Run-Length Code를 이용한 제약없이 쓰여진 한글 필기체 주소열 분할

(An Approach to Segmentation of Address Strings of Unconstrained Handwritten Hangul using Run-Length Code)

김 경 환 † 윤 정 석 ††
(Gyeonghwan Kim) (Jason J. Yoon)

요 약 대부분의 문자 인식기들이 인식대상 영상이 인식단위로 분할되어있다는 가정아래 개발되고 있으나, 실제 필기한글의 분할에 대한 연구는 미미한 실정이다. 본 논문은 Run-length code를 이용한 능동적인 한글 분할방법을 제시한다. 전처리와 인식단위 분할에 응용할 수 있는, 한글의 구조적 특성을 반영한, 기울기 보정 알고리즘을 제안하고, 필기자들의 일반적인 필기 습관과 한글이 갖는 2차원 구조의 특성을 반영하면서 문자의 접촉점을 적극적으로 찾아내기 위한 기초 함수들과 접촉점들의 분류 방법을 제시한다. 임의의 필기자로부터 수집한 필기 한글 주소열 데이타를 이용해 수행한 실험을 통해, 초과분할을 포함하여, 88.2%의 접촉 문자들을 분리할 수 있었다.

Abstract While recognition of isolated units of writing, such as a character or a word, has been extensively studied, emphasis on the segmentation itself has been lacking. In this paper we propose an active segmentation method for handwritten Hangul address strings based on the Run-length code. A slant correction algorithm, which is considered as an important preprocessing step for the segmentation, is presented. Three fundamental candidate estimation functions are introduced to detect the clues on touching points, and the classification of touching types is attempted depending on the structural peculiarity of Hangul. Our experiments show segmentation performance of 88.2% on touching characters with minimal over-segmentation.

1. 서 론

최근 수년간 하드웨어의 비약적 발달과 컴퓨터를 이용한 문자인식에 대한 관심의 증가에 힘입어 한글 인식에 대한 연구에 상당한 진전이 있어왔다. 그러나 인쇄체 한글인식과 몇몇 제한적 상황에서의 필기체 문자인식을 제외한 대부분의 일반적인 오프라인 필기체 한글 인식의 경우는, 한글이 서양 언어권 문자에 대해서 갖는 패턴조합의 복잡성과 필체 변화의 다양성으로 인해 연구에 어려움을 겪어 왔다.

오프라인 필기체 한글인식의 어려움은 크게 두 가지의 원인으로 요약할 수 있다. 첫째는 높은 인식률을 위한 최소 인식단위 설정과 인식대상 개수의 반비례적 모순관계에 따른 인식의 방법론적 문제이며, 둘째는 한글 특유의 2차원적 글자구성에 따른 구조적 복잡성으로 인하여 정확하게 분할된 인식단위 제공의 어려움을 들 수 있다. 인식방법의 경우 오늘날까지 많은 연구가 진행되어 자소단위 또는 음절단위의 인식과 같은 다양한 해결방안이 제시되어 왔다[1]. 반면, 대부분의 한글 인식기가 완벽한 분할을 가정하고 개발되고 있음에도 불구하고 한글 분할 자체에 대한 관심은 매우 낮은 편이다. 분할의 문제를 최소화하기 위하여 문자 영상의 수집과정에서 인위적인 분리를 위한 장치를 마련하거나[2], 영어로 대표되는 유럽언어권 문자와 한글의 구조적 차이점이 명확함에도 불구하고 지금까지 연구된 한글 분할방법은 한글의 특징을 충분히 반영하지 못한 채 영문자에 대한 분할 기술을 그대로 적용하는 경우가 많았다.

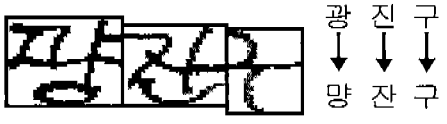
지금까지 한글분할에 관한 연구의 대부분은 필기과정

· 본 연구는 서강대학교 산업기술연구소의 일부 지원을 받음.
† 정 회 원 : 서강대학교 전자공학과 교수
gkim@ccs.sogang.ac.kr
†† 비 회 원 : 런던 City University 전기전자정보공학과
Jeung-Suk.Yoon@bmw.de
논문접수 : 2001년 1월 3일
심사완료 : 2001년 8월 20일

에서 이루어지는 필기도구의 2차원적 진행의 결과로 영문자에 비해 높은 획간 접촉의 문제를 해결하지 못한 채 문자의 강분할을 시도하는 것이 일반적이었다. 예를 들어, 기존의 한글 분할 방법 중에는 영문자와 같은 최소 수직 투영(minimal vertical profile)과 함께 한글 개개 음절이 차지하는 영역이 일정할 것이라는 가정을 이용하여 분할 점을 찾는 방법을 채택해 왔다. 그러나 일반적으로 가정되는 경우와는 달리, 실제로 제약없이 쓰여진 필기체 한글의 경우 수직 투영의 특징이 명확하지 않으며, 음절 영역의 주기성도 그다지 보장받지 못한다. 만일 최소 수직 투영이 명확하다 하더라도 일차원적 구조 특징을 갖는 서양 문자들과는 달리 한글을 강분할하는 경우, 추가적으로 제공되는 정보가 없이는 원하지 않는 분할 부산물들을 포함한 인식단위를 결과로 갖게 될 확률이 매우 높다. 이는 결과적으로 분할된 인식 단위의 모호성을 가중시켜 시스템의 다음 단계 처리과정의 전체적인 성능을 저하시키는 주요 원인으로 작용할 수 있다.

그림 1(a)에서는 최소 수직 투영기법으로 강분할 된 영상이 왜곡되어 인식되는 과정과 결과를 통해 완벽한 인식단위를 얻기 위해서는 한글의 구조적 특징을 충분히 반영할 수 있는 보다 적극적인 방법이 요구된다는 사실을 보여주고 있다. 그림 1(b)에서는 같은 영상에 대한 이상적인 한글분할의 결과를 보여주고 있다. 이러한 완전 분할이 인식과정에서 시스템의 처리속도와 인식성능 두 가지 측면에서 모두 바람직함에도 불구하고 현실적으로 신뢰도 높은 인식기의 도움을 받지 않는 한 완전 분할 시스템을 만드는 것은 쉽지 않은 일이다. 따라서, 본 논문은 이러한 문제를 최소화할 수 있는 방안으로써 한글의 구조적 특성을 충분히 고려한, 초과분할을 인정하는 한글분할 알고리즘을 제안한다.

또한, 일반적으로 문자분할에 중요한 영향을 미치는



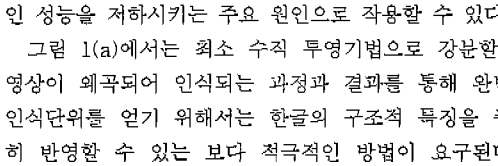
(a) 강분할 된 분할 결과 및 가능한 인식 예

(b) 이상적인 분할 결과

그림 1 문자 분할 결과

전처리단계로 알려진 기울기보정 알고리즘을 Run-length code를 이용해 수행하는 방안을 제안한다. 이 기울기 보정 방법은 영어권 문자에 비해 강한 대각선 성분을 많이 포함하는 한글의 구조적 특성을 반영할 수 있도록 통계적 분석을 통해 개발되었으며 적용결과를 보여준다.

본론으로 들어가기에 앞서 본 논문에서 사용하는 용어와 목적에 대하여 정의한다. 한글은 그림 2에서와 같이 총 19가지의 자음과 21가지의 모음으로 구성되며, 이 자음과 모음을 통틀어 본 논문에서는 자소라 칭한다. 그림 2(c)에서와 같이 자소는 특정한 영역과 형태로 6가지의 조합이 가능하며[3], 이러한 형태에 따라서 2~6개의 자소가 모여 '음절'을 구성한다. 이 때, 자소의 상대적 관계와 위치에 따라서 각기 '초성', '중성' 및 '종성'이라 한다[4]. 본 논문은 필기 문자열로부터 한글 음절을 분리해 내는데 그 목적이 있다.



(a) 자음, (b) 모음, (c) 6 문자 형식의 예

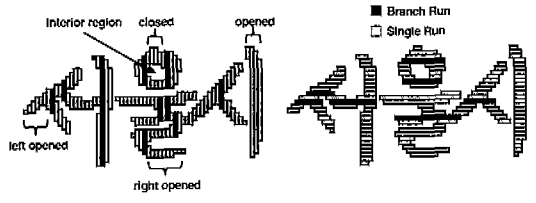
그림 2 자소와 문자 구성 유형

본 논문의 구성은 다음과 같다. 제 2장에서는 Run-length code를 이용하여 한글의 구조적 특징을 기반으로 하는 분할 처리단위의 분류 방법에 대하여 설명하고, 제 3장에서는 구조적 특징을 기반으로 하는 새로운 기울기 보정방법과, 음절 분할을 위한 기초적인 후보 평가 함수들을 정의하고 음절의 접촉형태에 따른 분할 방법을 제시한다. 제 4장에서는 임의의 필기자들로부터 수집된 600개의 한글주소 문자열 영상을 이용한 실험내용과 그 결과를 보이고, 마지막으로 제 5장에서 검토사항과 함께 결론을 맺는다.

2. 한글의 구조적 특징에 따른 한글 분할 접근

한글의 자소들은 천(·), 지(ㅡ), 인(ㅣ)이라고 하는 세 가지 원소들의 결합으로 창제되었다. 즉, 한글 문자열은 결국 강한 수직, 수평 성분들과 작은 점들의 규칙적 반복으로 구성되며, 이는 결국 임의의 한글 음절은 확연한 차이를 갖는 몇 개의 일관된 패턴들의 반복으로 분리해 낼 수 있음을 의미한다. 한글에서의 두 음절의 접촉은 강한 수직 성분과 직교의 형태로 빈번하게 발생

하게 되는데 이는 중성에 사용되는 모음이 일반적으로 강한 수직 성분을 가지며 한 음절 내에서 가장 긴 길이를 차지하기 때문이다. 결국 이는 강한 수직 성분들을 주시하는 경우에 접촉점들을 보다 쉽게 찾아낼 수 있다는 논리의 근거가 된다.



(a) (b)

그림 3 구간(section)의 정의: (a) Vrun, (b) Hrun

본 논문은, 이와 같은 한글의 구조를 고려하여, 보다 체계적인 접근을 위해 Run-length code를 기반으로 하는 기본 분할 처리단위를 정의한다. 한쪽 면에 두 개 이상의 인접한 런을 갖고 있는 런을 '분기 런'으로 정의하는데 그림 3에서 검은 런으로 표시되어 있다. 그림 3(a), (b)는 각각 수직 런길이 코드(Vrun)와 수평 런길이 코드(Hrun) 구조에서 정의하는 3가지 종류의 '구간(section)'을 나타내고 있다. 우선 두 개의 분기 런들 사이에 위치하는 일련의 런 집단을 '닫힌 구간(closed section)'이라 정의한다. 그리고 런 집단의 한쪽 끝이 분기 런이 아닌 경우 이를 '반 닫힌 구간(half closed section)'으로 정의하며, 마찬가지로 양쪽 끝이 모두 분기 런이 아닌 경우를 '열린 구간(opened section)'이라고 칭한다. 반 닫힌 구간은 다시 '오른쪽과 왼쪽(수평 런에 대해서는 위와 아래) 닫힌 구간'으로 구분할 수 있다.

이러한 구획에 대한 정의는 비슷한 성질을 갖는 연속된 런들을 집단으로 다룰 수 있게 됨으로써, 분할에 있어서 일반적으로 후보가 될 수 없는 분기 런들과 열린 구간 등을 쉽게 배제하고 런 집단의 특징에 따른 효과적인 처리가 가능하게 됨으로 결국 전체 시스템의 수행 능력을 높일 수 있게된다.

또한, 구간은 런 집단에 대한 몇 가지 중요한 정보를 전달한다. 첫 번째로 구간은 대부분 같은 방향으로 진행하는 런 집단을 성분으로 하기 때문에(그림 3) 해당 구간에 포함된 획의 정확한 기울기를 구할 수 있다. 이는 제안하는 구간 방법의 도입을 통해 얻을 수 있는 가장 중요한 이득의 하나로서 다음 장에서 이를 이용한 새로운 기울기 보정방법을 소개한다. 두 번째는 전체 문자열에서의 평균 획 두께(average stroke width) 정보를 얻기 위해 각 구간의 평균 두께를 구해 사용하는 것이 가능하다. 각 구간이 동질의 런 성분으로 구성된 점을 감안할 때, 이를 통해 매우 의미있는 획 두께를 추출할 수 있게된다. 마지막으로, 구간 그 자체의 길이는 해당 구간으로부터 구할 수 있는 각종 정보에 대한 가중치로 사용 될 수 있다. 예를 들어 닫힌 구간의 경우에 그 구성 런이 단일 런(single run)이라면 그 구간으로부터 얻을 수 있는 정보는 아무 것도 없다. 또한 이러한 단일 런 구간 또는 상대적으로 의미가 적은 구간을 해석과정에서 배제함으로써 기울기 보정과정에서 발생할 수 있는 역보정(overestimation)을 방지할 수 있다.

3. 기울기 보정 및 음절 분할

3.1 연구 동기

한글에 일반적으로 사용되는 분할 방법은 각 음절 사이의 수직 투영 정보는 항상 최소값을 유지한다는 가정하에 인식기의 도움을 받는 형태로 연구되어 왔다[8]. 그러나 히스토그램이나 투영정보를 이용한 접근 방법은 대부분 오류의 잠재성이 큰 편이다. 특히 영문과 달리 2차원적으로 진행되는 필기형태를 갖는 한글의 경우, 2차원적 특징을 1차원으로 투영하는 접근방법의 신뢰도는 낮아질 수 있다. 그림 1에서와 같이 접촉이 심하지 않은 경우조차 이와 같은 방법은 큰 왜곡을 일으키는 경우가 많다. 또한 분할 단계에서 정확한 분할을 하기 위해 빈번히 인식기의 도움을 요구하는 것은 인식기의 전체 성능을 처리속도 측면에서 크게 떨어뜨리는 결과를 초래할 수 있다.

입력된 문자열의 획간 접촉이 심한 경우 문자 분할은 더욱 어려워진다. 임의의 필기자들로부터 수집된 600개의 한글 주소열에 대한 접촉의 빈도 수 조사 결과 그 중 76.9%가 접촉된 음절을 포함하였다. 또한, 전체 10,449개의 음절 중 13.7%는 1개의 접촉을, 그리고 1.6%는 2개 이상의 접촉을 가졌다. 수집된 한글 주소열에 많은 숫자가 포함되어 있고, 일반적으로 숫자의 접촉 정도는 한글에 비해 매우 낮다는 점을 감안할 때 조사된 접촉률은 실제로 더 높아질 수 있는 여지가 남아있다. 한글주소를 표기하기 위해 사용되는 최소 음절의 수가 10개 이상이라는 점을 생각해 볼 때, 대부분의 우편 주소의 한글 문자열에는 1개 이상의 접촉 음절이 존재한다고 말할 수 있다. 결국 이는 한글의 경우 분할 방법에 따라 인식률이 크게 좌우될 수 있음을 보여주고 있다.

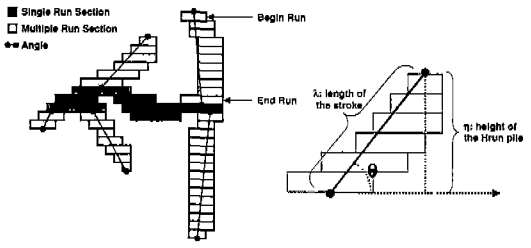
따라서, 본 논문의 목적은 제약없이 필기된 한글 문자열에서 음절의 분할 후보 점들을 찾아내는 것이다. 사용된 문자열은 상당수의 음절 접촉을 갖고 있음을 가정하

고, 최소한의 초과 분할을 인정하는 것을 전제로 알고리즘을 개발하였다.

3.2 기울기 보정

분할 기반 인식 시스템에 있어서 전처리 과정은 매우 중요하다. 전처리 과정은 영상이 입력되는 순간부터 인식의 최종 단계까지 모든 단계에서 잠재적으로 영향을 미치게 된다. 특히 초기 단계에서 잘못 적용된 기울기 보정은 입력 문자열 자체를 사용할 수 없을 정도로 왜곡시키기도 하는데, 이러한 왜곡은 잘못된 분할을 유도하고 결국 인식 실패의 중요한 요인이 된다.

문자인식과 관련된 기울기 보정에 관한 많은 연구들이 오늘날까지 발표되어 왔다. 대부분의 기본적 접근방식은 입력된 전체 문자열에서 직선성분들을 추출한 뒤 그들의 평균 방향성분을 취함으로써 문자열의 평균 기울기를 구하는 것이다[5][6]. 이러한 직선성분을 추출하기 위해 많은 방법들이 연구되어 왔지만, 이미 알려진 대부분의 기울기 보정 방법들이 숫자 및 영문자를 근거로 개발되었기 때문에 원천적으로 다른 구조를 갖는 한글을 포함하는 동양권 문자들의 경우에도 같은 성능을 보장하리라고 기대하기는 어렵다.



(a) (b)

그림 4 구간(section)에서의 기울기 정의: (a) 각 구간의 기울기, (b) 보정을 위한 구간의 길이 계산

앞에서 제안한 구간 방법은 동질의 런 집합을 하나의 단위로 묶어 사용할 수 있게 한다. 그림 4(a)에서 볼 수 있듯이 각 구간의 기울기는 쉽게 예측할 수 있는데 이렇게 구해진 기울기들은 실제 각각의 직선 성분들의 기울기를 잘 반영하고 있음을 알 수 있다. 구간의 기울기는 시작 런(begin run)과 끝 런(end run)의 중점을 잇는 선분의 기울기를 계산하여 구할 수 있다. 실험에서는 이러한 방법을 통해서 일반적으로 구간이 2개 이상의 런을 갖고 있는 경우에 실제 직선 성분의 기울기와 상당히 유사한 결과치를 얻을 수 있었다.

또한, 제 2장에서 설명했던 것과 같이 각 구간의 길이는 앞서 구한 구간의 기울기에 대한 가중치로 사용될 수 있다. 일반적으로 런의 개수로 구간의 길이를 계산했던 단순한 방법과는 달리, 앞서 구한 구간의 기울기를 역으로 이용하여 구간의 길이를 보정할 수 있다. 그림 4(b)는 구간의 길이를 계산하는 방법을 보여주고 있다. 임의의 구간 k에서의 런의 개수, 즉 높이를 η_k, 그리고 수평에 대한 구간의 기울기를 θ_k라고 할 때, 보정된 구간의 실제 길이 λ_k는 식 (1)과 같다.

$$\lambda_k = \eta_k \cdot \mathrm{cosec}(\theta_k), \quad k = 1, \ldots, m \tag{1}$$

여기서, m은 문자열에서의 구간의 개수이며, 전체 문자열의 평균 기울기를 식 (2)와 같이 정의하여, 상대적으로 긴 길이를 갖는 구간에서 구한 기울기가 평균 기울기를 결정하는데 기여하는 정도가 더 클 수 있도록 한다.

$$\theta_{slant} = E[\lambda_k \cdot \theta_k] \tag{2}$$

그런데 여기서 문자열의 단순 평균 기울기가 그 문자열 전체의 실제 기울기를 나타내는 것인가에 대한 의문을 가질 수 있다. 그림 5는 구간 방법을 통한 영문과 한글에서의 각 구간 기울기의 분포를 도시화 한 것이다. 그림에서 살펴볼 수 있듯이 영문의 경우를 살펴보면 기울기는 0을 중심으로 거의 완벽한 대칭 분포를 갖고 있는데 반해, 한글은 한쪽으로 약간 치우쳐져 있음을 확인할 수 있다. 이는 영문은 대부분이 곡선 성분으로 이루어져 있으며 특정 방향으로의 획 성분이 과다하게 포함되어 있지 않음에 반해, 한글은 통계적으로 대각 획, 특히 오른쪽 위에서 왼쪽 아래로 향하는 성분을 많이 포함하고 있기 때문이다. 따라서 한글 문자열의 기울기를 영문과 동일한 방법을 통해 구한다면 필연적으로 잘못된 결과를 얻을 수밖에 없음을 알 수 있다.

이와 같이 경험적으로 얻은 지식을 바탕으로 본 논문에서는 보다 개선된 기울기 보정법을 다음과 같이 제안한다. 그림 5에서 볼 수 있듯이 영문과 한글 모두 기울기의 분포는 정규 분포에 가까운 양상을 띄고 있다. 우리는 구간 기울기의 발생 빈도와 필기자의 필기 습관을 반영하기 위하여, 실제 분포에 근접하는 가상의 정규 분포(그림 5)를 이용하여 수식 (2)를 수식 (3)과 같이 확장할 수 있다[7].

$$\theta_{slant} = E[\lambda_k \cdot \theta_k] = \sum \lambda_k \cdot \theta_k \cdot f(\theta_k) \tag{3}$$

이때 사용된 확률 밀도 함수 f는,

$$f(\theta_k) \approx \mu \cdot e^{\nu[\theta_k - \mu_\theta]^2} \tag{4}$$

로 나타내어진다. 여기서 μ와 ν는 임의의 상수를 뜻하며, μ_θ는 문자열 전체에 대한 구간의 단순 평균 기울기를 의미한다.

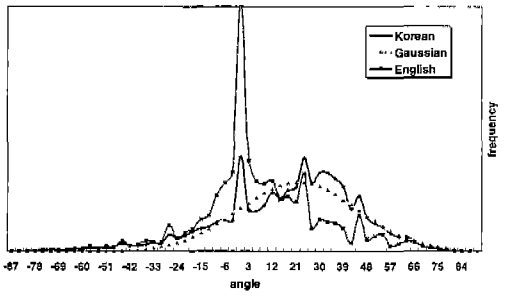
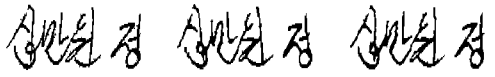


그림 5 구간(section) 기법에 따른 기울기 각도의 분포

제안된 방법은 모든 한글 문자열에 대해 기존의 방법보다 좋은 결과를 나타낸다. 그림 6에서는 기존의 방법과 제안된 방법에 대한 비교 영상을 보여준다. 한글 필기영상에서 자주 볼 수 있는 강한 대각선 성분으로 인해 역보정 될 수 있는 상황을 잘 처리하고 있음을 확인할 수 있다.



(a) (b) (c)

그림 6 기울기 보정 결과 비교: (a) 보정 전의 영상, (b) 일반적인 보정 결과, (c) 제안한 방법에 의한 보정 결과

3.3 음절분할

영문 분할에 일반적으로 사용되는 분할 방법 중의 하나는 입력된 문자열을 적정 수준으로 초과 분할한 후 필요에 따라 적절한 시기에 인식기의 도움을 받거나 인식과정에서 재 조합해 내는 것이다. 이 방법은 인식기의 도움 없이 정확한 분할 후보 점을 찾아야하는 어려움을 극복하고, 분할 및 인식과 관련된 문제들을 독립적으로 처리하기 위한 방법이다. 이러한 선분할/후조합의 방법을 통하여 영문의 경우 접촉 문제를 효과적으로 풀어낼 수 있었으며[9], 한글의 경우도 분할기로부터 적당하고 일관성있는 형태의 분할 단위를 제공 받을 수 있다면 영문 분할에서와 같은 접근방법을 적용시킬 수 있을 것이다.

본 논문은 적극적인 분할을 수행하기 위하여 음절 사이의 접촉들에서 발생하는 특징을 분류하여 분할 후보점을 찾는 데에 적용하였다. 그 중 특이할 만한 특징들은 다음과 같다. 1) 두 개의 직선획이 큰 각을 이루며 접촉하는 경우 대부분 급격한 획 넓이와 방향의 변화를 관찰할 수 있다. 2) 문자의 끝 부분은 필기자의 습관에 따라 일반적으로 획 넓이가 가늘어지며 진행되는 경향이 있다. 3) 마지막으로 미약한 접촉이 발생하는 경우는 접촉부에서 최소의 획 넓이를 발견할 수 있다.

이러한 세 가지 특징을 갖는 분할 후보들을 추출해 내기 위하여 다음의 세 가지의 기초 함수들이 제공된다. 각 함수들은 독립적으로 사용되지 않고 적당한 조합을 통해 후보들을 추출한다. 이들은 인접한 두 개의 런에 대해 작용하여 특정 값을 되돌려주며, 그림 7에서는 각 함수들에 대한 설명을 보여주고 있다. 각 함수에 대한 정의는 다음과 같다.

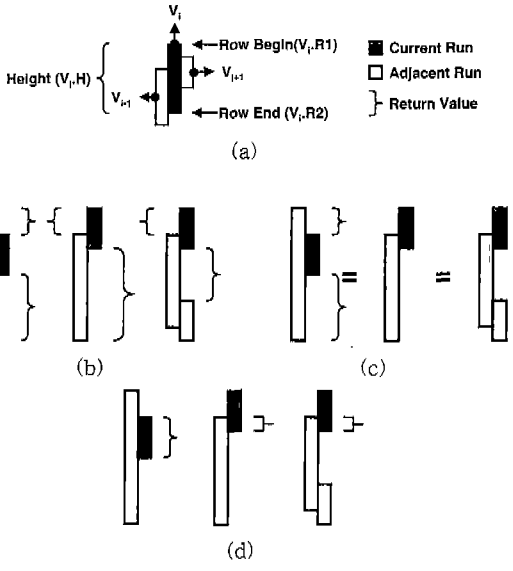


그림 7 세가지 기본 함수의 정의: (a) i번째 Vrun, (b) Discreteness, (c) Difference, (d) Overlap

Discreteness function: 이 함수는 인접한 두 개의 수직 런이 얼마나 급격한 변화를 보이는 가를 나타낸다. 그림 7(a)에서 정의하고 있듯이 V_i를 i번째의 수직 런으로 가정하고 V_i.R1과 V_i.R2를 각각 그 수직 런에서의 시작 런과 끝 런이라고 할 때, 그림 7(b)에서 보이는 discreteness function $d(V_i, V_{i-1})$는 식 (5)와 같이 정의한다.

$$d(V_i, V_{i-1}) = |V_i.R1 - V_{i-1}.R1| + |V_i.R1 - V_{i-1}.R1| \quad (5)$$

Difference function: $V_i.H$를 i번째 런의 높이라고 할 때 difference function은 식 (6)과 같이 정의된다. 그림 7(c)에서도 볼 수 있듯이 이 함수는 인접한 런들의 높이 차이를 되돌려주며, 런들의 높이가 같은 경우

어떠한 형태에서도 같은 결과값을 보여준다.

$$\delta(V_i, V_{i-1}) = |V_i.H - V_{i-1}.H| \quad (6)$$

Overlap function: 이 함수는 인접한 두 개의 런이 접쳐져 있는 양을 그림 7(d)와 같이 돌려준다. 입력 영상의 기준점을 좌상단으로 할 때 overlap function은 식 (7)과 같이 주어진다.

$$o(V_i, V_{i-1}) = \min(V_i.R2, V_{i-1}.R2) - \max(V_i.R1, V_{i-1}.R1) \quad (7)$$

Nishiwaki와 Yamada는 접촉을 6가지 종류로 분류해서 각각의 특징에 따라 다른 분할방법을 적용함으로써 숫자를 성공적으로 분할해 내었다[10]. 본 논문에서는 이와 같은 분류법을 앞서 제안한 세 유형의 기초 함수들을 사용하여 한글의 구조적 특징을 반영할 수 있도록 개선하였다. 새로 개선된 제안하는 방법에서는 접촉의 유형을 네 가지의 형태로 구분했으며, 모든 구간에 대하여 해당되는 접촉의 종류를 찾는 단계가 진행된다. 하나의 구간 안에 존재하는 최대 접촉 수는 한 개로 가정했으며, 한 구간 내에서의 다중 접촉은 없는 것으로 가정하였다. 실제로 구간은 매우 잘게 나뉘어져서 하나의 구간에 두 개 이상의 접촉을 포함하는 경우는 거의 없었으며, 존재하는 몇몇 경우도 과다한 왜곡이 가해져 회복이 불가능한 경우가 대부분이었다. 그림 8은 각각의 접촉의 유형을 검출하는 방법에 대한 순서도를 수직런을 이용하는 세 가지 유형에 대해서 보여주고 있으며, 그림 9에서는 각각의 접촉에 대한 예를 보여준다.

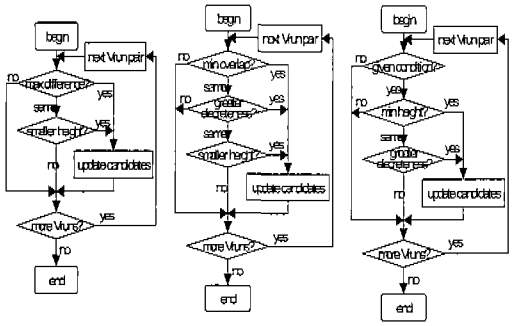


그림 8 세가지 접촉 유형의 검출과정: (a) destination, (b) weak bridge, (c) construction

Destination : T자 형의 접촉을 찾기 위한 알고리즘으로서 discreteness function의 최대 반환 값을 갖는 런들 중 가장 작은 높이를 갖는 런을 반환한다.

Weak bridge : 음접간의 접촉에서 흔히 발생하는 획의 감소 경향을 찾아내는 알고리즘이다. Overlap function을 이용하여 접촉부의 가장 약한 부분을 나타낸다.

Construction : 수직 획이나 원형 획이 깊게 접촉하는 경우 흔히 발생하는 접촉 유형이다. 대부분의 경우 difference function을 이용하여 그 특징 점을 찾아낼 수 있는데, 구간의 시작 런과 끝 런의 높이가 후보 런의 높이보다 커야 하는 조건을 갖는다. 언어의 종류와 필체에 따라 기타의 조건을 덧붙여 보다 정밀한 알고리즘을 수행할 수 있다.

Concavity : 이 알고리즘은 수직 런이 아닌 수평 런 구조에서 수행되며, 구간 전체를 조사하지 않고 두 개 이상의 상위 런을 갖는 모든 후보 런들 중 적당한 조건의 수평 런이 선택된다. 정확한 분할 점을 수직 런에서 추출된 후보 런들과 동기 시키기 위하여, 수평 후보 런을 중심으로 연결 요소(connected component)의 무게중심 등을 조사하여 적당한 비율로 후보 런에서의 분할 점을 찾아낼 수 있다.

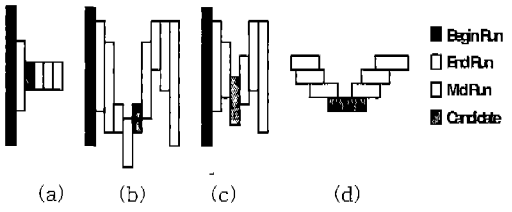


그림 9 네가지 접촉 유형: (a) destination, (b) weak bridge, (c) construction, (d) concavity

제 2장에서 설명했던 세 가지 종류의 구간에 따른 여러 가지 조건들의 활용을 통해 불필요한 수행시간을 줄일 수 있다. 일반적으로 열린 구간의 경우 접촉을 가진 경우는 매우 드물며, 반 닫힌 구간의 경우도 통계적 조건을 신중하게 부여하여 알고리즘을 수행함으로써 최종 후보 점의 개수를 상당히 낮출 수 있다.

본 논문과 비슷한 목적으로 최근에 발표된 두 편의 연구논문들[11][12]과 접근방법의 유사성과 차이점을 정리하면 다음과 같다. 한글 인식을 위한 획 성분의 추출이 인식기에 종속되는 경우 발생할 수 있는 문제점을 최소화하기 위해 인식방식에 대해 가능한 독립적으로 동작할 수 있는 분할 방법을 제안하고 있다. 그리고 필기 한글이 갖는 구조적 특성을 분할 과정에 반영하기 위해 노력하고 있음을 볼 수 있다. 또한, 필기문자 영상의 해석과정에서 작용할 수 있는 방해요소들을 최소화

하기 위해 널리 사용되는 세선화 및 직선근사 방법을 적용하지 않고, 필기영상의 원형을 가능한 최대한 유지하면서 분할 할 수 있는 방법을 공통적으로 제시하고 있다. 그러나, 이 비교 논문들은 분할의 대상이 이미 독립적으로 분할된 음절을 자소단위로 구분하기 위한 방법을 제안하고 있는 반면, 본 논문에서는 더 복잡한 접촉 양상을 포함하는 문자열의 분할을 목적으로 하는 알고리즘을 제안하고 있다. 실제, 우편주소 해석과 같은 응용분야에서는 실험실 환경과는 달리 음절간 접촉을 쉽게 발견할 수 있고, 인식단계 이전에 접촉된 음절에 대한 조치가 필요하기 때문이다.

4. 실험 결과

본 장에서는 임의의 필기자로부터 얻은 600개의 한글 주소 문자열 영상에 대하여 제안하는 분리 방법을 적용한 실험결과에 대하여 서술한다. 실험과정 중 복구할 수 없는 스캐닝 왜곡을 포함하는 두 영상을 제외시켰고, 음절 분할된 최종인식 단위가 인식기에서 오인될 소지가 없는 정도의 음절로 구분될 수 있는지를, 제공된 문자열과 음절의 개수를 시각적으로 검사하여 확인하였다.

수집된 문자열들의 문자열 당 평균 음절수는 17.5개였으며, 문자열 당 평균 연결성분의 개수는 27.5 였다. 또한 이 문자열들 중 29.6%가 숫자, 그리고 2.9%가 영문을 포함하고 있고 한글의 2차원적 조어법에 의해 음절 내에서의 접촉이 음절간의 접촉에 비해 훨씬 빈번하게 나타날 수 있음을 생각해볼 때, 실제로 한글 위주로 구성된 문자열에서는 각 음절 당 연결 성분의 개수는 실험영상을 통해 구한 1.57에 비해 훨씬 높아질 수 있음을 예상할 수 있다.

표 1에는 음절 분할 전후의 음절에 대한 연결 성분수의 변화를 나타내고 있다. 분할 전의 1.57배에서 분할 후의 3.92배로 약 2.5배 정도 연결성분이 증가했음을 확인할 수 있다. 이는 초과분할을 가정하고 분할을 수행했기 때문인데, 한글 음절 하나가 2~6개의 자소로 이루어져 있고 자소 간의 접촉이 상당히 빈번하게 발생할 수 있음을 고려해 볼 때 납득할 만한 수준의 초과 분할로 간주할 수 있다.

표 1 음절 당 연결성분 수

음절 개수	분할 전	분할 후
10,449	16,366(157%)	41,011(392%)

표 2는 음절에 대한 분할 성능을 보여준다. 수집된 자료에서 총 10,449개의 음절 중 1,427(13.7%)개의 단일 접촉과 164(1.6%)개의 다중 접촉을 발견할 수 있었는데 결국 전체의 15%정도의 음절에서 접촉이 발생하는 매우 높은 접촉률을 보여주었다. 표 2에서는 접촉 수에 따른 음절 분할 성공률을 나타내고 있는데 단일 접촉의 경우 90%정도의 높은 분할 성공률을 보였으며, 다중 접촉의 경우에도 74%에 대하여 분할에 성공할 수 있었다.

표 2 한글 음절에 대한 분할 성능

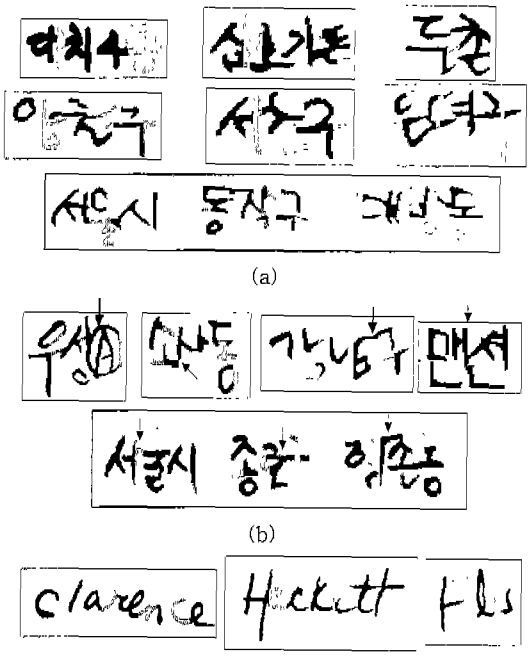
	단일 접촉	다중 접촉	총합
접촉 수	1,427	164	1,591
성공적 분할 수	1,283(89.9%)	141(73.8%)	1,404(88.2%)

마지막으로, 앞서 수행한 실험을 음절 기준이 아닌문자열에 적용하여 표 3과 같은 결과를 얻을 수 있었다. 전체 문자열 중 476개가 접촉을 포함하여 앞서 밝혔던 문자열 당 평균 음절 수(17.5)와 접촉을 포함하는 음절의 비율(15.3%)을 감안해 볼 때 통계적으로 문자열 당 2.7개 정도의 접촉을 포함하고 있고, 이는 전체 시험 문자열의 80%에 해당하는 비율이다. 그리고 접촉을 포함하는 476개의 문자열 중 24%정도에 해당하는 115개의 문자열이 다중 접촉을 포함하고 있다. 다중 접촉에 대한 분리율이 73%정도인 상황에서 다중 접촉이 존재하는 문자열에 대한 완전 분리율은 64% 정도로 나타났는데 이는 한 문자열에 복수의 다중 접촉이 자주 발생하고 있음을 수치적으로 보여주고 있다. 따라서, 상대적으로 낮은 분리율을 보여주는 다중 접촉에 대한 보다 근본적인 대안 없이는 완전한 형태의 인식 단위 제공이 어렵다는 것을 확인할 수 있다. 실험에 사용된 문자열에 대한 문장의 완전 분리율이 74.2%로서 높은 결과치를 나타내었다.

표 3 한글 문자열에 대한 분할 성능

	접촉이 있는 경우			총합
	단일 접촉	다중 접촉	총합	
문자열 수	473	115	476	598
성공적 분할 수	354(74.8%)	74(64.3%)	326(68.5%)	448(74.2%)

그림 10(a)에서는 제안된 알고리즘을 적용하여 음절 분할이 성공적으로 수행된 실험 결과들을 볼 수 있으며, 그림 10(b)에서는 음절 분할에 실패한 몇 가지 경우를 나타내었다. 분할에 실패한 경우의 대부분은 심한 접촉으로 인해 추가적인 복원과정 없이는 완전한 상태의 음절을 만들어 낼 수 없는 경우들이었다. 또한 수직 런 구간을 중심으로 분할 후보점 검색을 수행하였기 때문에, 수직으로 진행된 접촉은 근본적으로 구분하기 어려웠다. 그리고 같은 넓이의 수평 획의 접촉과 같이 특이점을 찾기가 어려운 접촉도 있었다. 그림 10(c)에서는 제안된 알고리즘을 영문에 적용시킨 결과를 보여주고 있는데, 전체적 분할률은 영문 전용 분할기에 비해 상대적으로 낮지만 기존의 영문 분할기에서 문제가 되었던 몇 가지 경우에 대해 좋은 분할 결과를 보임으로써 다중 언어 분할기로서의 적용도 기대할 수 있다.



(a)

(b)

(c)

그림 10 분할 결과:(a) 성공한 경우, (b) 실패한 경우, (c) 영문에 적용한 경우

제 4장에서 언급한 최근의 논문들[11][12]과의 성능 비교는 다음과 같이 정리될 수 있다. 기본적으로 본 논문이 강조하고 있는 필기 문자열의 분할이 비교논문들이 목적하고 있는 분리된 음절의 자소분리와 차이가 있기 때문에 객관적인 비교는 불가능하다. 그러나, 비교 논문 [11]에 제시된 89.5%의 분리율은 본 논문에서 얻은 89.9%(단일 접촉) 및 88.2%(평균)와 비슷한데, 본 논문에서는 한 개 이상의 음절을 포함하는 문자열을 입력으로 가정하여 얻은 결과이기 때문에 제안하는 알고리즘의 유효성을 쉽게 확인할 수 있다.

5. 결 론

제약없이 필기된 한글 우편 주소열에 대한 적극적인 음절 분할방법에 대해서 소개하였다. 기존의 한글 분할 방법의 분할 오류 문제를 개선하기 위하여 한글의 특징과 필기자의 습관을 적극적으로 반영할 수 있는 방법에 연구의 초점을 맞추었다. 이를 위해 한글의 구조적 특성을 반영한 몇 가지의 유용한 함수들과 알고리즘들이 제안되었으며, Run-length code를 기반으로 하는 새로운 구조체를 도입함으로써 기존의 분할 방법들이 가진 공통적 문제점들을 상당히 해결할 수 있음을 확인하였다. 또한, 분할에 중요한 영향을 미치는 기울기 보정방법을 Run-length code를 이용해 수행할 수 있는 방안을 제시하였고 실험을 통해 제안된 방법의 유효성을 검증하였다.

제안된 방법의 또 다른 특징은 제안된 함수의 적절한 조합과 몇몇의 제한 조건을 도입함으로써 다른 필체나 필기자집단 및 다른 언어에도 적용할 수 있는 가능성을 검증하였다(그림 10). 실제로 Nishiwaki와 Yamada는 비슷한 방법으로 숫자의 능동적 분할을 성공적으로 수행할 수 있었다[10]. 또한 인식기의 도움을 최대한 줄이면서 분할기의 수행을 최대한 독립시켜 인식기의 부담을 최대한 줄일 수 있도록 노력하였다.

제안된 알고리즘을 개선하기 위하여 분할 후 재조합 방법에 대한 지속적인 연구가 뒤따라야 하는데, 이 과정은 인식기 내부에서 수행되며 인식기의 구조 및 특성을 고려하여 효율적으로 수행될 수 있도록 고려하여야 한다. 제약없이 쓰여진 필기 문자열에서 음절간 경계를 발견하기 어렵고 이를 극복하기 위해 초과분할을 인정하는 분할방법을 적용하는 것이 일반적인 접근방법이다. 이 경우 인식과정에서 여러 조합 가능성에 대해 검증이 수행되어야 하는데, 재조합의 대상이 많을수록 인식기의 처리속도와 인식율이 크게 영향을 받기 때문이다.

참 고 문 헌

[1] 정선화, 김수형, "과다 분리 및 사전 후처리 기법을 이용한 한글이 포함된 무제약 필기체 문자열의 오프라인 인식", 한국정보과학회 논문지, 제26권, 제5호, pp.

647-655, 1999.
[2] 김수형, "최소거리 분류 및 사전기반 후처리의 강결합에 의한 필기 한글 주소열의 인식", 한국정보과학회논문지(B), 제25권, 제8호, pp. 1195-1205, 1998.
[3] 김민기, 권오성, 권영빈, "모음의 구조적 형태와 조합 규칙에 충실한 한글 문자의 유형분류", 한국정보과학회논문지(B), 제25권, 제4호, pp. 686-695, 1998.
[4] S.-W. Lee and E.-S. Kim, "Efficient post processing algorithms for error correction in handwritten Hangul address and human name recognition," *Pattern Recognition*, vol. 27, no. 12, pp. 1631-1640, 1994.
[5] D. Guillevic, *Unconstrained Handwriting Recognition Applied to the Processing of Bank Cheque*, PhD thesis, Dept. of Computer Science, Concordia University, 1995.
[6] S. Madhvanath, G. Kim and V. Govindaraju, "Chaincode Contour Processing for Handwritten Word Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 928-932, 1999.
[7] H. Stark and J. W. Woods, *Probability, Random Process and Estimation theory for Engineers*, Prentice Hall, 1994.
[8] 황순자, 김문현, "자소 클래스 인식에 의한 off-line 필기체 한글 문자 분할", 한국정보처리학회 논문지, 제3권, 제4호, pp. 1002-1013, 1996.
[9] G. Kim and V. Govindaraju, "Handwritten phrase recognition as applied to street name images," *Pattern Recognition*, vol. 31, no. 1, pp. 41-51, 1998.
[10] D. Nishiwaki and K. Yamada, "Holistic Recognition of Touching Digits," *In Proc. of 6th International Workshop on Frontiers in Handwriting Recognition(IWFHR VI)*, Taejon, Korea, pp. 359-377, August 1998.
[11] 곽후근, 최영우, 정규식, "모음 구조와 경험적인 규칙을 이용한 필기된 한글의 자소 분리 방법", 한국정보처리학회 논문지, 제8권, 제1호, pp. 10-19, 2001.
[12] 박정선, 홍기천, 오일석, "필기 한글 문자의 모양 분해", 한국정보과학회 논문지:소프트웨어 및 응용, 제28권, 제7호, pp. 511-523, 2001.

김 경 환
정보과학회논문지 : 소프트웨어 및 응용
제 28 권 제 6 호 참조

윤 경 석
1999년 서강대학교 전자공학과 학사.
2001년 서강대학교 전자공학과 석사.
2001년 ~ 현재 런던 City University, 전기 전자, 정보공학 박사과정. 관심분야는 Machine Vision, Image Processing 등