경우

그림15는 인간의 지각적인 판단으로 유사 블록(near convex)에서 벗어나 보이는 분해 경우를 보여주고 있다. '7'의 경우 수평으로 뻗은 부품과 '9'의 경우 네 개 부품 모두가 유사 블록이 아닌 걸로 볼 수 있다. 이러한 경우는 알고리즘의 임계값(두 번째 단계의 임계값 T_4 , T_5 , T_6)을 조정하면 더 작게 분해하여 인간의 지각적 판단에 대하여 유사 블록하게 만들 수 있다. 하지만 더 작은 부품으로 분할하게 되면 과분할에 따른 문제점이 발생하고, 인식기 등 다음 처리 단계에 대해 부담으로 작용할 수 있다.

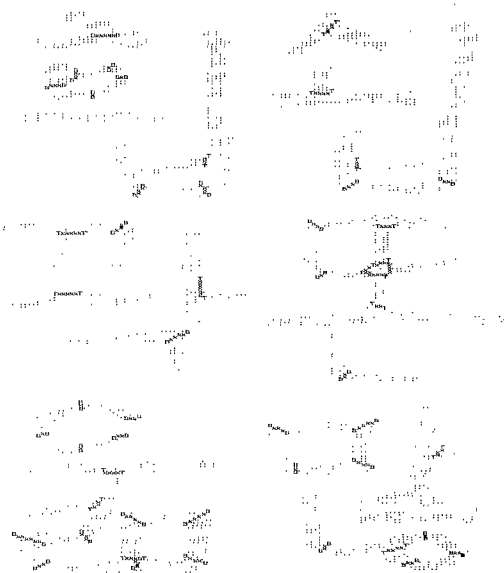


그림 16 자모의 미분할

그림16는 자모 단위로 분할이 이루어지지 않는 경우이다. '쉽'과 '휩' 패턴의 경우는 오른쪽 수직 획과 받침의 획이 일직선으로 만나기 때문에 분할에 필요한 특징점이 첫 번째와 두 번째 단계 모두에서 탐지되지 못해 자모간 미 분할이 발생하였다. 실제로 이런 경우가 종종 발생한다. 이런 경우는 글자에 대한 지식을 사용하는 인식 단계에서 해결해야 한다. '꽂'과 '똥'의 경우는 쌍 자음에서 미 분할이 발생하였다. 앞에서 설명한 바와 같이 각 부품을 명시적으로 가지고 있으므로, 각 부품에서 다양한 특징(면적, 길이, 곡률등)을 쉽게 계산할 수 있으므로 이러한 작은 오차는 인식 단계에서 흡수 할 수 있을 것이다.

6. 결론

필기 한글 모양을 분석하고, 이에 적절한 모양 분해 알고리즘을 제안하였다. 모든 패턴에 대해서 유사 블록 부품들로 분해하였으며, 약간의 예외를 제외하고는 최소 집합이 되도록 분해하였다.

향후 연구 과제로 분할된 패턴으로부터 인식기가 사용할 특징을 추출하는 과정이 남아 있다. 이를 위해서는 분해된 부품과 그들의 연결 관계를 분석하여 골격선을 추출하는 작업이 뒤따라야 한다. 기존의 필기 한글 인식기들이 골격선을 입력으로 받기 때문에, 이러한 인식기에서 골격선 추출하는 모듈만을 대치하여 인식 성능을 평가하는 작업도 중요한 향후 연구과제이다. 또한 모양 분해 알고리즘은 분해된 부품을 명시적으로 출력하므로, 각 부품에서 인식기가 필요로 하는 다양한 특징들을 추출할 수 있다. 이러한 작업을 종합적으로 수행하여 필기 한글 인식기 구축이 가장 중요한 향후 연구이다.

참고 문헌

- [1] 이성환, 박희선, "한글 인식의 사례연구 : 최근 5년 동안의 연구 결과를 중심으로", 제1회 문자인식 워크샵 발표 논문집, 충북대학교, pp.3-46, 1993.
- [2] 김수형, 정선화, 오일석, "필기 한글 문자의 오프라인 인식에 관한 사례 연구", 한국 정보과학회 98 가을 학술발표논문집(II), 아주대학교, pp. 396-398, 1998.
- [3] H.Y. Kim and J.H. Kim, "Handwritten Korean character recognition based on hierarchical random graph modeling," Sixth International Workshop on Frontiers in Handwriting Recognition, Taejon, Korea, pp.577-586, August 1998.
- [4] 황영섭, 방승양, "제약조건 만족 그래프를 이용한 조각의 조합과 필기 한글 인식", 한국 정보과학회 논문지(B), Vol.25, No.10, pp.1493-1505, 1998.

- [5] 강민오, 조형제, "속성 그래프를 이용한 오프라인 필기체 한글 인식", *한국정보과학회 '97 봄 학술 발표 논문집(B)*, 한림대학교, pp.525-528, 1997.
- [6] W.S. Kim and R.H. Park, "Off-line recognition of handwritten Korean and alphanumeric characters using hidden markov models," *Pattern Recognition*, Vol.29, No.5, pp.845-858, 1996.
- [7] H.J Kim and P.K. Kim, "Recognition of off-line handwritten Korean characters," *Pattern Recognition*, Vol.29, No.2, pp.245-254, 1996.
- [8] I.S Oh, J.S Lee, T.W Yoo, J.S Park, and K.C Hong, "Region-based shape decomposition algorithm for Hangul character patterns," *Sixth International Workshop on Frontiers in Handwriting Recognition*, Taejon, Korea, pp.557-566, August 1998.
- [9] 한동협, 조성배, "오프라인 필기체 한글 영상의 세선화를 위한 지식기반 시스템에 관한 연구", *제10회 영상처리 및 이해에 관한 워크샵 발표 논문집*, 제주, pp.163-168, 1998.
- [10] J.W. Suh and J.H. Kim, "Stroke extraction from gray-scale character image," *Progress in Handwriting Recognition*(Ed. by A.C. Downton and S. Impedovo), World Scientific, pp.593-598, 1997.
- [11] S.W. Lee and Y.J. Kim, "Direct extraction of topographic features for gray scale character recognition," *IEEE Tr. on Pattern Analysis and Machine Intelligence*, Vol.17, No.7, pp. 724-729, July 1995.
- [12] H. Nishida and S. Mori, "Structural analysis and description of curves by quasi-topological features and singular points," *Structured Document Image Analysis*(Ed. by Henry S. Baird, Horst Bunke, Kazuhiko Yamamoto), Springer-verlag pp.139-187, 1991.
- [13] C. Arcelli and L. Serino, "Finding near-convex shapes in complex patterns," *International Workshop on Structural and Syntactic Pattern Recognition*, pp.217-226, 1992.
- [14] H.S. Kim and R.M. Haralick, "A hybrid shape decomposition using hyperquadrics and mathematical morphology," *IEEE International Conference on Image Processing (II)*, Austin, Texas, pp.101-105, 1994.
- [15] J. Xu, "Morphological decomposition of 2-D binary shapes into conditionally maximal convex polygons," *Pattern Recognition*, Vol. 29, No.7, pp.1075-1104, 1996.
- [16] K. Abe, C. Arcelli, T. Hisajima, and T. Ibaraki, "Parts of planar shapes," *Pattern Recognition*, Vol.29, No.10, pp.1703-1711, 1996.
- [17] D.S. Doermann and A. Rosenfeld, "The interpretation and reconstruction of interfering strokes," *Third International Workshop on Frontiers in Handwriting Recognition*, Buffalo, New York, USA, pp.41-50, May 1993.
- [18] G.H. Kim and V. Govindaraju, "A lexicon driven approach to handwritten word recognition for real-time applications," *IEEE Tr. on Pattern Analysis and Machine Intelligence*, Vol.19, No.4, pp.366-379, April 1997.
- [19] C.N. Lee and B. Wu, "A Chinese-character-stroke-extraction algorithm based on contour information," *Pattern Recognition*, Vol.31. No.6, pp.651-663, 1998.
- [20] C.Y. Liou and H.C. Yang, "Selective feature-to-feature adhesion for recognition of cursive handprinted characters," *IEEE Tr. on Pattern Analysis and Machine Intelligence*, Vol.21, No.2, pp.184-191, February 1999.
- [21] S.H. Kim, S. Jeong, and C.Y. Suen. "A lexicon driven approach for off-line recognition of unconstrained handwritten Korean words," *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, Bangalore, India, pp.507-510, September 1999.
- [22] Y.S Hwang and S.Y Bang, "Recognition of a handwritten Korean characters by combining segments using constraint satisfying graph," *Sixth International Workshop on Frontiers in Handwriting Recognition*, Taejon, Korea, pp.515-526, August 1998.
- [23] C.H. Teh and R.T. Chin, "On the detection of dominant points on digital curves," *IEEE Tr. on Pattern Analysis and Machine Intelligence*, Vol. 11, No.8, pp.859-872, August 1989.
- [24] Y. Zimmer, R. Tepper, and S. Akselrod, "An improved method to compute the convex hull of a shape in a binary image," *Pattern Recognition*, Vol.30, No.3, pp.397-402, 1997.
- [25] D.H Kim, Y.S Hwang, S.T Park, E.J Kim, S.H Paek and S.Y Bang, "Handwritten Korean Character Image Database PE92," *IEICE Trans. on Information and Systems*, Vol.E79-D. No.7, pp.943-950, July 1996.



박정선

1998년 전북대학교 컴퓨터과학과 학사.
 2000년 전북대학교 전산통계학과 석사.
 2000년 ~ 현재 전북대학교 전산통계학과 박사과정. 관심분야는 패턴인식, 컴퓨터 비전



홍 기 천

1995년 전북대학교 전산통계학과 학사.
1997년 전북대학교 전산통계학과 석사.
2000년 현재 전북대학교 전산통계학과
박사과정. 관심분야는 패턴 인식, 컴퓨터
비전



오 일 석

1984년 서울대학교 컴퓨터공학과 학사.
1986년 한국과학기술원 전산학과 석사.
1992년 한국과학기술원 전산학과 박사.
1992년 ~ 현재 전북대학교 컴퓨터과학
과 교수. 관심분야는 컴퓨터 비전, 문서
및 문자 인식 등임.