논문 01-26-4B-14 한국통신학회논문지 '01-4 Vol.26 No.4B

제한된 필기글꼴을 이용한 휴대형 정보기기용 한영 온라인 문자인식에 관한 연구

정회원 홍 성 민*, 국 일 호**, 조 원 경***

A Study on the On-line Recognition of Korean-English Characters Using Constrained Strokes for PDAs

Sungmin Hong*, Ilho Kook**, Wonkyung Cho*** *Regular Members*

요 약

본 논문에서는 제한된 필기글꼴을 이용한 휴대형 정보기기용 온라인 문자 인식 알고리즘을 제안하였다. 한글과 영숫자를 동시에 사용하는 문자 인식은 애매성으로 인하여 인식율이 낮아지며, 이를 극복하기 위하여 모드 변환이나 영역 분리 등의 제약을 하게 된다. 본 논문에서 제안한 인식 알고리즘은 한글과 영문자, 숫자를 혼용하여 사용할 수밖에 없는 우리의 문자 환경에서 사용자의 편의성을 최대한 살려 입력 모드 전환이나 필기 영역 분리 등의 제약을 하지 않는 단일 알고리즘이다. 또한 역추적에 의하여 인식 과정에서 발생할 수 있는 미인식 또는 오인식을 보정할 수 있도록 한다.

제안한 알고리즘은 전체 알고리즘의 크기가 작으며 계산량이 적어서 메모리와 속도 등의 성능에 있어서 자원의 제약을 가질 수밖에 없는 초소형 휴대형 정보기기의 입력 장치로서 적합하도록 연구하였다. 실험 결과 영숫자 98%, 한글 97%의 인식율을 얻어 유용성을 확인하였다.

ABSTRACT

In this paper, an on-line character recognition algorithm using constrained strokes is proposed, which can be used for mobile computing device like PDAs. With this algorithm, a character can be recognized without any other external distinction method of character set; Korean or ANSI-Characters. This is very useful especially for mobile computing devices, but causes very high ambiguity problem between written character sets to recognize. For the case of hand-written Korean vowels that is consist of relative simple strokes, it's very hard to distinguish from ANSI characters. The algorithm has forward- and backward-state transition rules. Trough the forward-tracing process, a character can be recognized and backward tracing of graph corrects recognition fail.

Proposed algorithm has recognition capacity of Korean and alphanumeric character without any external distinction method of character sets; changing input mode or separated-, boxed-writing area. This feature is good for limited size of mobile computing devices and increases the usability for two kinds of character sets that is must written alternatively like ours. And the algorithm uses less hardware resources like memory. Practically code size is about 56K bytes when targeted to x86 CPU including recognition algorithm and database. By the experimental results, recognition rate is obtained 98% in Alphabet and 97% in Korean characters.

* 여주대학 전자과 ** ㈜코덱실 공동대표,
*** 경희대학교 전자정보학부
논문번호 : 99247-0616, 접수일자 : 1999년 6월 16일

Ⅰ. 서 론

최근, 초소형 컴퓨터는 팜-탑(Palm-Top) 또는 핸드-헬드(Hand-Held) 컴퓨터, PDA(Personal Digital Assistant) 등 여러 가지 형태의 휴대형 정보기기로 발달하고 있다. 휴대형 정보기기로서는 소형화가 가장 중요하며, 이를 이루기 위한 기술적 요소는 여러 가지가 있으나 하드웨어 기술을 제외하면 사용자 인터페이스가 소형화에 가장 큰 걸림돌이 되고 있다. 사용자 인터페이스는 입력장치와 출력장치로 나누어 볼 수 있으며, 출력장치는 특별한 경우를 제외하고는 대부분 액정화면을 사용하며, 현재로서는 가장 좋은 출력 장치라 할 수 있겠다. 그러나 입력장치로는 좀더 다양한 방법이 있어서 키보드, 문자인식, 음성인식 등을 사용할 수 있다. 핸드-헬드 컴퓨터 등에서는 소형화 된 키보드를 주로 사용하며, 팜-탑 컴퓨터, PDA등에서는 문자인식을 주로 사용하고, 음성인식은 제한된 목적 이외에는 사용되지 못하는 실정이다.

사용자 인터페이스로 키보드를 사용하는 경우에는 키보드의 물리적인 크기로 인하여 기본적으로 소형화에 한계가 있을 수밖에 없으며, 음성인식은 데이터의 양과 계산량이 많아지며, 사람마다 전혀 다른 음색을 갖고, 주위의 소음 등에 의한 변형이 커지는 등의 이유로 다양한 사용자와 단어, 문장을 인식하기 어려운 현실적 기술의 한계로 인하여 많이 사용되지 못하고 있다. 따라서 포켓 사이즈(Pocket Size)의 소형화 된 휴대형 정보기기에서는 문자인식을 사용하는 것이 현재로서는 최선이라 할 수 있다. 현재 대부분의 휴대형 정보기기에서는 액정화면 위에 투명 태블릿(tablet)을 설치하고 펜과 문자인식을 통하여 문자 정보를 입력한다. 문자인식의 어려움으로 인하여, 몇몇 정보 기기들은 펜의 원시 데이터 형태로 입출력 처리하는 경우도 있으나 이는 정보기기로서 문자정보 수집, 저장, 작성, 검색 및 양방향 문자 통신 기능 수행 등 정보처리에 큰 제약이 될 수 있다.

대표적인 PDA 기종인 Pilot이나 Zoomer, Newton, MessagePad, Celvic 등이 문자 인식을 사용자 인터페이스로 채택한 소형 컴퓨터이며, 이들에는 영문자 인식기가 내장되어 있어 영문자 환경에서 사용하기에는 큰 불편이 없으나, 한글 인식기가 없는 PDA에서는 한글을 입력하기에는 매우 불편하다. 컴퓨터 내부에서의 한글은 소프트웨어에 의한 한글 지원 프로그램을 설치하여 사용할 수 있으나, 한글의 입력에 있어서는 키보드를 에뮬레이션하거나 영문자 인식기를 사용하여 로만지 입력 방식으로 대응하는 한글 자소로 변환하여 사용하는 형편이다[1].

온라인 문자 인식에 있어서 흘려 쓰기, 단어 단위 입력을 허용하는 등 필기의 자유도를 높이거나 인식율의 향상을 위한 복잡한 알고리즘으로 통계적인 모델, 뉴럴 및 퍼지 모델 등이 제안되어 주로 데스크 탑이나 웍스테이션 컴퓨터에서 이루어진 연구가 대부분으로 인식기의 크기가 매우 크며, 계산량이 많아 PDA 등의 휴대형 정보기기에의 적용은 어려운 실정이다[2][3][4][5]. 또한 한글과 영문자에 대하여 별도의 인식 알고리즘, 혹은 인식 데이터 베이스, 인식 파라미터를 사용함으로써 인식 문자 모드 전환이나 필기 영역 분리 등의 방법으로 인식기를 바꾸어주어야만 한다. 특히 영문자 인식기만을 내장하고 있는 Pilot, Zoomer, Newton, MessagePad 등의 PDA에서는 영문자와 숫자, 제어문자, 특수문자 등의 문자 그룹들을 모드 전환하여 인식하고 있으며, 높은 인식율을 달성하기 위하여 온라인 문자 인식기 전용으로 대부분 1획으로 구성된 그래피티라는 필기 글꼴을 사용하고 있다[6]. 별도의 필기체를 가지는 영문의 경우 이와 같은 1획의 그래피티 방법이 효과적일 수 있지만, 한글의 경우 기존에 제안되었던 한글 필기체를 살펴보면 자소의 원형을 유지하지 못하는 경우가 많아 특수한 경우를 제외하고 널리 사용되지 못하고 있다[7][8].

한글과 영문자, 숫자를 혼용하여 하나의 알고리즘으로 인식하고자 할 때, 가장 어려운 점은 영숫자의 경우 획으로 문자를 구성하며, 한글은 획으로 문자가 아닌 자소가 구성되고 자소의 조합으로 문자가 구성된다는 점이다. 일반적인 사용자의 편의성을 강조한다면, 한글도 문자 단위로 인식되는 것이 필요할 것이지만, 이 경우 인식 대상 문자의 수가 2350자 이상이 되어야 하며, 이는 인식기의 크기와 그 계산량 등에 있어서 큰 부담이 되어 휴대형 정보기기에 적합하지 않은 알고리즘이 될 수 있다. 따라서 입력 장치로서 키보드를 대체하는 수준이라면, 키보드상의 문자에 초점을 맞추어 영숫자와 같은 레벨로 한글을 자소 단위로 입력하거나, 문자 단위 입력을 하더라도 자소 단위로 분리하여 인식하고 모아쓰기 오토마타에 의하여 문자를 조합하는 것으로 해결할 수 있다.

본 논문에서는 초소형 휴대형 정보기기의 입력 방법으로 적합한 온라인 문자 인식 알고리즘을 제

안하고자 한다. 이의 조건으로는 인식기의 크기가 작아서 메모리의 요구가 크지 않아야 하며, 계산량이 적어 하드웨어 자원에 부담을 주지 않으면서도 인식율 등이 실용 가능한 수준으로 되어야 함은 물론이다. 또한 한글과 영문자를 동시에 사용하는 우리의 문자 문화와 휴대형 정보기기로서의 역할에 따라 한글과 영문자, 숫자 등의 인식이 가능하여야 하며, 사용자의 편의성을 향상 시키기 위하여 입력 대상 문자를 구분하지 않고, 필기 영역을 분리하지 않는 방식으로 각 문자 집합을 혼용하여 사용할 수 있는 알고리즘이어야 한다. 따라서 본 논문에서는 인식기의 크기가 60KB 이내인 작은 크기, 복잡한 산술 연산을 사용하지 않을 것, 인식 속도나 인식율이 사용자를 불편하게 하여서는 안될 것, 모드 전환이나 영역 구분 없이 한글, 영문자, 숫자 등의 문자 집합들을 인식할 것 등의 기능을 갖는 알고리즘을 제안한다. 이를 위하여 한글 및 영숫자 필기 입력 방안에 대하여 연구하여 제안하였던 한영 혼용 문자 필기 글꼴을 사용하고[8], 문자의 필기는 하나의 문자 단위로 하며, 한글의 자소와 영문자, 숫자를 같은 수준에서 인식한 후, 모아쓰기에 의하여 한글 문자를 조합한다.

제안하는 인식 알고리즘의 흐름은 필기 된 획을 방향 코드열로 변환하여 인식한 후, 인식된 획의 번호와 획 사이의 상대 위치정보를 이용하여 계층적 상태 그래프 추적을 통하여 자소 또는 영숫자를 인식한다. 인식 알고리즘은 인식율을 높이기 위하여 한글의 제자원리에 따라 상태그래프를 계층화하였으며, 상태 그래프의 순방향 추적에 의한 자소 또는 영숫자 인식 이외에 유사한 획을 사용하는 한글 및 영문 혼용 인식에서 발생할 수 있는 문자들의 오인식을 줄이기 위하여 역추적에 의한 후처리를 포함한다.

본 논문은 획 인식 후의 자소 또는 문자의 인식에 초점을 맞추었으며, 따라서 상대적으로 비중이 약하며 새로운 방법이 아닌 기존에 제안하였던 전처리나 획 인식 알고리즘에 대하여는 간단한 소개로 대신하며, 주로 상태그래프의 구성에 의한 자소 또는 문자 인식 알고리즘에 대하여 기술한다. 또한 이후 알고리즘을 하드웨어화 할 수 있도록 전개하고자 한다. 논문의 구성은 2장에서 필기 글꼴과 획 인식에 대한 간략한 소개와 자소 및 문자 인식을 하기 위한 입력 정보의 표현을 다루며, 3장에서는 문자 인식을 위한 계층적 상태그래프의 구성 및 그래프 추적에 의한 자소 및 문자의 인식 알고리즘을 설명한다. 4장에서는 초소형 휴대형 정보기기에의 적용을 위한 효용성을 검증하는 실험을 다루고, 5장에서 결론을 맺는다.

Ⅱ. 문자의 글꼴과 입력 획의 표현

1. 입력 문자의 필기 글꼴

입력 획은 태블릿에 쓰여진 펜 데이터의 좌표열을 방향 코드열로 바꾸어 코드열 정합 알고리즘에 의하여 사전에 정의된 획으로 인식된다. 이 때 사용되는 한글과 영숫자 필기 글꼴은 초소형 휴대형 정보기기에의 적용을 위하여 영문자 PDA 들에서 사용되는 1획의 그래피티 글꼴처럼 약간의 변형과 인쇄체, 필기체, 대문자, 소문자들 중의 선택으로 이루어진 휴대형 정보기기의 문자 입력을 위한 필기 글꼴을 사용한다. 그림 1은 영문자 그래피티 글꼴을 나타내며, 그림 2는 한글과 영숫자 혼용을 위하여 제안된 필기 글꼴을 나타낸다.

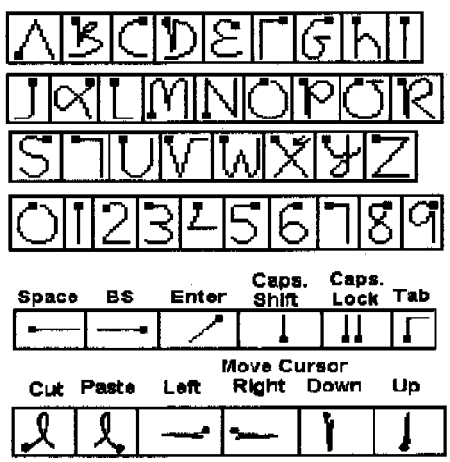
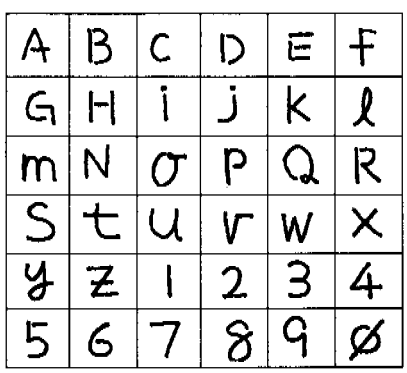
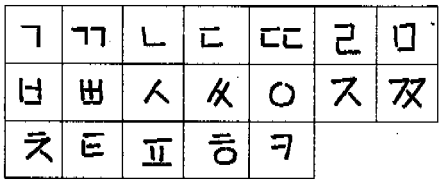


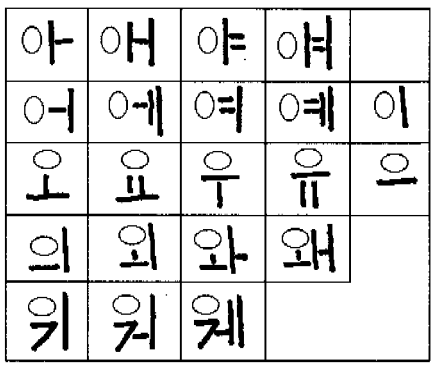
그림 1. 그래피티 영숫자 필기 글꼴



(a) 영-숫자 글꼴



(b) 한글 자음 글꼴



(c) 한글 모음 문자 필기 글꼴

그림 2. 휴대형 정보기기용 문자 인식 알고리즘의 필기 글꼴

그래피티 글꼴은 현재 상용 PDA 들에서 사용되고 있기는 하지만 글꼴 중 일부가 영문자와 숫자에서 똑같은 형태를 사용하고 있어서 입력 모드 전환이나 필기 영역 분리 등의 방법을 사용하는 것을 기본으로 하고 있다. 또한 한글 문자에 대한 고려는 전혀 없는 관계로 그래피티를 이용하여 한영 혼용 문자 인식을 시도하는 것은 불가능하다. 제안된 한영 혼용의 필기 글꼴은 이러한 문제를 상당 부분 해결하였으며, 어쩔 수 없이 나타나는 한글 모음에서의 단순한 글꼴에 의한 중복 형태는 인식 알고리즘에서 처리한다.

2. 입력 획의 인식과 정보의 표현

태블릿 상의 펜의 궤적으로부터 입력된 획의 글꼴은 최초에는 좌표점열의 형태로 입력되며, 이는 전처리에서 방향 코드열로 변환된다. 방향코드로의 변환은 획 인식의 효율을 높이기 위하여 코드열 정합 알고리즘을 사용하기 위함이다. 방향코드는 획의 궤적을 나타내는 특징 공간의 표현 외에도 획 사이의 상대위치 정보를 나타내는 목적으로도 사용되며 그림 3과 같다.

코드열 정합 알고리즘은 두개의 서로 다른 코드열 간의 차이를 거리로서 나타내주는 방법으로 LD(Levenstein Distance), WLD(Weighted Levenstein Distance), MWLD(Modified WLD), HWLD

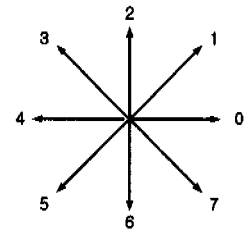


그림 3. 8방향 코드

(Hamming WLD), EWLD(Enhanced WLD), BWLD(Banded WLD) 등으로 잘 알려져 있으며[10][11][12][13][14][15], 본 논문에서는 BWLD 알고리즘을 사용한다. 두 코드열 간의 거리는 정합 메트릭 상에서 하나의 코드열을 또 다른 코드열에 일치시켜 나가는 과정에서 필요로 하는 삽입(insert), 삭제(delete), 대체(substitute)의 세 가지 변환의 횟수에 기초한다. LD는 단순한 변환의 횟수를 거리로 산출하며, WLD와 MWLD는 변환의 종류에 따라 가중치를 부여한 거리, HWLD는 코드간의 해밍거리를 이용한 방법이며, EWLD와 BWLD는 유사 코드의 변환시가 전혀 다른 코드의 변환시보다 짧은 거리가 되도록 변환되는 코드의 종류에 따라 가중치를 주는 방식이면서, VLSI 설계에 적합하도록 개선한 알고리즘이며, 특히 BWLD는 최적의 경로 주변만을 계산하도록 개선되어 두 코드간의 거리 계산 시간을 단축시킨 알고리즘이다.

WLD 정합 메트릭은 가로축이 표준 획의 코드열 패턴이 되며, 세로축이 입력 획의 코드열 패턴이 되는데, WLD에서는 '입력 코드열의 길이 × 표준 코드열의 길이' 만큼의 노드가 생성되지만, BWLD 알고리즘의 밴드(band) 매핑은 그림 4와 같이 대각선 방향으로 이루어져 3개의 노드상에서 최적의 경로 주변만을 검사하게 된다. 이들 3개의 노드는 각각 삽입, 삭제, 대체 변환의 경로에 대응하게 되며, 이들 노드에서는 삽입과 삭제, 대체 변환의 각각의 경로를 나타내는 국소거리(LD:Local Distance)를 계산하고, 최적의 경로를 결정하여 국소거리를 전체거리에 누적한다. 이것을 두 개의 코드열이 일치할 때까지 반복하면 코드열 간의 거리가 구해진다. 획 인식에서는 하나의 코드열을 입력 획으로 하고 또 하나의 코드열을 사전상의 표준획으로 하면 표준획 중 가장 유사한 획으로 인식하게 된다.

필기된 문자는 N개의 획으로 구성되며 각 획은 획 인식 코드(S_No) 이외에 획 사이의 4종류 상대적인 위치 정보(SS, SE, ES, EE)를 포함한다. 획의

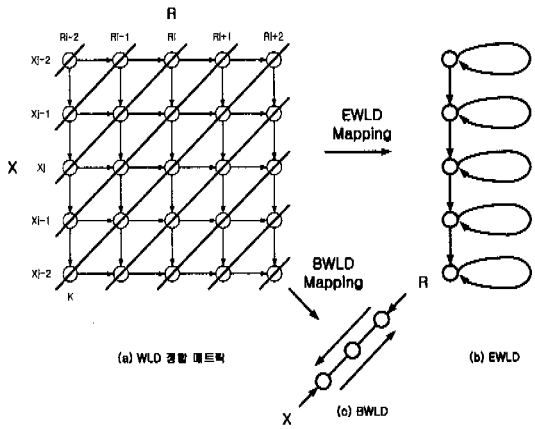


그림 4. WLD 정합 메트릭과 BWLD 매핑

위치 정보는 상대 위치 방향 정보를 사용하며, 그림 5와 같이 앞의 획과 현재 획의 시작점과 끝점 사이의 위치를 8방향 코드로 표현한다.

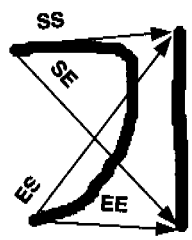


그림 5. 입력 획 위치 정보의 예

입력 획의 순서적인 표현은 그림 6과 같이 모든 방향의 정보를 포함하는 첫 획으로 시작하여 필기의 종료를 나타내는 마지막 획까지 N개의 획으로 구성된다. 획 시퀀스의 각 획의 정보는 인식된 획의 번호와 위치 정보를 포함하며, 첫 획의 위치 정보는 상태그래프 상에서 어떤 위치 정보를 갖는 노드와도 결합될 수 있도록 하기 위하여 모든 방향 성분을 1로 세트한다. 입력 획의 위치 정보 SS, SE, ES, EE는 상태그래프의 추적시 빠른 처리 시간을 위하여 각 8비트를 할당하고, 8비트 중 1비트만 세트되는 원-핫-코드(one-hot-code)를 사용하며, 각 방향의 위치 정보에 대한 원-핫-코드는 표 1과 같다.

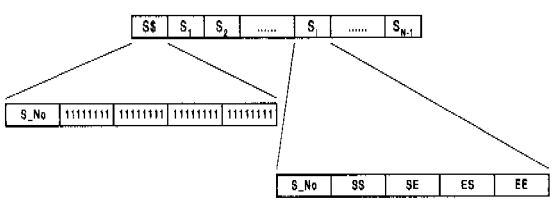


그림 6. 입력 획 시퀀스

표 1. 입력 획의 위치 정보 코드의 표현

8방향 코드	원-핫-코드
	76543210
0	00000001
1	00000010
2	00000100
3	00001000
4	00010000
5	00100000
6	01000000
7	10000000

입력 문자와 획은 <정의 1>로 나타내며, 각 획의 정보 표현은 <정의 2>와 같다.

<정의 1> 입력 문자와 획

1. 펜-다운(Pen-Down)과 펜-업(Pen-Up) 사이의 펜의 궤적을 획으로 하며 유한개의 표준 획 중 하나로 인식된다.

$$S\_No \neq 0(\text{null}) \quad (1)$$

2. 펜-업(Pen-Up) 후 일정 시간이 경과한 뒤 펜 다운이 발생하지 않을 경우 펜 샘플링을 중지하며, 이때까지의 획들로 1개 이상의 영-숫자 혹은 한글의 자소를 표현한 것으로 한다.
3. 입력 문자는 유한개의 획 정보를 갖으며 특별한 정보를 갖는 첫 획과 마지막 획이 있다.
4. 한글 모음은 반드시 자음 이후에 나타나며 자음은 2번 이상 반복될 수 있다.

<정의 2> 입력 획의 데이터 구조

1. 필기 순서에 따라 입력되는 N개의 획 중 i번째 획의 정보 S_i는 다음과 같다.

$$S_i = \{S\_No, SS, SE, ES, EE\},\ i=0...N-1 \quad (2)$$

여기서,

S_No : 인식된 표준획 번호

SS : *(i-1)*번째 획의 시작점과 *(i)* 번째 획의 시작점 사이의 8 방향 정보

SE : *(i-1)*번째 획의 시작점과 *(i)* 번째 획의 끝점 사이의 8 방향 정보

ES : *(i-1)*번째 획의 끝점과 *(i)* 번째 획의 시작점 사이의 8 방향 정보

EE : *(i-1)*번째 획의 끝점과 *(i)* 번째 획의 끝점 사이의 8 방향 정보

2. 획 사이의 상대위치 방향정보 *SS,SE,ES,EE*는 8방향 중 한 방향만 갖도록 다음의 조건에 따라 원-핫 인코딩(One-Hot Encoding)된다.

$$\sum_{m=0}^{7} S_i[SS_m] = \sum_{m=0}^{7} S_i[SE_m] = \sum_{m=0}^{7} S_i[ES_m] = \sum_{m=0}^{7} S_i[EE_m] = 1 \quad (3)$$

m : 방향 성분의 비트맵 인덱스

3. 첫 획, S$는 i=0일 때로서 모든 방향의 상대위치 방향 정보를 포함하도록 다음과 같이 표현된다.

$$\sum_{m=0}^{7} S_\$[SS_m] = \sum_{m=0}^{7} S_\$[SE_m] = \sum_{m=0}^{7} S_\$[ES_m] = \sum_{m=0}^{7} S_\$[EE_m] = 8 \quad (4)$$

Ⅲ. 상태 그래프를 이용한 문자 인식

필기 순서에 따라 입력된 획의 정보를 이용하여 미리 구성된 상태 그래프를 추적(Trace)하여 한글 자소 및 영-숫자로 인식된다. 상태 그래프는 두 개의 분리된 계층(Layer)으로 구성하며, 제1계층은 영문자와 숫자, 특수문자, 그리고 초성과 종성의 한글 자음을 인식하고, 제2계층은 복모음을 포함한 한글 모음을 인식한다. 각 계층의 구조는 <정의 3>, <정의 4>와 같다. 한글의 경우 제자 원리에 따라 자음과 모음 인식을 위한 그래프를 계층화하였다. 한글의 모음은 자음에 비하여 단순한 획의 조합으로 이루어지는데 비하여 영-숫자와의 유사도가 매우 크므로 자음과의 종속성을 두었으며 이는 온라인 문자 인식에서 획의 필기 순서가 인식에 매우 중요한 정보가 될 수 있기 때문이다. 자소 인식 상태 그래프를 구성하는 각 노드의 구조는 깊이와 폭이 정형화되어 있지 않는 그래프 구조를 갖는다.

<정의 3> 계층적 상태 그래프 및 종속성 정의

1. 계층적 상태 그래프는 2개의 분리된 계층으로 구성되며 다음과 같이 정의한다.

   Q : 제 1계층 그래프

   V : 제 2계층 그래프

2. 제 2계층은 반드시 제 1계층에 종속 된다.

<정의 4> 상태 그래프의 구조

1. 상태 그래프는 1개의 시작 노드 *q$* 가 존재하며, 추적 되는 깊이에 따라 수준(level)으로 구분한다. 수준 l 에는 유한개의 노드(node)들이 존재한다. l 번째수준에서 j 번째 노드의 표현은 다음과 같다:

$$q_{l,j} = \{S\_No, SS, SE, ES, EE, Ch\_Code, L\_Index, L\_Count\} \quad (5)$$

여기서,

S_No : 획 번호

SS, SE, ES, EE : 획의 상대위치 방향 정보

Ch_Code : 자소 코드, 한글 조합형 5비트, 혹은 ASCII 코드

L_Index : *(l+1)*번째 수준에서의 추적 시작 노드

L_Count : *(l+1)*번째 수준에서의 추적할 노드의 개수

2. 노드 *ql,j*의 위치 정보는 각각에 대하여 1개 이상의 방향 성분을 갖도록 다음과 같은 조건을 만족하여야 한다.

$$\sum_{m=0}^{7} q_{l,j}[SS_m] > 0$$
$$\sum_{m=0}^{7} q_{l,j}[SE_m] > 0$$
$$\sum_{m=0}^{7} q_{l,j}[ES_m] > 0$$
$$\sum_{m=0}^{7} q_{l,j}[EE_m] > 0 \quad (6)$$

3. $q_{l+1,j}[L\_Count]=0$ 일때, 종료 노드 q_{Final}, v_{Final}로 표현하며 반드시 Terminal 노드이다.

4. 그래프를 구성하는 중간 노드 $q_{i,j}$, $v_{i,j}$는 자소 코드에 따라 Terminal 노드와 Non-Terminal 노드로 구분된다. Terminal 노드는 그래프 탐색이 중지되었을 때 자소 조합이 이루어질 수 있으며 Terminal 노드만이 종속계층을 가질 수 있다.

Terminal Node, q_T, v_T : $q_{i,j}$ [*Ch_code*] 0 (7)

Non-Terminal Node, q_{NT}, v_{NT} : $q_{i,j}$ [*Ch_code*]=0 (8)

<정의 4>에서 각 수준의 노드들은 상위 노드들에서 전이될 수 있는 노드를 의미하며, 하나의 획에 다른 획, 또는 다른 위치 정보를 갖는 획들의 조합에 의한 자소 또는 문자의 구성을 의미한다. l 수준의 j번째 노드의 정보 중 *L_Index*는 다음 수준, 즉, $l+1$ 수준의 시작 노드의 위치를 가리키며, *L_Count*는 시작 위치로부터 몇 개의 노드가 자노드인가를

나타낸다. 이는 상태그래프의 저장과 추적을 단순하고 빠르게 하며, 메모리의 효율적인 사용과 함께 하드웨어화를 할 수 있도록 하기 위함이다.

노드에 있는 상대 위치 방향 정보는 그래프의 전이 허용을 나타내는 것으로 다음 입력 획이 일치할 때, 위치 정보가 일치하는 자노드로 전이가 이루어지도록 한다. 그래프의 전이는 <정의 5>의 조건에 따라 이루어진다. 전이는 하나의 문자나 자소를 인식한 후가 아니면 반드시 자신의 자노드로만 전이가 가능하며, 부노드가 하나인 자노드들은 획이 다르거나 아니면 위치정보가 다른 경우만 존재한다.

<정의 5> 계층 상태 그래프 전이를 위한 노드 조건

1. 수준 l 에서의 전이에 의하여 추적될 수준 $l+1$ 의 노드들의 범위는 다음과 같다.

$$0 \le L\_Index < J_{l+1}, 0 \le L\_Count < J_{l+1} \qquad (9)$$

J_{l+1} : 수준 $l+1$에서의 노드 총 개수

2. $q_{l-1,j} = q_{l-1,k}$ 인 두 노드 $q_{l,j}$와 $q_{l,k}$는 한 노드의 자노드들로 완전하게 구분되어야 하므로 다음 조건을 만족하여야 한다.

$$\begin{aligned} & q_{l,j}[S\_No] \neq q_{l,k}[Ch\_Code] \parallel \\ & \sum_{m=0}^{7} \left(q_{l,j}[SS] \oplus q_{l,k}[SS]\right)_m > 0 \parallel \\ & \sum_{m=0}^{7} \left(q_{l,j}[SE] \oplus q_{l,k}[SE]\right)_m > 0 \parallel \\ & \sum_{m=0}^{7} \left(q_{l,j}[ES] \oplus q_{l,k}[ES]\right)_m > 0 \parallel \\ & \sum_{m=0}^{7} \left(q_{l,j}[EE] \oplus q_{l,k}[EE]\right)_m > 0 \end{aligned} \qquad (10)$$

정의에 의한 한글 자소 및 영숫자 문자인식을 위한 계층적 상태 그래프는 그림 7과 같은 형태로 구성된다. 수준 l에서 추적될 $l+1$의 노드들은 중복될 수 있도록 함으로서 상태 그래프를 효율적인 구성이 가능하도록 하였다. 이는 일반적인 그래프 구조에서와 달리 하나의 자노드가 여러 개의 서로 다른 부노드를 가질 수 있도록 하여 노드의 수를 줄일 수 있음을 의미한다. 그러나 <정의 5>의 조건에 따라 서로 완전 구분되는 노드들에 한하여 추적 범위의 중복을 허용한다.

계층 상태 그래프를 통한 문자 인식은 입력된 펜 데이터로부터 얻어진 획과 위치정보를 이용하여 순

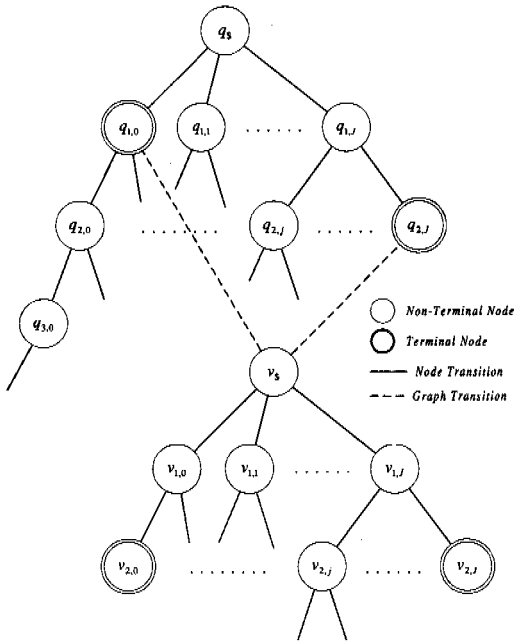


그림 7. 상태그래프의 구조와 표현

방향 그래프 추적으로 이루어지며 추적 실패에 따른 미인식과 오인식의 보정을 위한 역추적이 포함된다. 순방향 추적 알고리즘은 제1계층과 제2계층 모두 동일하며, <정의 3>에 따라 두 계층은 종속관계에 있다. 문자 인식을 위한 상태 그래프 추적은 <알고리즘 1>로 이루어진다. 추적 알고리즘은 계층 그래프를 선택한 후, 하나씩 입력 획을 받아들이며 획의 일치와 위치 정보의 일치를 판단한다. 다음 계층 그래프의 선택은 인식된 자소가 한글 초성일 경우만 제2계층으로 전이하며, 제2계층에서의 인식후는 반드시 제1계층으로 전이된다.

상태 그래프의 순방향 추적 실패에 의한 미인식과 오인식을 보정하기 위한 후처리 과정으로 역추적을 수행함으로써 후보 문자 인식을 시도할 수 있도록 한다. 이를 위하여 순방향 추적 과정의 이력 정보가 저장되도록 하며, 이를 위한 데이터 구조는 <정의 6>과 같다. 이력 정보의 저장은 계층 그래프(G)와 수준(Level), 그리고 노드 번호(Node)가 저장되며 이를 역추적하여 오인식을 줄이도록 한다.

<정의 6> 순방향 추적 이력 정보 데이터 구조

필기 순서에 따라 입력되는 N개의 획에 대하여 상태 그래프의 순방향 추적 이력 정보 H_i는 다음과 같다.

<알고리즘 1> 자소 인식 상태 그래프의 추적

자소 혹은 영-숫자 인식을 위한 상태 그래프의 추적 알고리즘은 다음과 같다.

입력 : N 개의 획 정보와 계층 그래프 Q 또는 V
출력 : 자소 노드 번호와 추적 이력 정보

```
Forward_Trace:
Set State Graph: G = {Q|V};
For Stroke Si and Node gi,j
begin
        If (Si[S_No] == gi,j[S_No]) then
            Find_Matched_Node:
            If ( (Si[SS] & gi,j[SS]) && (Si[SE] & gi,j[SE]) &&
                 (Si[ES] & gi,j[ES]) && (Si[EE] & gi,j[EE]) ) then
                Record_Trace_History(i, j,l);
                if ( i < N ) then
                    Next_Stroke_Index: i=i+1;
                    if (qi,j == qFinal)
                        Transition_to_Next_Graph: G = {Q|V};
                    else
                        Transition_to_Next_Level: l = l + 1;
                else
                    if ((gi,j == gTerminal) || (gi,j == gFinal))
                        End_of_Trace();
                    else
                        Trace_Fail(i);
                end if;
            else
                if (LAST_NODE(j))
                    Trace_Fail(i);
                else
                    Next_Node_Index : j = j + 1;
            end if;
        else
            if (LAST_NODE(j))
                Trace_Fail(i);
            else
                Next_Node_Index : j = j + 1;
        end if;
end for;
```

$$H_i = \{G, l, j, Ch\_Code\},\ i = 0...N-1 \quad (10)$$

여기서,

$G = \{Q|V\}$: 상태 그래프

l : i번째 획의 추적 상태 그래프의 수준

j : l 번째 수준에서 추적된 노드

Ch_Code : 추적된 노드의 자소 코드

그림 8은 순방향 추적 이력 정보 데이터의 구조를 나타낸다. 순방향 추적 이력 정보는 역추적의 중요한 정보이며 인식 결과를 갖고 있다. 그림 8의 H_2와 H_3의 예는 그래프 계층의 전이가 있음을 나타내며, 계층 전이가 이루어진 H_2의 자소 코드가 그래프 순방향 추적에 의한 인식결과가 된다.

역추적은 상태그래프의 순방향 추적에서 오인식이나 미인식이 발생하면 이루어지며 순방향 추적의 이력 정보를 이용함으로써 입력된 획들의 형상에 가장 근접한 문자 조합이 이루어지도록 한다. 오인식은 인식된 자소들로 문자를 조합할 때, 문자 사전상에 없는 문자를 만들거나 대표적으로 오인식되는 패턴들에 대한 문맥 검사 등에서 오인식으로 판단될 때이며, 미인식은 <정의 1>에 위배될 때와 그래

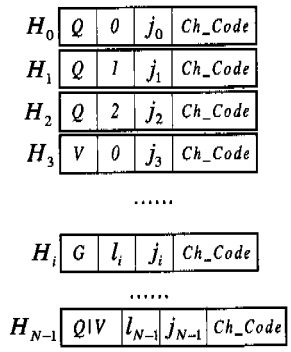


그림 8. 순방향 추적 이력 정보 데이터의 구조

프의 추적 중 터미널 노드가 아니면서 입력 획이 끝난 경우와 입력 획은 남아 있으나 종료 노드를 만난 때 등이다.

역추적의 방법은 <알고리즘 2>와 같으며, 동일 계층 내에서 역추적을 수행하고, 역추적 중 터미널 노드를 발견하면 그래프를 전이하고, 전이된 계층 그래프에서 다시 순방향 추적에 의한 후보 문자 인식이 이루어지게 된다.

<알고리즘 2> 자소 인식 상태 그래프의 역추적

상태 그래프의 순방향 추적 실패에 따른 역추적은 다음과 같다.

입력 : N개의 획에 대한 순방향 추적 이력 정보와 계층 그래프 Q 또는 V
출력 : 역추적 노드 번호

```
Backward_Trace:
Set State Graph: G = H_{i-1}[G];
While(i>0)
begin
        Rewind_Stroke_Index: i = i 1;
        If (H_i[Ch_Code] != NULL) then
          Change_Graph: G = {Q|V};
          Forward_Stroke_Index : i = i + 1;
          Set_Level_Zero : l = 0;
          Forward_Trace();
        end if;
end;
```

본 논문에서 제안하는 알고리즘을 이용한 휴대형 정보기기용 온라인 문자인식 시스템은 그림 9와 같이 구성될 수 있다.

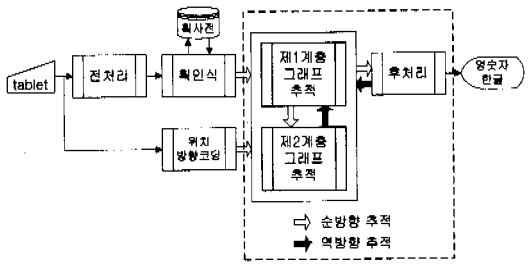


그림 9. 제안된 알고리즘을 이용한 온라인 문자인식 시스템의 구성

Ⅳ. 실험 및 고찰

본 논문에서 제안한 알고리즘의 인식 실험 대상 문자는 영숫자와 한글이며, 영문자, 숫자 한글을 구분하지 않고 혼용하여 필기하는 것으로 실험하였다. 온라인 문자 인식의 기초 정보는 획 정보(Stroke Information)로서 태블릿 상에서 펜-다운(Pen-Down)과 펜-업(Pen-Up) 구간의 궤적을 획으로 인식하고 획 사이의 상대적인 위치 정보를 방향 코드로 표현한 것이다. 획은 2차원 좌표점열의 펜 샘플을 8방향 코드열로 변환한 후 고속 코드열 탄력정합 방법인 BWLD 알고리즘을 사용하여 획으로 인식한다. 획 사전은 인식 대상으로 삼는 문자에 쓰이는 형태에 따라 구성되어야 하며, 획의 종류는 문자의 다양성과 흘려쓰기의 허용 정도에 의하여 달라지고, 학습 기능에 따라서도 달라진다. 제안된 인식 알고리즘에서 사용한 획의 종류는 정의된 필기 글꼴에서 사용되는 획으로 그림 10과 같이 한글 자소 및 영숫자 필기 글꼴을 대상으로 추출한 38종 51개로 정의하였다.

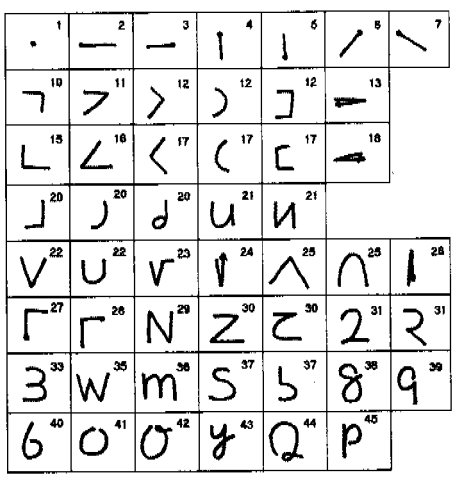


그림 10. 필기 글꼴 기본 획

실험을 위하여 구성된 상태그래프는 한글과 영문자, 숫자 등에 대하여 약 50,000여자의 필기 글꼴을 분석하여 생성하였으며, 학습에 의한 자동 생성시 변형이 너무 다양하며, 같은 정보로 서로 다른 문자를 발생시키는 경우가 많아서 정형화하여 상태그래프를 최적화하였다. 인식 실험은 20인의 피실험자를 대상으로 약 30분 정도의 연습 후, 한글 잦기 순위 522자[16]와 영문 알파벳, 숫자를 섞어서 제시하고 태블릿 상에 필기 하도록 하여 실험하였으며, 그 결과는 표 2와 같다. 실험 결과, 97% 이상의 대체로 만족할 수준의 인식율을 보였으며 한글의 경우 초-중성으로 구성된 문자에 비하여 초-중-종성으로 구성된 문자에서 높은 인식율을 보였다. 오인식 또는 미인식후 역추적에 의한 재인식에서 대표적인 온라인 문자 인식 오류의 전형적인 패턴인 'ㄹ'을 포함하는 문자의 경우 '근'으로 인식되는 일이 개선되었다. 이는 오인식과 미인식이 함께 나타나는 경우로, 하나의 자소 'ㄹ'을 하나의 문자 '근'으로 오인식한 후 획이 남게 되는 경우이다. 또 다른 전형적인 오인식과 미인식이 나타나는 경우로는 한글 자음 'ㄷ'과 영문자 't'의 경우와 한글 자음 'ㅌ'이 'ㄷ'로, 'ㅋ'이 'ㄱ'로 인식되는 경우가 종종 있으나 이는 다음 획이 남게 되는 경우가 대부분인 관계로 역추적에 의한 후처리에서 인식 가능하다.

오인식은 필기형태에 따라 숫자 '01'과 한글 '이', '4'와 '니'와 같은 유사한 형태의 문자들에서 발생되었으며, 한글의 '지'가 '거', '정'이 '갱'으로 인식되는 경우가 있었다. 이는 한글 자음 'ㅈ'이 들어가는 초성이 자음 'ㄱ'과 종모음이 결합되는 형태로의 오인식으로, 이에 대한 후처리 방법의 보완이 필요하다. 그러나 전체적인 문자 인식의 결과를 보면 획 인식 실패, 즉 획의 변형 등에 의한 획 오인식이 자소 또는 문자 조합의 오인식에 결정적인 역할을 하는 것으로 나타났다. 대표적인 획 오인식의 종류는 한글 자음 'ㅇ'과 숫자 '6'의 경우로 이는 문맥을 조사하여 획을 바꾸어주는 후처리가 필요하다.

표 2. 실험 결과 평균 인식율

구분	정인식율	오인식율	미인식율
영문자	98.0%	1.8%	0.2%
숫자	98.2%	1.8%	0%
한글	97.0%	2.5%	0.5%
평균	97.73%	2.03%	0.35%

제안한 문자인식 알고리즘은 휴대형 정보단말기를 위한 것으로 어느 정도 필기 글꼴에 제한을 두고 있다. 실험을 통하여 제안된 필기 글꼴에 대한 평가는 정량적인 평가와 정성적인 평가를 시도하여, 정량적으로 약 4% 포인트 정도의 인식율을 높이는 결과를 가져왔으며, 이는 애매성을 갖는 한글과 영숫자의 필기 형태를 서로 다르게 정의한 결과라고 할 수 있다. 또, 정성적인 평가로, 제안된 필기 글꼴로 인한 필기의 편의도 평가를 위하여 임의의 문자를 입력토록 한 후 필기 글꼴에 대한 만족도를 조사한 결과는 대체로 좋은 반응을 얻었으며, 그래피티 문자를 아는 실험자는 매우 만족하는 결과를 보였다. 특히 영문자와 숫자, 그리고 한글의 모드 전환이나 영역 구분 없이 혼용 필기하는 글꼴 및 알고리즘에 대하여 높은 만족도를 보였다.

실험에 의한 물리적인 결과로서 속도와 크기에 있어서도 만족할 만한 결과를 보여서, 인식 속도는 사용자의 필기 속도에 전혀 지장을 주지 않는 상태이다. 실제로 그래프를 추적하는데 소요되는 연산량은 노드 하나의 전이를 결정하는데 단지 4회의 AND 논리 연산만을 필요로 하는 등 전체적인 연산량과 연산의 복잡성이 매우 적었다. 또한 구현된 알고리즘의 실제 크기는 전처리와 인식 알고리즘, 상태 그래프, 획 사전 및 인식 데이터를 포함하여 전체 크기가 총 56KB로 메모리의 요구가 크지 않아 휴대형 정보기기에의 적용이 적합함을 확인하였다.

V. 결 론

본 논문에서는 휴대형 정보기기의 입력 장치로서의 온라인 필기 문자 인식 알고리즘을 제안하였다. 휴대형 정보기기는 그 특성상 데스크탑 컴퓨터와는 달리 CPU의 성능, 속도, 메모리 등 자원의 제약을 피할 수 없으나, 제안된 알고리즘은 복잡한 연산이 수반되지 않으면서도 그 크기가 매우 작은 알고리즘으로 PDA 등의 휴대형 정보기기에 적합함을 확인하였다. 또한 한글과 영문자, 숫자를 혼용하여 사용하는 우리의 문자 생활을 고려할 때, 사용자의 불편을 최소로 줄일 수 있는 알고리즘으로, 인식 대상 문자인 한글과 영문자, 숫자에 대하여 입력 모드 전환이나, 필기 영역의 구분 없이 단일 알고리즘으로 이들을 구현할 수 있었다. 반면, 이러한 편의를 위하여 일상 생활에 사용하는 문자 형태 그대로를 인식하지 못하고, 인쇄체와 필기체, 대문자와 소문자

를 섞어 정형화된 필기 글꼴을 사용하게 되었으나, 글꼴의 인지도는 매우 높으며 짧은 시간의 연습으로 충분히 높은 인식율로 사용할 수 있었다.

인식율 실험 결과, 영문자와 숫자는 98%, 한글은 97%의 인식율을 얻었으며, 숙달된 사용자는 100% 가까운 인식율을 보여 유용함을 확인하였다.

본 논문의 알고리즘은 이후 제스춰를 포함한 제어문자로의 인식 대상 확대와 그래프 탐색 방법의 개선이 필요하고, 오인식 및 미인식을 줄이기 위한 방법으로 문맥 검사 등에 의한 후처리의 보완작업과 알고리즘의 VLSI화를 위한 설계를 진행하고 있으며, 현재 x86과 ARM CPU를 사용하는 PDA에의 구현이 완료되었다.

참 고 문 헌

[1] 한메소프트, "한글 입력 방법", 한메 한글 for PalmPilot-Quick Reference Guide, 1997.

[2] 성태진, 방승양, "문자조합 규칙 학습에 의한 온라인 한글 인식", 한국정보과학회 논문지 제20권, 제3호, pp. 305-316, 3, 1993.

[3] 권오성, 권영빈, "스트링 정합 방법에 기반한 온라인 자소 인식", 한국정보과학회 논문지 제21권, 제5호, pp. 750-756, 5, 1994.

[4] 신봉기, 김진형, "자소 탐색에 기반한 온라인 한글 인식", 한국정보과학회 논문지 제23권 B편, 제11호, pp. 1135-1144, 11, 1996.

[5] 김찬우, 김병만, 강오한, "퍼지 방향코드와 확장 은닉 마르코프 모델을 이용한 온라인 인식", 한국정보과학회 논문지, 제24권, B편, 제5호, pp. 551-560, 5, 1997.

[6] US Robotics, "User's Guide for PalmPilot",

[7] 장봉선, "한글 풀어쓰기 교본", 한풀 문화사, 1989

[8] 홍성민, 국일호, 조원경, "휴대형 정보기기의 한글 및 영숫자 필기 입력 방안", 대한전자공학회 논문지, 제35권, T편, 제3호, pp. 285-292, 12, 1998.

[9] 신봉기, "온라인 필기 모형화를 위한 은닉 마르코프 모형 기반의 통계적 방법론", 한국과학기술원 박사학위 청구논문, 1995

[10] Eiichi Tanaka, Tamotsu Kasai, "Synchronization and Substitution Error Correcting Codes for the Levenstein Metric", IEEE Trans. Information Theory, Vol. IT-22, No. 2, pp. 156-176, 3, 1976.

[11] Martin H. Ackroyd, "Isolated Word Recognition Using the Weighted Levenstein Distance", IEEE Trans. ASSP, Vol. ASSP-28, No. 2, pp. 243-244, 4, 1980.

[12] 박종진, 김은원, 조원경, "MWLD 알고리즘을 이용한 문자열 정합 1차원 Bit-Serial 어레이 프로세서의 설계", 대한전자공학회 논문지, 제29권, B편, 제2호, pp. 1-8, 2, 1992.

[13] 김현우, 홍성민, 조원경, "문자인식 시스템을 위한 코드열 정합에관한 연구", 대한전자공학회 추계종합학술대회 논문집, 제15권, 제2호, pp. 204-207, 11, 1992.

[14] 국일호, 홍성민, 조원경, "EWLD 알고리즘을 이용한 코드열 정합 프로세서의 설계", 대한전자공학회 논문지, 제31권, A편, 제4호, pp. 127-135, 4, 1994.

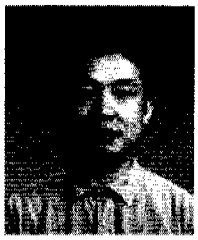
[15] 국일호, 홍성민, 조원경, "BWLD 알고리즘을 이용한 코드열 정합 프로세서의 설계", 대한전자공학회 1995년도 CAD 및 VLSI 설계연구회 학술발표회 논문집, 5, 1995

[16] 한글 기계화 연구소, "한글 잦기순위 522자", 한글 기계화 연구소, 1975

홍 성 민(Sungmin Hong)

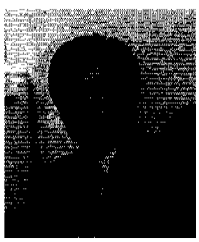
1983년 : 경희대학교 전자공학과 졸업 공학사,
1985년 : 경희대학교 대학원 전자공학과 졸업 공학석사
2000년 : 경희대학교 대학원 전자공학과 졸업 공학박사
1989년~1993년 : 충남전문대학 전자계산기과 조교수
1993년~현재 : 여주대학 전자과 부교수
<주관심 분야> 문자인식, 패턴인식, 디지털 회로 설계

국 일 호(Ilho Kook)



1987년 : 경희대학교 물리학과 졸업 이학석사
1989년 : 경희대학교 대학원 전자공학과 졸업 공학석사
2001년 : 경희대학교 대학원 전자공학과 졸업 공학박사
1999년~현재 : ㈜앤스랩 대표
2000년~현재 : 경희-다반 ASIC 설계 연구소 상임연구원
2001년~현재 : ㈜코덱실 공동대표
<주관심 분야> 멀티미디어 코덱 알고리즘 및 반도체 설계, IP 및 SoC 설계 검증, PDA 및 문자인식

조 원 경(Wonkyung Cho)



1971년 : 경희대학교 전자공학과 졸업 공학사
1973년 : 한양대학교 대학원 전자공학과 졸업 공학석사
1986년 : 한양대학교 대학원 전자공학과 졸업 공학박사
1978년~1980년 : 경남대학교 전자공학과 전임강사
1980년~현재 : 경희대학교 전자정보학부 교수
1988년~1989년 : 미국 오레곤주립대학교 교환교수
<주관심 분야> 디지털 회로 설계, VLSI 설계, 멀티미디어 코덱 설계, 인공지능