論文93-30B-4-13

DP 매칭과 퍼지이론을 이용한 흘림체 온라인 한글인식

(Recognition of Cursive Korean Characters Using DP Matching and Fuzzy Theory)

沈 東 圭*, 咸 永 國*, 朴 來 弘**

(Dong Kyu Sim, Young Kug Ham, and Rae-Hong Park)

要 約

본 논문에서는 DP정합과 퍼지개념을 이용한 온라인 한글 흘림체 인식방법을 제안하였다. 본 방법은 입력 패턴의 크기와 회전에 영향을 받지 않으며, 한글의 구조적 특성을 이용하여 종래의 글자에 대한 정합이 아닌 자소별 정합을 통하여 DP 정합 시간을 대폭 줄였고, 정합특징으로서 입력 데이터와 표준 데이터 사이의 획간의 각도차와 길이비를 이용하였다. 또한 정합에 의하여 분리가 모호한 문자의 경우 자소의 특징과 자소간의 위치관계를 퍼지화하여 인식을 하였다. 컴퓨터 시뮬레이션을 통하여 제안된 방법이 효율적임을 보였다.

Abstract

This paper proposes an on-line recognition scheme of cursive Korean characters based on the DP matching and fuzzy concept. The proposed algorithm, invariant to rotation and size, reduces greatly the computational requirement of dynamic programming by matching phonemes rather than character patterns, where the angle difference and the ratio of lengths between input and reference patterns are adopted as matching features. The correct matching of poorly written cursive characters becomes possible by introducing the fuzzy concept in representing the features of phonemes and the ralative position between adjacent phonemes. Computer simulation results are observed to show the effectiveness of the proposed algorithm.

I. 서론

컴퓨터 성능의 발전으로 빠른 계산과 거대한 정보 저장, 검색 등 광범위한 데이타 처리가 가능케 되었다. 그러나 컴퓨터의 엄청난 능력에도 불구하고, 컴퓨터와 인간간의 인터페이스면에서의 발전은 느린 상태이다. 특히 입력 인터페이스의 경우는 더욱 심하여, 초창기의 천공카드에서 키보드 등으로 발전했으나, 이것은 인간이 사용하는 정보전달 수단과 달라 많은 사람들이 컴퓨터 이용에 어려움을 느끼고 있는 것이 사실이다. 이런 문제를 해결하기 위해 음성인식, 전자펜 등의 방법을 이용하는 연구가 많이 행해져 왔는데, 이중에서 펜에 의한 정보의 전달은 음성과 달리 문자인식을 통한 문자 정보 전송뿐만 아니라 그림 등 여러 가지 정보를 표현할 수 있는 이점이 있다. 그러나, 많은 사람은 흘림체에 익숙함에도 불구

*學生會員, **正會員, 西江大學校 電子工學科
(Dept. of Elec. Eng., Sogang Univ.)
接受日字: 1992年 12月 16日

하고, 지금까지 온라인 문자인식은 제한된 필기체에 국한되어 있어, 온라인 흘림체 문자인식이 요구되고 있다.

문자인식은 온라인 문자인식과 오프라인 문자인식으로 나누어진다. 오프라인 문자인식은 문서화되어 있거나 쓰여진 문자를 스캐너와 같은 입력도구를 이용하여 영상을 취득한 후, 문자를 분리, 인식하는 것이며, 온라인 문자인식은 전자펜 또는 마우스를 이용하여 문자를 입력받아, 인식하는 것이다. 이것은 오프라인과 달리 문자를 쓰는 필순, 획의 수, 방향정보 등을 이용할 수 있으나 이런 정보를 이용하려면, 입력문자의 형태가 제한될 수 밖에 없다.

온라인 문자인식에서 중요한 점은 사람마다 다른 필기 특성을 얼마나 잘 흡수하고 인식할 수 있느냐 하는 것이다. 기존의 많은 방법들은 필순이나, 획수에 제한을 두는 조건속에서 퍼지, 뉴럴, 정합 등의 방법을 이용하였다. 이러한 제한없이는, 필순이나 획수에 의한 정보를 이용하기 어렵기 때문에, 온라인 문자 인식이 오프라인과 비슷해지고, 시스템 구현이 너무 어려워진다. 그러므로, 우선 한글 문자 입력특성을 조사하는 것이 중요하다. 일반적으로 한글의 구조적 특성상, 초성, 중성, 종성순으로 입력된다. 그리고 필기자에 따라, 혹은 글자마다 다르게 초성과 중성 또는 중성과 종성을 붙여 쓰거나 모든 자소를 붙여 쓴다. 또한 한 자소를 필기할 때 획을 나누어 쓸 수도 있고, 한번에 쓸 수도 있으며, 여러 번에 나누어 쓰기도 하는데, 이런 문자인식을 흘림체 문자인식 분야로 구분하고 있다.[1,2] 흘림체 한글 인식을 위해 본 논문에서는 단순화된 DP (Dynamic Programming)정합을 퍼지개념과 결합하여 구조적으로 적용했다. 우선 모든 입력 획은 흘림체로 바꾸어 정합을 하였다. 기존의 DP정합의 경우 글자에 대한 정합을 행하였으나, 본 논문에선 한글의 구조적 특성을 이용하여, 초성, 중성, 종성순으로 하였고, 정합시 방향벡터를 이용하였으며, 정합정도를 판단하는 비용함수 (cost function)는 방향벡터와 획간의 길이의 비를 이용하였다. 그런데 이 특징만을 이용하면 자소간의 위치관계를 평가할 수 없기 때문에, 정합된 글자에 한하여, 자소끼리의 위치관계를 고려함으로써, 기존의 방법으로 분리되지 않던 것을 분리할 수 있었다. 이와같이 구조적 정합을 이용함으로써 정합 횟수를 대폭 감소시키고, 기존의 흘림체에서 제한된 글자만이 가능하였던 것에 비해 조합 가능한 한글을 거의 모두 인식할 수 있었다. 또한 기존 정합 방법에서 초성, 중성, 종성을 각각 여러 벌을 가지고 있어야 하는 단점을 개선했다. 초성, 중성, 종성 여러 벌을 기준 데이터로 가지고 있음으로 인해 많은 메모리가 필요하고 세분화된 분류가 요구되기 때문이다.

DP정합방법을 이용함으로써 글자의 전체적인 특성 및 국소적인 특성을 모두 평가할 수 있어 글씨를 쓰는 사람의 오류나 왜곡 등에 크게 영향을 받지 않고 글자를 분류해 낼 수 있었다. 이것은 기존의 퍼지 추론의 장점인 애매성의 흡수 능력과 병렬성을 모두 갖는다고 볼 수 있다. 그러나 인간은 전체적 모양에 모두 같은 가중치를 가지고 문자를 인식하는 것이 아니고, 어떤 특정한 부분의 특징이 인식에 중요한 요소가 될 수 있다. 이런 점을 고려하여 자소분리가 모호한 상황에서 중요한 특징점을 퍼지화하여 DP정합과 함께 인식에 사용하였다. 그리고 정합은 기준 패턴과 입력 패턴에 대하여 정합하는 것이므로, 기준 패턴만 바꾼다면 여러가지 다른 문자인식이 가능하고, 같은 글자의 다양한 표현이 가능하고, 데이타의 추가에 의해 인식 시스템의 변화없이 확장이 가능하다. 또한 이런 시스템은 적응 학습에 의한 기준 패턴의 생성 및 갱신이 가능하고, 적응 학습에 의한 시스템 설계가 가능하여 사용자마다 다른 입력 특성에 잘 반응하도록 만들 수 있다.

본 논문의 Ⅱ장에서는 온라인 문자인식의 동향에 대해 서술하였으며, Ⅲ장에서 제안된 흘림체 한글문자인식 방법에 대하여 서술하고, Ⅳ장에서는 실험결과에 대하여 서술하였으며, Ⅴ장에서 결론을 맺었다.

Ⅱ. 온라인 한글 문자인식 동향

온라인 문자인식은 평판이나 마우스와 같은 입력도구를 이용하여 펜의 접촉 여부와 위치정보를 알아내고, 이를 통해 얻은 시작점, 진행각, 길이와 끝점 등의 정보를 이용하여 실시간 인식을 행한다.[3] 온라인 문자인식은 패턴정합 방법, 통계적 방법과 구조적 방법을 기본으로 하고 있다. 최근에는 이러한 개념을 필순에 제한을 두거나, 획수에 제한을 두는 조건속에서 퍼지이론, 신경회로망 기법 등과 함께 이용하는 방법이 제안되었다.

퍼지개념을 이용한 문자인식 방법은 획의 특징점을 퍼지화하여 문자의 획을 인식하고, 인식된 획간의 위치관계를 퍼지화하여 문자를 인식하는 방법이다.[4-6] 이것은 한글이 7가지 (ㅡ, ＼, ／, ㄱ, ㄴ, ㅇ, ㅣ)의 기본적 요소로 이루어져 있다는 것과, 기본획은 한번에 입력된다는 것을 전제로 하였다. 이 방법은 초기의 방법이 이진 논리를 사용하여 애매한 상황에서 오인식되던 것을 보완하였으나 대부분의 퍼지를 적용하

는 정도를 획의 길이에 관계없이 일정하게 적용함으로써 문제가 발생할 수 있었다. 보통 문자입력시 획이 큰 경우는 획의 전체적 특성을 그대로 유지하나, 획의 길이가 짧은 경우는 각도의 변화나 구부러진 정도 등이 이상적인 경우와는 많이 다르게 나타난다. 그러므로 확인식시 획의 길이를 고려한 퍼지 소속함수를 사용함으로써 성능개선을 꾀할 수 있다

신경망을 이용한 온라인 문자인식으로, 기본 획으로 분리하여 쓰는 경우는 특징을 자소 단위로 추출하여 인식할 수 있도록 하여, 자소인식 네트웍을 필요로 한다.[7] 그러나 이것은 흘림체에 적용하기 어렵고, 흘림체를 인식하기 위해서는 문자인식하는 신경망이 필요하여 모든 문자를 네트웍으로 구성해야하는 단점이 있다.

적응 학습에 의한 온라인 문자인식방법은 기준 데이타를 학습에 의하여 생성하거나 이미 생성된 기준 데이타를 오인식이나 미인식한 경우 갱신한다.[8,9] 이 방법도 기준 데이타가 자소별로 준비되어 있을 경우에는 오인식 혹은 미인식된 흘림체를 이용한 학습에는 문제가 있다. 이것은 인식되지 않은 것이므로 자소추출이 불가능하여 자소별 기준 데이타의 갱신에 대한 자동화가 어렵기 때문이다.

DP정합을 이용한 방법은 한글 흘림체 인식에도 적용되었다. 그러나 이 방법은 DP정합방법 자체의 제한 때문에 많이 이용되지는 않았다.[11] DP정합 자체가 많은 시간을 요구하며, 글자 단위로 정합을 행하므로 한글의 경우 조합 가능한 모든 글자에 대하여 정합한다는 것이 불가능하기 때문이다. 그러므로 이것을 해결하기 위해 기존의 방법에선 대분류를 통하여 정합하였다. 그러나 모호한 상황에서 잘못 분류될 경우 오인식이나 미인식이 발생할 수 있는 문제점이 있다. 대분류는 흘려 써도 변화하지 않는 특징을 찾아야 하는데, 기존의 방법에선 최초 선분과 최후 선분의 방향각으로 글씨를 분류하지만, 이곳은 국소적 특성으로, 조금만 빨리 쓰면 최후 선분 등의 방향이 모호하게 된다. 물론 방향코드를 적은 수로 분류하면 잘못 분류될 가능성은 줄어드나, 반대로 정합을 시도해야 할 글자가 많이 늘어나게 된다. 문자에 대한 DP 정합을 하므로 정합횟수가 많이 증가하고, 항상 정합시 글자단위로 정합하므로 한 문자에 대한 정합시간도 길어, 모든 한글 문자를 정합시키는데 무리가 있다.

Ⅲ. 제안된 흘림체 한글 문자인식

본 논문의 온라인 문자인식 시스템의 전체 구성도 그림 1과 같다. 테블릿이나 마우스 등을 이용하여 획을 구성하는 좌표열을 얻고, 전처리과정을 통하여 잡음를 제거하여 선형 근사화를 시킨 후 기준 초성과 획의 방향 벡터를 이용하여 정합을 행하고, 정합정도를 평가하여, 가능성이 있으면 초성과 중성을 연결하는 흘림 벡터를 계산하여 계속해서 중성 정합 부분으로 들어간다. 그후 기준 중성 자소와 정합을 행하고 정합정도를 평가하여 글자가 완성되었는지 혹은 추가적으로 종성 정합이 필요한지 또는 전혀 문자를 만들 수 없는지를 평가한다. 만약 종성 정합이 더 필요하다면 종성 정합부분으로 간다. 본 논문에서는 자소 정합과 함께 자소의 특징점과 자소간의 위치관계를 퍼지화하여 인식에 사용하여, DFS (Depth First Search) 방법을 통해 탐색한 후, 최종 인식된 문자에 대하여 최소의 상이도를 가진 문자를 정합된 문자로 결정하는 알고리즘을 제안하였다. 본 장에서는 흘림체 한글의 특성 및 제안한 각 단계를 설명한다.

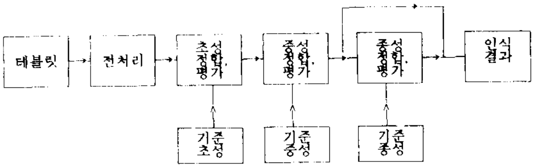


그림 1. 제안한 방법의 전체 시스템 구성도

Fig. 1. Blockdiagram of a proposed system.

1. 흘림체 한글의 특성

그림 2에 한글의 기본적인 6가지 유형구조를 보였다. 여기서 초성, 중성만으로 되어 있는 것은 유형 1, 2, 3이며, 이 유형은 중성의 위치에 따라 나누어진다. 유형 4, 5, 6은 초성, 중성, 종성이 모두 있는 경우를 보여주는데 이것 또한 중성의 위치에 따라 분류한 것이다.

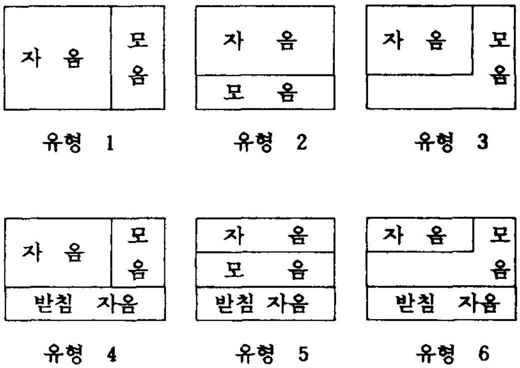


그림 2. 한글의 6가지 유형

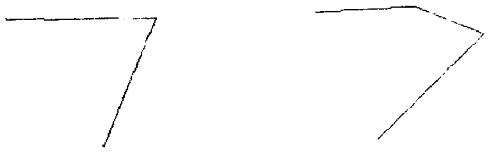
Fig. 2. 6 Korean character types.

그러므로 같은 초성일지라도 6가지 유형으로 나누어짐을 알 수 있다. 또한 중성도 종성의 유무에 따라 2가지 유형으로 나누어진다. 그림 3은 초성, 중성, 종성 각 유형에서 사용될 수 있는 자소를 보여준다.

	유 형	음 소
초성	1, 2, 3 4, 5, 6	ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅍ ㅎ ㄲ ㅃ ㅆ ㅉ
중성	1, 4	ㅏ ㅑ ㅓ ㅕ ㅣ ㅔ ㅐ ㅒ ㅖ ㅒ
	2, 5	ㅛ ㅠ ㅡ ㅜ ㅗ
	3, 6	ㅘ ㅚ ㅙ ㅢ ㅝ ㅞ ㅟ
종성	4, 5, 6	ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㄲ ㅄ ㄻ ㄾ ㄶ ㄺ ㅆ ㄵ

그림 3. 각 유형에서의 가능한 자소
Fig. 3. Possible phonemes in various character types.

흘림체의 인식에서 흘림체의 특성을 아는 것이 중요하다. 흘림체는 정자체에 획간의 불필요한 획이 추가되거나, 각도의 변화가 급격하거나, 급격히 변화되어야 할 곳에서 완만하게 변화하는 특성을 가지고 있다. 그리고 자소 사이에 짧은 획이 생략되는 경우도 있다. 그러나 획의 끝점과 다음 획의 시작점 사이에 불필요한 획 이외에 또 다른 불필요한 획이 추가되는 경우는 없다. 그리고 흘림체에서도 전체적 모양은 유지하며, 획의 길이의 비율도 거의 변화가 없다. 이상적인 경우, 방향 벡터가 급격히 변화하는 곳에서 완만하게 변화할지라도 초기획과 마지막획은 거의 같은 방향을 가지며, 중간의 방향각은 그 둘 사이의 각을 갖는다. 그림 4의 'ㄱ'의 예에서와 같이 이상적 자소 데이터로 흘림체의 자소를 표현해도 각도와 길이의 변화는 크지 않고, 완만한 변화에 의해 생긴 획도 초기의 획과 마지막 획의 각 사이에 있기 때문이다.



(a)　　(b)

그림 4. (a) 이상적 'ㄱ' (b) 흘림체 'ㄱ'
Fig. 4. (a) ideal 'ㄱ', (b) cursive pattern 'ㄱ'.

그림 5는 각도 변화가 이상적인 경우의 초성의 각도 변화 (0~180°)를 보여준다. 예를 들어 'ㄱ'의 경우 2개의 획으로 구성되어 있고, 첫번째 획의 각도는 0도이고, 다음 획의 각도는 230도이다. 그러므로 두 획간의 방향 변화는 130도가 된다. 그림 5의 데이타에서 보듯이 곡선으로 이루어진 자소 'ㅇ'과 'ㅎ'을 제외하면 한글의 각도 변화는 매우 큼을 알 수 있으며, 본 논문에선 'ㅇ'과 'ㅎ'의 인식을 위하여 별도의 방법을 사용하였다. 그림 5와 같이 한글 자소의 각도 변화는 매우 크게 변화하며 자소간 획의 각도는 많은 차이를 보인다. 그리고 획의 합성으로 한 자소를 만들고 거기에 또 다른 획이 첨가됨에 따라 다른 자소가 된다. 그리고 이 자소의 결합에 의해 흘림체 문자가 형성된다. 각 자소는 이와같은 규칙으로 흘림체 기준 자소가 형성되며, 자소내에서의 흘림과 자소간의 연결이 있을 수도 있다. 그런데 자소 사이의 흘림획은 자소와 자소에 의해 결정되므로 매우 많은 방향벡터를 갖는다. 그래서 본 논문에선 입력 획들을 입력순서로 모두 연결하여 일차원 방향각 함수로 만들어, 이상적인 경우의 자소의 방향각을 가지고 DP정합하여 자소를 분리, 인식한다.

자소	흘림체 획의 각도							각도 변화					
ㄱ	0	230						130					
ㄴ	270	0						90					
ㄷ	0	180	270	0				180	90	90			
ㄹ	0	270	180	0	180	270	0	90	90	180	180	90	90
ㅁ	270	90	0	270	180	0		180	90	90	90	180	
ㅂ	270	70	270	0	90	0		160	160	90	90	90	
ㅅ	230	50	330					180	80				
ㅈ	0	230	50	330				130	180	80			
ㅊ	290	180	0	230	50	330		110	180	130	180	80	
ㅋ	0	230	70	0				130	160	70			
ㅍ	0	180	270	45	270	180	0	180	90	135	135	90	180

그림 5. 흘림체 자음의 각도 변화
Fig. 5. Angle features of reference consonants.

2. 전처리 과정

이 과정은 불필요한 획을 제거하거나 정규화시키는 과정 혹은 평활화 과정 등을 말한다. 본 논문에선 획의 방향벡터를 이용함으로써 획의 방향벡터 정보를 잃지 말아야 한다. 그러나 필기자나 필기구의 특성에 의해 부분적으로 생긴 왜곡은 각도면에서 전체적 획의 방향과 많이 다를 수 있다. 즉 이런 작은 부분에서 심한 각도 변화를 일으킨 부분을 제거하는 것은 인식률 향상과 관계되며, 전체적 모양을 유지하는 한도 내에서 되도록 적은 좌표열로 감소시키는 것은 인식 속도나 인식과정에서 필요한 메모리 양의 감소와 관계가 있다. 그러므로 문자모양을 왜곡시키지 말고 잡음을 제거하면서도 되도록 적은 좌표열로 바꾸는 것이 필요하다. 이를 위한 다각형 근사화 방법은 지금까지 많이 제안되었으며 이는 평활화 작용과 필터링을 모두 수행하는 효과를 갖는다. Gaussian함수를 이용한 방법[10]은 많은 곱셈을 요구하며 필기속도가

항상 같은 것이 아니므로, 각 샘플링 좌표의 길이가 일정치 않아 계산이 복잡해지기 때문에 본 논문에서는 단순한 근사화 방법을 사용하였다.[11] 즉 시작점으로부터 계속되는 점 사이에 있는 점 중에 거리가 가장 먼 점까지의 거리가 임계값보다 크면 그것을 구석점으로 선택하고 그렇지 않으면 계속 진행해 나간다. 그리고 삐침 (hooking)을 제거하기 위하여 획의 초기점과 시작점의 입력시 임계치보다 긴 길이가 아니면 입력 데이터로 받아들이지 않는다. 그림 6과 같이 입력문자에서 많은 표본점들 중에서 인식에 필요한 점들만을 선택하여 인식부분에서 사용한다. 다음으로 입력 좌표열을 각 세그먼트의 방향각과 길이를 구한다. 기존의 방법에선 방향각을 방향 코드화하여 8방향 혹은 16방향 등을 사용했으나 본 논문에선 방향각을 유사도 평가에 사용하므로 방향각을 그대로 사용했다. 또한 각 세그먼트의 길이도 유사도 평가에 사용되므로 입력시 계산하는 것이 좋다.

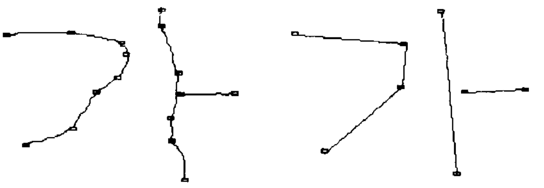


그림 6. 전처리 전후의 입력 패턴

Fig. 6. Input patterns before and after preprocessing.

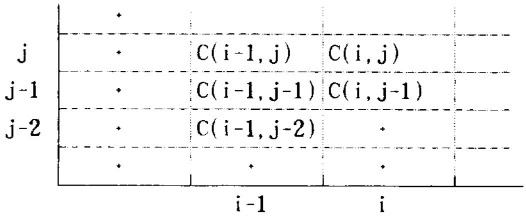
3. 자소 정합 및 문자인식

본 절에서는 초성, 중성, 종성을 인식하기 위해 제안한 방법에서 사용한 DP정합과, 정합의 모호성을 흡수하기 위해 특징량을 퍼지화하는 과정에 대해 서술하였다. 본 논문에서는 회전된 입력문자가 들어오는 경우 첫번째 획의 방향을 모든 획에서 빼줌으로써 기준 방향 0도로 입력패턴을 회전시켜 매칭부분에서 사용한다.

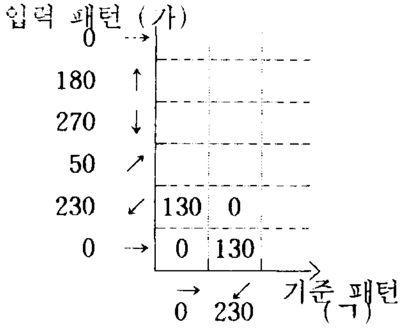
1) 기본적 DP정합

DP정합은 전체적 및 부분적 특성을 반영하는 특성을 가지고 있어 음성인식 및 영상처리 방법으로도 사용되고 있다. 이 방법은 기준이 되는 패턴과 왜곡되거나 변형된 입력과의 정합정도를 평가하는 지능적 방법이다. 이와같이 전체적 및 국소적 특성을 모두 정합하는 방법으로 tree matching 등의 방법[12]이 있으나, DP 정합은 수행이 반복적 계산에 의하여 이루어지므로 구성이 간단하며, 최적의 정합경로를 찾아낸다. DP정합의 장점중 다른 하나는 최적해외에도 부최적해까지 모두 구할 수 있다는 것이다. 이 정합법은 다항식 시간 복잡성을 가지고 있다. DP정합은 waveform 정합이므로 부분적으로 왜곡이 있어도 전체적 특성만 비슷하게 나타나면 유사도는 크게 나타난다.[13,14] 또한 기준 패턴 (reference pattern)의 어떤 부분이 입력 패턴의 어느 부분으로 정합되는지도 알 수 있어, 이것을 이용하여 특징점 추출이 가능하다.

그림 7(a)에서, X(i), i = 1, 2,⋯, N을 기준 신호라고 하고, X'(j), j = 1, 2, ⋯, M을 입력신호라 하면, (i, j) 배열에 각 index에서 신호의 차를 계산하고, 그 다음 최적 경로를 구하기 위해 초기점에서부터 상이도 (dissimilarity measure)를 누적하는 방법으로 계속 배열의 값을 갱신시켜 나간다. 그림에서 '·'은 이미 비용이 계산된 점을 나타낸다.



(a)



(b)

그림 7. DP정합에 의한 최적 경로계산 과정 및 정합의 예

(a) DP정합에 의한 최적 경로 계산 과정

(b) 정합예

Fig. 7. Optimal path by the DP matching and a matching example.

(a) computation of optimal path by the DP matching.

(b) matching example.

$$d(i,j) = |X(i) - X'(j)| \text{ 상이도 함수}$$

$$C(i,j) = \min_{1 \le k \le M} \{Arc(C(i-1,k),\ d(i,j)),\ Inc(C(i,j-1),\ d(i,j))\} \quad (1)$$

여기서 Arc(C(i-1,k), d(i,j))는 (i,j)에서의 누적 상이도 함수로, 위 식에선 (i,j)에서 (i-1,k)로부터 (i,j)로 경로가 형성될 때의 비용을 계산하기 위한 함수이고, Inc(C(i,j-1), d(i,j))는 (i,j)에서의 누적 상이도 함수로, 위 식에선 (i,j-1)에서 (i,j)로 경로가 형성될 때 비용계산을 위한 함수이다. 계산량을 줄이기 위하여 변형된 형태로 현재 위치에서 주위의 값만을 가지고 정합정도를 반복적으로 계산하는 형태는 식 (2)과 같다.[13]

$$C(i,j) = \min \{C(i-1,j) + d(i,j),\ C(i-1,j-1) + 2d(i,j),\ C(i,j-1) + d(i,j)\} \quad (2)$$

2) 단순화된 DP정합

기존의 정합법은 최적의 정합경로를 구하기 위해 모든 (i,j) 평면에서 상이도를 구하고 이것을 이용하여 기준 패턴의 i번째 세그먼트마다 가장 잘 정합된 것을 찾기 위해 모든 j에 대하여 상이도 값을 계산했다. 그러나 한글의 특성상 문자 waveform의 모양을 크게 변화시킬만한 획이 첨가되는 일은 거의 없다는 것이다. 잡음에 의해 생긴 비이상적 특성은 크지 않아 보통 전처리 과정에서 없어지고, 남아 있다 해도 이는 전체적 모양을 변화시키지 못하고 주위의 방향각과 비슷하다고 가정한다. 그러므로 기본적인 DP정합을 사용하여 많은 시간을 소비하면서 좋은 성능을 얻지 못하는 경우 단순화된 방법을 사용하여 계산시간을 대폭 감소시킬 수 있다. M이 입력패턴의 세그먼트의 수이고, N이 표준패턴의 획의 수라 할 때, 식 (1)에서의 DP정합의 order는 $M^2 \times N$이고, 식 (2)의 경우에는 $M \times N$이 된다. 이것은 큰 차수는 아니나 정합의 수가 많을 때는 문제가 될 수 있다.

여기서 DP정합을 시도하려는 것은 문자가 아니라 자소에 대한 정합이나, 입력 패턴이 흘림체이므로 자소와 자소가 붙어있어 기존의 정합법을 적용할 수 없다. 그러므로 본 논문에선 정합의 정도뿐만 아니라 어디까지가 인식하려는 자소인지를 판단할 수 있게 두가지 가정을 한다. 입력 패턴은 자소의 앞부분과 뒷부분은 생략되어선 안된다는 가정으로 자소의 앞부분과 뒷부분이 없으면 자소를 구성할 수 없기 때문이다. 두번째로 심한 각도변화를 일으키는 획의 첨가는 없어야 한다는 가정이 필요하다.

그림 7(b)는 '가' 입력 패턴에 대한 기준 자소패턴 'ㄱ'의 정합 예이다. 그림 7(b)의 세로축은 입력 패턴 '가'의 각 자소 성분을 각도로 나타냈으며, 가로축은 기준 패턴 'ㄱ'에 대한 방향 각도를 보여주고 있다. 기존의 모든 (i,j) 평면에서 상이도을 계산해 나가는 것이 아니고 정합이 제대로 된다면 정합 경로는 항상 진행하는 방향으로 나아간다. 그러므로 모든 좌표에 대하여 계산한 다음 거꾸로 최적 경로를 찾는 것이 아니라, 단순히 정합의 가능성을 평가하면 되므로 (i,j)에서는 (i+1,j), (i,j+1), (i+1,j+1)의 기준 패턴과 입력 패턴의 방향각의 차를 계산하여 최소값을 갖는 경로를 따라 간다. 그러다가 기준패턴의 끝에 최초로 정합된 부분을 자소로 분리될 부분으로 우선 생각한다. 그리고 다음 입력패턴의 방향각이 임계치를 넘지 않는다면 더 진행하여 여기까지를 자소의 정합된 부분으로 생각한다. 만약 정합이 되지 않으면 기준패턴 끝까지 가기 전에 입력패턴의 끝까지 가게 된다. 일단 정합된 것의 정합정도를 평가하여 기준범위안에 들면 인식된 것으로 생각하고 계속 정합을 진행해 나간다. 이렇게 함으로써 계산량은 선형 차수를 갖게 된다. 이것은 최적의 결과를 만드는 것으로 볼 수 없으나, 한글 문자인식에서 문자를 분리하는 데에는 충분하다.

본 논문에선 방향각의 차를 이용하고 있다. 기존의 많은 방법에선 방향각을 8방향 혹은 16방향 코드로 바꾸어 이것을 특징으로 사용하였으나 본 방법에선 자소의 정합정도를 정확하게 평가하기 위하여 방향각을 그대로 사용하고 있다.

단순화된 DP정합 후 정합정도의 평가가 필요하게 된다. 이는 DP의 특성상 조금만 비슷해도 정합하므로, 잘못된 정합을 제거하기 위한 것이다. 기존의 방법에선 방향각과 선분의 on-off (실제획-가상획) 그리고 자소추출을 이용하여 입력패턴과 기준패턴의 거리를 산출했다.[1] 본 논문에선 길이 가중치를 가진 방향각과 기준자소의 세그먼트 길이의 비를 이용하여 상이도를 평가했다. 또한 본 논문에선 자소추출을 기본으로 하므로 이것을 상이도에 포함시키지 않았다. 식 (3)은 자소추출후 상이도를 평가하는 식이다.

$$d_k = \sum_{i=1}^{M} \sum_{j=1}^{ci} L_{ji} \times \min\left(|R_{ji} - I_i|,\ 360 - |R_{ji} - I_i|\right)$$

$$LR_i = \sum_{j=1}^{ci} L_{ji}, \qquad 1_k = \sum_{i=1}^{ci} |L_i - LR_i| \quad (3)$$

$$e_k = d_k + \alpha \cdot 1_k, \qquad E = \sum_{k:all\ phonemes} e_k$$

여기서 M은 기준 패턴의 획의 수이고, 기준 패턴의 i번째 획 (I_i)에 정합된 입력 패턴의 세그먼트는 R_j (j = 1, ⋯ , C_i)이다. 같은 방법으로 기준 패턴의 i번째 획의 길이의 비는 L_i이고, 이 기준획에 정합된 세그

먼트들의 길이의 비는 L_j (j = 1, ⋯ , C_i)이다. d_k는 길이 가중치를 고려한 세그먼트의 방향각의 차이고, l_k는 기준 패턴의 획의 길이의 비와 입력 패턴의 길이의 비를 계산하는 식이고, e_k는 이 두 결과를 결합한 것이고, E는 글자를 구성하는 모든 자소의 상이도를 누적한 것이다. 그림 8는 예로서 기준 자소와 입력 자소가 정합되었을 때의 각 파라메터가 어디로 대응되는지를 보여주고 있다. 그림 8에서 기준 자소 'ㄱ'이 두 개의 획으로 이루어졌기 때문에 M은 2이고, 오른쪽과 같은 입력 패턴이 들어왔을 때, 기준 자소의 I1과 정합되는 부분이 R11, R21, R31이고, I2와 정합되는 부분이 R12, R22가 된다. 이렇게 정합된 결과를 이용하여 식 (3)에 따라 상이도를 구한다. 기준패턴의 첫번째 획 I1과 입력패턴 R11, R21, R31 사이의 길이가중치를 고려한 방향각의 차를 계산하고, 기준패턴의 I2와 입력패턴의 R12, R22 사이의 방향각 상이도를 구한다. 그리고 L11, L21, L31을 합하여 L1과의 차를 구하고, L12와 L22를 합하여 L2와 차를 구하여 상이도를 구하는 것이다.

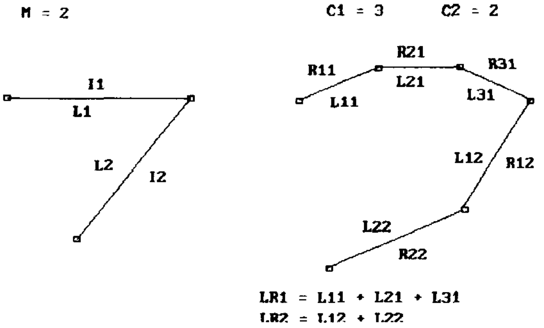


그림 8. 기준 자소와 정합된 자소와의 상이도 계산
Fig. 8. Dissimilarity calculation between reference and matched patterns.

기존방법에서 방향벡터의 산술적 차를 이용하던 것을 길이 가중치를 가진 방향벡터의 차를 이용함으로써, 작은 획의 경우 각도 변화가 심하나, 제안한 방법에서는 상대적으로 적은 효과를 내도록 하였다. 즉 중요한 획은 큰 가중치를 주고, 중요하지 않은 획은 적은 가중치를 두었으며, 기준 자소의 세그먼트 길이의 비를 가지고 입력된 자소의 길이의 비를 비교함으로써 얼마나 유사한 자소인지를 판단한다. 이 두 판단함수를 적절한 비율로 합하여 기준 자소와 입력자소의 상이도로 한다. 또한 이것은 자소에 대한 것이고 결국 문자에 대한 상이도를 구해야 하므로 이것에 다음 자소의 상이도를 누적한다. 최종적으로 인식된 글자들에 대하여 상이도가 가장 작은 것을 인식결과로 한다. 여기서 초성과 중성으로만 되어 있는 것과 초성, 중성 그리고 종성으로 되어 있는 문자는 서로 가중치를 주어 평가해야 한다. 여기서 α 값은 다른 입력패턴간의 방향각 상이도 값의 변화량과 세그먼트 길이의 비에 의한 상이도 값의 변화량이 같은 비율이 되도록 실험적으로 결정된다. 위의 평가 방법에서 사용한 가중치는 모두 전체 길이 중에서 차지하는 길이의 비이므로 입력 문자의 크기와 상관없이 글자의 형태만을 평가한다.

또 잘못된 자소 정합을 피하기 위하여, 획의 on-off를 정합 조건으로 사용하였다. 문자를 흘려 쓰므로 off인 획이 어디 있다는 것은 항상 보장할 수 없으나, 문자를 이루기 위해서는 꼭 on이 되어야 하는 세그먼트가 있다. 이것을 사용하는 목적은 흘림체 인식을 위하여 흘림체로 쓰지 않은 문자도 모두 흘림체로 고쳐서 인식하기 때문에 생기는 문제를 해결하기 위한 것이다. 흘림체로 만든다는 것은 떨어진 획을 붙여 주는 것인데, 이 붙여 준 획이 정합부분에서 획인 것처럼 생각하는 경우가 발생하기 때문이며, 이는 항상 자소의 시작 획에서 일어난다. 그리고 자소의 첫번째 획이 생략되는 경우는 없다는 특성을 이용하여 자소의 첫번째 세그먼트가 on인지를 확인한다. 만약 on이 아니면 자소로 인식하지 않는다.

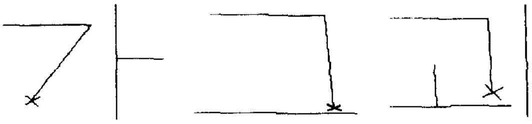
회전된 문자의 인식을 위하여 전처리에서 입력 패턴을 기준방향 0도로 회전시킨 것과 같이 기준 패턴도 회전시켜 정합한다. 이렇게 함으로써 입력 패턴의 방향에 관계없이 정합된다. 이것은 첫번째 획이 많은 각도 변화가 있을 때 실패할 수도 있으나, 한글에서 첫번째 획이 심하게 변화하는 경우는 'ㅊ'과 'ㅎ' 둘 뿐이고 DP정합 자체가 회전을 고려하지 않아도 약간의 회전은 흡수할 수 있으므로 가능하다. 'ㅊ'과 'ㅎ'은 일단 0도로 회전시켜 'ㅊ'과 'ㅎ'을 인식하고 2번째 직선선분의 각도를 추출하여 이것을 0도로 회전시켜 인식한다. 그림 9(a)와 (b)는 각각 '나'와 '구'자의 입력 패턴과 입력 패턴을 기준방향 0도로 회전시킨 것과 기준 패턴을 나타낸다.

(a)

(b)

그림 9. 입력문자, 회전된 문자, 표준패턴의 예
(a) '나'　(b) '구'

Fig. 9. Example of a skew input, rotated pattern, and reference pattern.
(a) '나',　(b) '구'.



(a)　(b)　(c)

그림 10. 각 유형의 'ㄱ'의 끝점 위치
(a) 유형 1 (b) 유형 2 (c) 유형 3

Fig. 10. End point for each character type.
(a) type 1, (b) type 2, (c) type 3.

3) 기준 자소의 구성

기준 자소는 초성, 중성, 종성으로 구성되며 각 자소에 대해 방향각과 길이의 비를 정의할 수 있다. 기존의 방법은 문자유형에 따라 여러가지 자소를 가지고 필요한 문자를 만들기 위해서는 그 문자의 유형에 맞는 자소를 선택하여 합성한다. 이렇게 함으로써 복잡도와 메모리 요구량이 증가한다. 한글에서는 문자유형이 바뀌어도 자소의 방향각이나 세그먼트의 길이비는 변하지 않으며, 자소간의 상이도가 매우 크므로 다른 자소로 정합 되는 일은 거의 없다. 그러나 'ㅇ'과 'ㅎ'은 곡선의 형태이어서 방향각으로 유일하게 표현하기 어려워 2가지의 방향각과 길이의 비를 기준 자소로하고, 다른 모든 자소는 문자의 유형에 관계없이 한 개의 기준 자소를 가지면 되나, 본 방법은 필순의 제한으로 인하여 필순이 2가지 이상으로 존재할 경우 추가적인 자소 패턴이 요구된다.

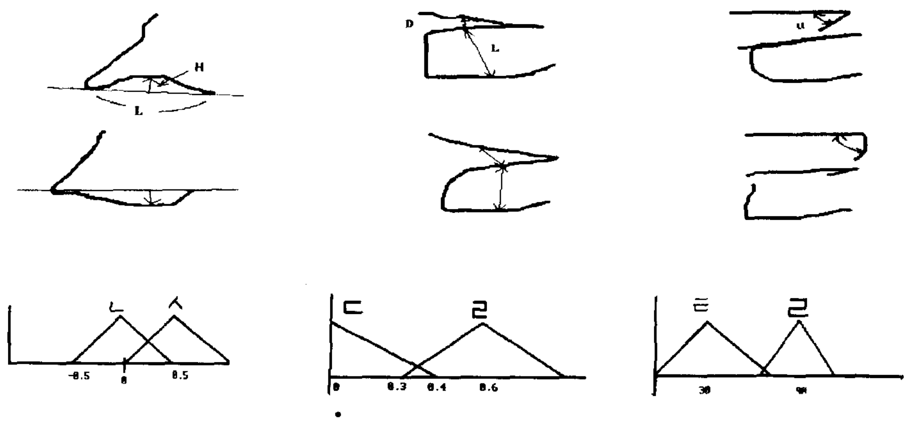
그런데 여기서 문제되는 것은 자소와 자소 사이의 방향 벡터를 만드는 것이다. 기존의 방법에선 유형에 따라 자소를 가지고 있으므로, 유형에 맞는 자소를 연결하여 자소와 자소 사이의 방향벡터를 구했다. 본 방법에선 초성의 형태가 중성에 의하여 결정되므로 유형을 3가지로 구분한다. 그러므로 초성의 각 자소마다 자소의 끝점의 위치를 3가지씩 가지고 있다. 중성은 시작점의 위치를 가지고 있으며, 이것을 이용하여 정합시 중성의 종류에 따라 초성의 끝점위치를 취하여 중성의 시작점과 방향 벡터를 계산하여 중성 정합시 사용한다. 중성과 종성 사이의 관계는 종성이 있느냐 없느냐에 따라 결정되는데, 있을 경우 이것도 방향 벡터를 만들어 주어야 하므로 중성의 끝점과 종성의 시작점을 가지고 방향벡터를 계산한다. 그림 10에서 보면 'ㄱ'의 끝점 위치는 중성의 종류에 따라 다르므로, 끝점으로부터 중성의 시작점을 연결하여 방향 벡터를 구한다.

4) 중성, 종성의 DP정합과 문자인식

초성의 정합시 문제되는 것은 마지막 획이 어디까지 초성으로 인식되어야 하는가이다. 이것은 중성 정합의 시작점이 되므로 매우 중요한 것이다. 이것을 해결하기 위하여 임계치안에 들 때까지 정합시켜 나간다. 그러나 이것은 실제로 부정확한 결과를 낼 수 있기 때문에 본 방법에서는 초성 정합시에는 마지막 획의 첫번째 정합되는 획까지만 정합을 실시하고, 중성의 정합시에는 초성의 마지막 방향벡터를 함께 이용한다. 중성과 종성 정합시에 초성과 달리 더 고려하여 줄 사항은, 초성의 마지막점과 중성의 초기점간의 방향벡터를 추가적으로 정합시켜야 한다는 것이다. 그러므로 초성의 마지막획의 방향벡터와 초성과 중성사이에 들어가는 방향벡터를 가지고 초성 분리 부분에서 자른 다음부터 정합시켜 나간다. 이를 위하여 정합될 첫번째 방향벡터가 초성의 마지막 획과 초성과 중성사이의 방향벡터중 어느 쪽에 더 가까운가를 판단하여 그것을 시작으로 하는 기준 패턴으로 정합시킨다. 이러한 방법으로 중성과 종성을 정합시켜 나가며 상이도 계산결과를 누적하여 나간다. 이러한 과정을 DFS 방법으로 문자인식 탐색영역을 모두 탐색한 후, 인식결과중 최소의 상이도를 갖는 문자를 인식된 문자로 선택한다.

5) 퍼지개념을 이용한 자소인식

문자나 자소 인식시 전체적 형태뿐만 아니라 부분적 특징도 매우 중요한 역할을 한다. 기존의 많은 방법들이 특징점을 추출하고, 그 특징점을 평가하는 방법을 사용한 이유가 여기에 있다. 그러나 흘림체에선 특징점을 추출하는 자체가 어렵기 때문에 본 논문에서는 획의 추출과 자소인식의 많은 부분을 DP정합법에 의하여 행하고, DP로는 분리가 모호한 자소에 대해 특징점을 추출하여 퍼지 소속집합에 포함정도를 평가하여 자소를 인식한다. 여기서 퍼지 소속 함수는



(a) (b) (c)

그림 11. 퍼지 소속함수
(a) 'ㄴ, ㅅ' (b) 'ㄷ, ㄹ' (c) 'ㅌ, ㄹ'
Fig. 11. Fuzzy membership function.
(a) 'ㄴ, ㅅ', (b) 'ㄷ, ㄹ', (c) 'ㅌ, ㄹ'.

그림 11처럼 삼각형 모양을 사용하였다. 이러한 자소의 예로 'ㄴ, ㅅ', 'ㄷ, ㄹ', 'ㅌ, ㄹ' 등이 있다. 그림 11(a)에서처럼, 'ㅅ'의 경우 DP 방법에 의하면 'ㄴ'으로 더 잘 정합되는 것으로 나타난다. 그러나 인간은 이것을 'ㄴ'과 비슷하지만 마지막 부분의 구부러진 방향을 더 중요한 판단 기준으로 사용하므로 'ㅅ'으로 판단할 것이다. 그리고 'ㄹ'의 경우 필순에 따라 두가지의 기준 패턴이 존재하는데, 그림 11(b)는 'ㄷ'과 구분이 모호하게 됨을 보여준다. 본 방법에선 그림 11(b)처럼 길이의 비 D/L을 특징으로 퍼지소속함수로 소속정도를 평가하였다. 다음으로 'ㅌ'과 'ㄹ'은 첫번째 획과 2번째 획의 방향각의 변화량을 특징량으로 사용하였다. 이러한 방법으로 DP정합 방법만을 가지고 인식이 불가능하던 것을 정인식할 수 있었다. 그림 11에서 각 그래프는 소속함수를 나타낸다.

그리고 자소의 판단과 함께 자소간의 위치관계를 퍼지화하여 문자를 인식하는데 이것은 DP정합 자체가 자소간의 위치관계를 자소의 마지막 점과 다음 자소의 시작점의 방향벡터를 이용하나 이것은 많은 오류를 낼 수 있다. 그림 12는 '냐'자가 '난'자로 오인식된 예이다. 그러므로 자소의 위치 관계를 추출하고 비교해야 하는데, 이것을 위해 종성의 유무와 관계없이 유형을 중성의 3가지 유형에 따라 나누고, 초성이 중성에 어떤 쪽이 얼마나 많이 포함되어 있느냐를 판단 기준으로 사용한다.

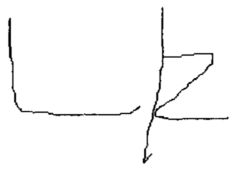


그림 12. '난'자로 오인식된 '냐'
Fig. 12. Mismatched example of '냐'.

우선 자소를 포함하는 최소 직사각형을 자소의 영역이라 하자. 그림 13(a)의 첫번째 유형은 초성영역이 중성영역의 왼쪽에 있는 정도를 판단하기 위한 영역을 보여준다. 즉 중성 영역의 오른쪽과 왼쪽이 각각 초성의 오른쪽과 왼쪽보다 큰 좌표 위치에 있어야 한다. 그리고 초성 영역의 중심과 중성 영역의 중심사이의 방향벡터를 퍼지 소속함수에 사용한다. 그림 13(b)의 유형 2는 중성영역이 초성영역의 밑에 있는 정도를 평가하기 위한 영역을 보여준다. 이것도 유형 1과 마찬가지로 초성의 최상단과 최하단이 중성의 최상단과 최하단의 위에 있어야하고, 각 자소의 중심위치의 방향 벡터를 퍼지화한다. 그림 13(c)의 유형 3은 초성영역이 중성영역에 얼마나 포함되어 있는지를 평가하기 위한 영역을 보여주는 것으로, 포함된 초성영역의 비를 이용하여 퍼지화한다. 그리고 그림 13(d)에서처럼 종성의 위치관계는 초성과 중성을 모

두 포함하는 최소 직사각형이 종성을 포함하는 최소 직사각형위에 있는 정도로 평가한다.

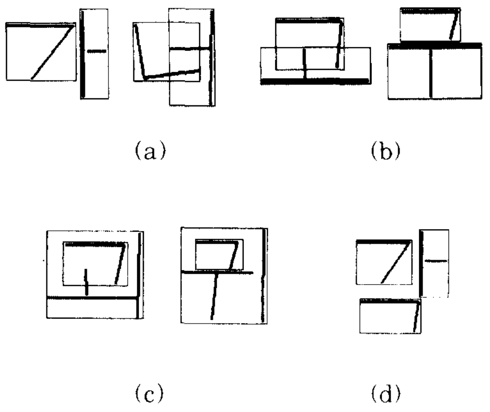


그림 13. 자소간 위치 관계
(a) 유형 1의 초성과 중성
(b) 유형 2의 초성과 중성
(c) 유형 3의 초성과 중성
(d) 초성, 중성과 종성

Fig. 13. Ralative position between phonemes.
(a) between the first consonant and vowel in type 1.
(b) between the first consonant and vowel in type 2.
(c) between the first consonant and vowel in type 3.
(d) between the first consonant, vowel, and last consonant.

최종적으로 인식된 문자를 결정하기 위하여 퍼지추론과 DP정합의 상이도를 함께 평가하는 것이 필요하다. 퍼지추론에 의하여 어떤 문자에 속하는 정도를 계산하고, 이것으로부터 문자에 소속하지 않는 정도로 바꾸어 이것을 DP정합의 상이도 값의 가중치로 사용하여 최소값을 갖는 문자를 최종 인식된 문자로 한다. 여기서 퍼지추론에 의한 문자에 대한 소속 정도는 초성에 속하는 정도 FIRST(초성)(X), 초성과 중성 사이의 위치 관계 FIRST_VOWEL(X), 중성과 종성의 위치 관계 VOWEL_LAST(X), 종성에 속하는 정도 LAST(종성)(X)의 퍼지 교집합을 구한다. 여기서 X는 입력 패턴을 나타내고, 정의되지 않는 소속 함수값은 1로한다. 예를 들어 모호한 '륵' 자가 입력으로 들어올 경우 '륵' 자와 '득' 자로 DP정합되어 각각 상이도를 가지게 된다. 여기서 '륵' 자에 소속 정도는 퍼지 추론에 의하여 FIRST('ㄹ')(X) ∧ FIRST_VOWEL(X) ∧ VOWEL_LAST(X) ∧ LAST('ㄱ')(X)와 FIRST('ㄷ')(X) ∧ FIRST_VOWEL(X) ∧ VOWEL_LAST(X) ∧ LAST('ㄱ')(X)을 모두 구하고, 소속하지 않는 정도를 구하여, 각각에 DP정합 상이도 값에 곱하여 최소의 값을 갖는 문자를 인식된 문자로 한다.

그리고 지금까지는 DP정합으로 문자를 분리하고 상이도를 평가하고, DP정합에 의하여 분류가 모호한 경우와 자소간의 위치 관계를 퍼지화하였다. 그러나 이것은 DP정합에 의하여 자소가 분리되었을 때에는 문제가 되지 않으나, DP정합에서 자소 분리가 실패하면 인식이 불가능해 진다. 한글의 대부분의 자소는 직선 획으로 이루어지고 각도의 변화가 심하나, 'ㅇ'과 'ㅎ'은 곡선형태이어서 방향벡터의 나열로 나타내어 정합을 실시하면, 자소추출의 실패확률이 높다. 물론 많은 기준패턴을 가지고 있으면 가능하나, 정합 횟수의 증가와 기준 패턴이 증가하는 단점을 가지고 있다. 그래서 본 논문에선 'ㅇ'과 'ㅎ'은 DP정합이 아닌 다른 방법을 사용하였다. 우선 'ㅎ'은 'ㅗ'까지는 DP정합에 의하여 분리하고 'ㅇ'자를 다른 방법으로 분리한다. 'ㅇ'자의 방향벡터의 특성을 보면 계속 벡터의 방향각이 증가하는 방향만으로 변화한다는 것이다. 본 방법에선 초기점부터 세그먼트의 방향각이 증가하는 양을 계속 누적하여 이것이 360도에 가장 가까워지는 점을 정합점으로 판단한다. 이러한 방법으로 인하여 크게 왜곡된 'ㅇ'자를 분리할 수 있다.

Ⅳ. 실험결과 및 분석

인식 시스팀은 IBM PC 486 (110 MHz)을 사용하여 C언어로 프로그램을 작성했다. 그림 14의 인식 시스팀은 문서작성의 초보적 형태로 공백, 줄바꿈, 뒤로 지움 등을 할 수 있도록 하였다. 실험결과 퍼지추론에 의한 필기체 인식은 95%의 높은 인식률을 보였다.

그러나 이 방법은 획의 7가지 기본패턴을 인식하는 과정을 통하여 문자를 인식하는 것으로 획을 붙여서 쓰는 경우, 즉 흘림체의 경우 인식이 불가능하다. 반면 DP정합에 의한 흘림체 문자 500자에 대해서는 85%의 인식률을 얻었으며, DP정합후 특징점을 퍼지추론에 적용할 경우 90%의 인식률을 얻었다. 여기서 DP정합에 의하여 후보문자로 인식된 문자들에 포함될 확률은 99%로 매우 높았다. 결국 오인식의 원인은 유사한 상이도를 갖는 문자들에 의한 것이므로, 정확한 정합정도의 평가와 많은 자소에 대한 특징량

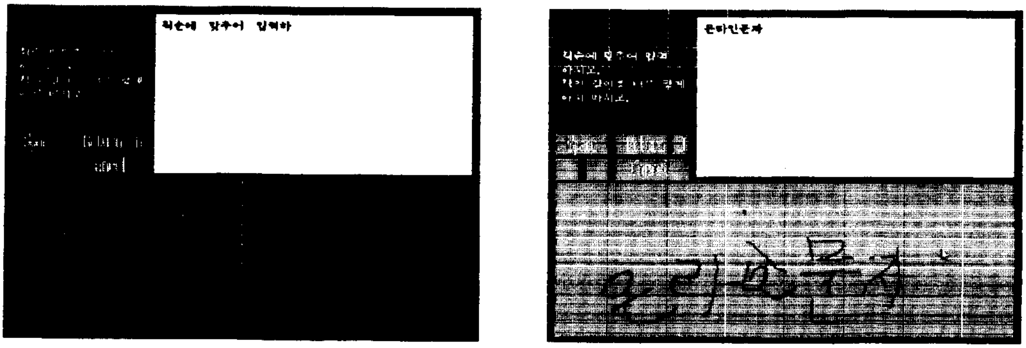


그림 14. 인식 시스템
Fig. 14. Recognition system.

의 퍼지추론으로 더욱 높은 인식률을 얻을 수 있음을 알 수 있다. 또한 DP정합의 적용에서 대분류 없이 모든 문자에 대하여 정합할 경우 평균 10초/자 이상이 소요되었으나, 자소별 정합과 퍼지 추론에 의한 탐색영역의 감소와 정합 크기의 감소로 0.16초로 대폭 감소되었다. 본 논문에선 모든 문자가 회전된 것을 가정하였으나, 인식률에 큰 영향을 주지 않음을 알 수 있다. 또한 식 (3)의 α 값은 인식률에 큰 영향을 주지 못했는데, 이것은 한 미지의 입력에 대하여 같은 값을 갖고, 정합된 것중에 상대적으로 적은 상이도를 선택하기 때문이다.

그림 15는 초성, 중성과 종성의 흘림을 기존의 흘림체 DP정합과 특징점을 사용하여 올바로 인식된 대표문자들에 대한 예이다. 실험결과에서 보듯이 문자에서 자소간의 흘림이 있고, 없고에 관계없이 인식됨을 알 수 있다. 그림 16은 기존의 DP 정합법으로는 문자의 유사성으로 인해 오인식되는 문자들이나, 본 논문에서는 단순한 DP 정합법에 특징량을 퍼지화함으로써 인식된 실험결과이다. 본 실험에선 같은 입력 패턴을 가지고 DP 정합만을 사용하였을 경우와 DP 정합과 퍼지를 사용할 경우를 실험하였다.

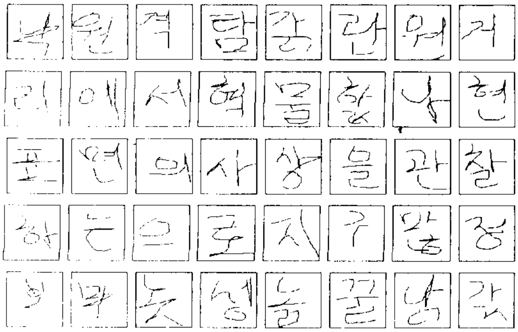
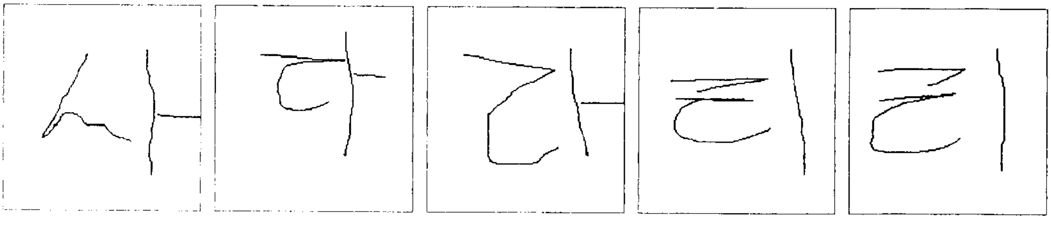


그림 15. 정인식된 문자의 예
Fig 15. Examples of recognized characters.

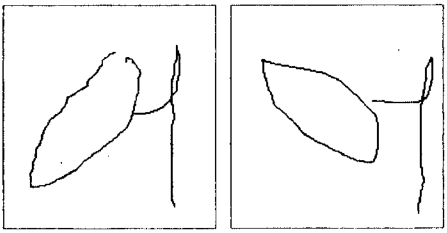
그림 16의 '사' 자는 DP 정합만을 할 경우 '나' 자로, '라' 자는 '다' 자로, '타' 자는 '라' 자로 인식되었으나, 같은 패턴을 가지고 퍼지개념을 이용하여 정인식할 수 있었다. 그림 17은 'ㅇ' 이 많이 왜곡되었을



사 다 라 티 리

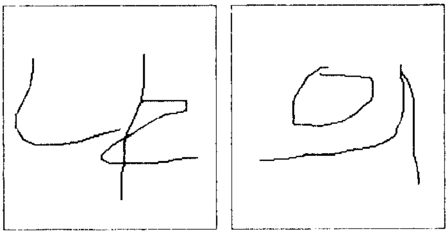
그림 16. 퍼지개념을 이용하여 인식된 문자
Fig. 16. Recognized characters using the fuzzy concept.

경우 단순한 DP정합에서는 인식이 어렵기 때문에, 'ㅎ'과 함께 각도변화량의 누적을 이용하여 인식하였다. 이러한 방법으로 이상적 자소와 크게 다른 입력패턴도 잘 인식할 수 있었다. 그림 18은 DP 정합에 의해서는 오인식되는 문자를 자소간의 위치관계를 퍼지화하여 인식한 예이다. 그림 19는 회전된 문자의 인식 예로서 입력 패턴의 첫번째 획을 0도 방향으로 회전시키고, 기준 자소도 첫번째 획이 0도가 되도록 회전시켜 놓고 DP정합과 퍼지 이론을 이용하여 인식하였다.



어 어

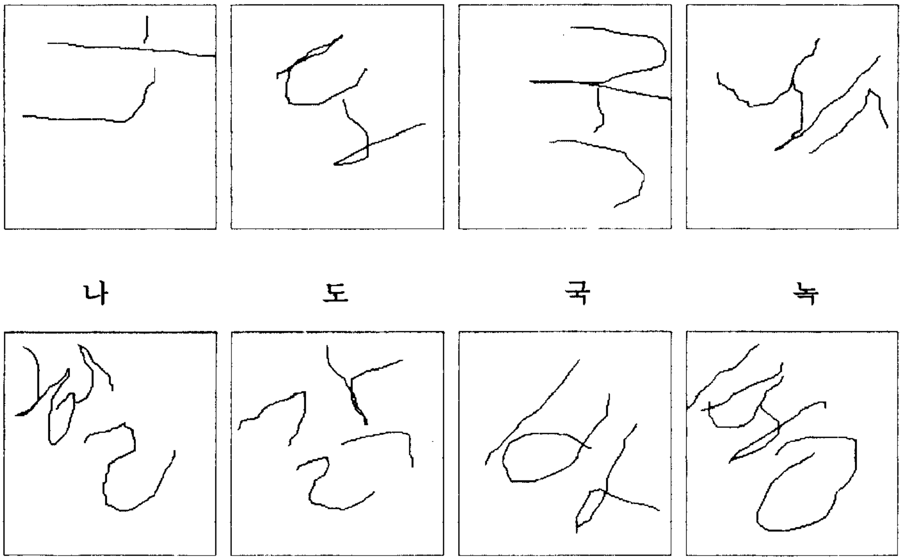
그림 17. 왜곡된 'ㅇ'의 인식예
Fig. 17. Recognized characters with distorted 'ㅇ' patterns.



냐 의

그림 18. 자소간의 위치관계를 이용하여 인식된 예
Fig. 18. Recognized characters using the relative position of phonemes.

실험 결과에서 보인 바와 같이 본 논문에서는 기존의 퍼지 이론만을 이용하면 흘림체 문자에선 특징점의 추출이 불가능하다는 점과 단순 DP정합을 함으로써 문자의 형태의 유사성으로 인해 오인식이 잦은 문자에 대해서 DP정합에 퍼지 이론을 결합함으로써 기존의 방법에서 발생했던 문제점들을 해결할 수 있었으며, 획순에 제한을 둔 필기체 및 흘림체에 대해서 인식이 가능하게 하였다. 그리고 자소간의 위치 관계를 퍼지화하여 인식률을 향상시키고 정합 횟수를 감



나 도 국 녹

헐 갉 바 통

그림 19. 회전된 문자의 인식 예
Fig. 19. Recognition of skewed characters.

소시킬 수 있었다. 또한 단순화된 DP정합에서 'ㅇ'과 'ㅎ'과 같이 많이 왜곡된 자소의 추출이 불가능할 수 있으므로 각도의 변화량의 누적값을 이용하여 자소를 옳게 인식하였다.

V. 결론

본 논문에서는 자소별 DP 정합을 통하여 정합 횟수를 대폭 감소시키고, 기존의 필기체뿐만 아니라 인간의 필기습성에 맞는 흘림체 인식까지 가능케 하였다. 정합은 기존의 최적화 정합을 단순화시켜 정합률을 떨어뜨리지 않으면서 계산 시간을 크게 줄였다. 기존의 방향벡터만을 이용한 상이도 계산의 문제점을 개선하기 위하여, 길이 가중치를 가진 방향벡터를 이용하여 중요한 세그먼트가 상이도 계산에 크게 기여하도록 하였으며, 또한 자소의 길이의 비를 이용하여 기준 자소와의 상이도를 계산하였고, 문자의 크기에 상관없이 인식하게끔 하였다. 또한 DP정합에 의하여 분리가 불가능한 모호한 자소와 자소간의 위치관계를 퍼지화함으로써 성능을 개선시켰다. 그리고 회전된 문자를 인식하기 위하여 입력 패턴과 표준 패턴을 기준점으로 회전시켜 인식하였다.

온라인 문자인식에서는 되도록 인간에게 편리한 필기환경과 정확한 인식률이 중요한 문제이다. 본 방법에선 기존의 획수의 제한을 없애기 위한 방법에 중점을 두었다. 앞으로의 과제는 획순의 제한을 가하지 않고, 그림과 문자를 분리 인식할 수 있는 시스템구현이다. 또한 한글뿐만 아니라 영문, 숫자, 한자와 제스처인식 등으로 인식 능력을 확장할 필요가 있다.

參考文獻

[1] 이 희동, 김 태균, 安居完猛, 中鳥正之, "확장된 DP정합법에 의한 흘림체 한글 온 라인 인식," 전자공학회논문지, 제 26권, 제 1호, pp. 29-37, 1989년 1월.

[2] 김 진형, "펜 컴퓨터의 현황과 추세," 정보과학회논문지, 제 9권, 제 1호, pp. 79 86, 1991년 2월.

[3] T. Wakahara, H. Murase, and K. Odaka, "On-Line handwriting recognition," *Proc. IEEE*, vol. 80, no. 7, pp. 1181-1194, July 1992.

[4] 김 재희, "문자인식에서의 퍼지이론응용," 정보과학회논문지, 제 10권, 제 1호, p p. 38-47, 1992년 2월.

[5] 심 영철, 오 경환, "온라인 필기체 한글 인식," 정보과학회 가을학술발표논문집, 제 18권, 제 2호, pp. 301-304, 1991년.

[6] 최 용엽, 최 갑석, "Fuzzy 추론을 이용한 온라인 필기체 한글문자 인식에 관한 연구," 전자공학회논문지, 제 27권, 제 11호, pp. 103-110, 1990년 11월.

[7] 최 정훈, 권 희용, 김 춘석, 황 희융, "신경망 모델을 이용한 한글 필기체 온라인 인식," 정보과학회논문지, 제 17권, 제 5호, pp. 540-549, 1990년 9월.

[8] 정 봉만, 권 오석, 김 태균, "온라인 입력 한글의 적응학습과 인식에 관한 연 구," 정보과학회논문지, 제 16권, 제 5호, pp. 487-498, 1989년 9월.

[9] 정 봉만, 권 오석, 김 태균, "적응학습법에 의한 문자집합별 온라인 인식," 정보과 학회논문지, 제 18권, 제 1호, pp. 14-22, 1991년 1월.

[10] J. J. Rodr guez, and J. K. Aggarwal, "Matching aerial images to 3-D terrain maps," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. PAMI-12, no. 1 2, pp. 1138-1149, Dec. 1990.

[11] F. Y. Feng and T. Pavlidis, "Finding vertices in a picture," *Computer Graphi cs Image Processing*, vol. 2, pp. 103-117, 1973.

[12] Y. C. Cheng and S. Y. Lu, "Waveform correlation by tree matching," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. PAMI-7, no. 3, pp. 299-305, M ay 1985.

[13] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoke n word recognition," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. A SSP-26, no. 1, pp. 43-49, Feb. 1978.

[14] H. Ma tre and Y. Wu, "Improving dynamic programming to solve image regis tration," *Pattern Recognition*, vol. 20, no. 4, pp. 443-462, 1987.

著者紹介

沈 東 圭 (學生會員)
1970年 7月 10日生. 1993年 서강대학교 전자공학과 졸업(학사). 1993年 2月 현재 서강대학교 전자공학과 석사과정 재학중. 주관심분야는 패턴인식, 컴퓨터 비젼, 동영상처리 등임.

咸 永 國 (學生會員) 第 29 卷 B 編 第 10 號 參照
현재 서강대학교 전자공학과 박사과정

•

朴 來 弘 (正會員) 第 29 卷 B 編 第 10 號 參照
현재 서강대학교 전자공학과 부교수