

이화여자대학교 대학원  
2001 학년도  
석사학위 청구 논문

데이터베이스에 근거한  
지능형 키보드에 관한 연구

전자상거래학과  
이계숙  
2002

데이터베이스에 근거한  
지능형 키보드에 관한 연구

이 논문을 석사학위 논문으로 제출함

2002년 5월

이화여자대학교 정보과학대학원

전자상거래학과 이 계 숙

이 계 속 의 석사학위 논문을 인준함

지도교수 용 환 승 \_\_\_\_\_

심사위원 \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

이화여자대학교 정보과학대학원

# 목 차

논문 개요.....	V
I. 서론.....	1
1.1 연구의 배경 및 목적 .....	1
1.2 연구 내용 .....	2
II. 관련 기술 및 연구 동향 .....	4
2.1 한글 코드 처리 기법 .....	4
2.2 MOBILE 기기에서의 한글 입력 방식 .....	6
2.3 구조적 가상 키보드에 관한 연구 .....	8
III. 데이터베이스에 근거한 지능형 키보드 .....	9
3.1 데이터베이스에 근거한 지능형 키보드 정의 .....	9
3.2 데이터베이스에 근거한 지능형 키보드의 설계 .....	10
3.2.1 자모의 구성 .....	11
3.2.2 지능형 키 도시 기능 .....	12
3.2.3 후입력 자동 완성 기능 .....	15
3.3 시스템 구현 환경 .....	16
3.4 데이터베이스에 근거한 지능형 키보드 알고리즘 .....	17
3.4.1 Overall Architecture.....	17
3.4.2 프로그램 순서도 .....	18
3.4.3 입력 파라미터 .....	21
3.5 데이터베이스에 근거한 지능형 키보드의 특징 .....	21

VI. 성능 평가.....	24
4.1 성능 평가 프로그램 알고리즘 .....	24
4.2.1 프로그램 순서도 .....	24
4.2.2 입력 파라미터 .....	26
4.3 샘플 단어 집합 .....	26
4.4 성능 평가 결과 .....	28
4.3.1 일반 키보드 방식과 데이터베이스에 근거한 지능형 키보드 방식 의 키 개수 비교 .....	28
4.3.2 일반 키보드 방식과 데이터베이스에 근거한 지능형 키보드 방식 의 키 선택 횟수 비교 .....	30
4.3.3 데이터베이스에 근거한 지능형 키보드에서 이중 모음과 이중 자 음 분리 시 성능 평가 .....	32
V. 결 론.....	35
5.1 연구 결과 및 의의 .....	35
5.2 향후 연구 과제 .....	37
참 고 문 헌.....	38
ABSTRACT .....	40

# 그림 목 차

[그림 2.1] 문자 인식 과정 .....	7
[그림 3.1] 표 3.1 과 같은 단어 집합이 주어졌을 때 키보드 변화 : .....	10
[그림 3.2] 데이터베이스에 근거한 지능형 키보드의 초기 키보드 .....	12
[그림 3.3] 지능형 키보드 입력 과정 예 .....	15
[그림 3.4] 후입력 자동 완성 기능 예 .....	16
[그림 3.5] SDI 형 MFC 프로그램 구조 .....	17
[그림 3.6] 데이터베이스에 근거한 지능형 키보드 블록 다이어그램 .....	18
[그림 3.7] 데이터베이스에 근거한 지능형 키보드의 순서도 .....	20
[그림 4.1] 성능 평가 프로그램 순서도 .....	25
[그림 4.2] 단어 집합의 크기에 따른 4,5 음절 단어 비율 .....	27
[그림 4.3] 단어 집합 크기 변화에 따른 키 버튼 개수 최대값 비교 .....	28
[그림 4.4] 단어 집합 크기 변화에 따른 키 버튼 개수 평균값 비교 .....	29
[그림 4.5] 단어 집합 크기 변화에 따른 키 선택 횟수 최대값 비교 .....	30
[그림 4.6] 단어 집합 크기 변화에 따른 키 선택 횟수 평균값 비교 .....	31
[그림 4.7] 이중 자음과 이중 모음 분리 시 키보드 형태 .....	32
[그림 4.8] 이중 모음과 이중 자음 분리 시 키 개수 비교 .....	33

## 표 목 차

[표 2.1] 한글 음소 .....	5
[표 2.2] 한글 코드 분류 .....	5
[표 3.1] 그림 3.1의 입력 단어 집합 .....	10
[표 3.2] 시스템 구현 환경 [표 3.2] 시스템 구현 환경 .....	16
[표 3.3] 입력 파라미터 .....	21
[표 4.1] 성능 평가 프로그램 입력 파라미터 .....	26
[표 4.2] 단어 집합 크기별 음절 수 비교 .....	27

## 논문 개요

최근 무선 인터넷과 정보 통신 기술의 발달로 정보 기기와 모바일 단말기의 크기가 점차 소형화 되어가고 있다. 특히 휴대폰과 PDA(Personal Digital Assistant)의 대중화로 소형 단말기를 통한 정보의 입력 횟수는 날로 증가하고 있는 추세이다. 이와 같은 모바일 단말기는 휴대하면서 사용하기 때문에 기존의 일반 컴퓨터에서 사용하는 키보드를 그대로 사용할 수 없고, 자체적으로 내장된 문자 입력 시스템이 있지만 버튼 수의 제한과 버튼 크기의 제한으로 인하여 입력의 효율성에서 한계를 가지고 있다.

본 논문은 일반 키보드를 사용할 수 없는 모바일 단말기에서 이미 데이터베이스 형태로 저장되어 있는 항목을 입력할 때, 키 버튼 선택 영역면과 키 선택 횟수면에서 효율성을 가지는 새로운 소프트웨어 키보드 입력 방식을 제안하였다. 입력하고자 하는 단어의 집합이 주어졌을 때 주어진 단어 입력에 필요한 최소한의 키 버튼만을 도시함으로써 키 버튼 선택 영역을 축소하고, 단어를 입력함에 있어서의 신속성을 위해 몇 번의 키 버튼 선택 후 입력 가능한 단어들의 수가 제한적일 때 후입력 자동 완성 기능을 제공함으로써 키 버튼 선택 횟수를 최소화할 수 있다.

기존의 입력 방식과 본 논문에서 제안한 지능형 입력 방식의 비교를 통하여 다양한 크기의 단어 집합에서 키 버튼 수와 키 버튼 선택 횟수에 대한 성능 평가를 수행한 결과, 본 논문에서 제안한 지능형 입력 방식에 의한 키 버튼 수와 키 버튼 선택 횟수가 감소되었음을 입증하였다.

본 연구 결과를 통하여 데이터베이스 형태로 저장된 단어 집합을 소프트웨어 키보드 형태로 입력하고자 하는 모바일 정보 단말기에서 입력 시스템의 성능 향상을 기대할 수 있다.



# I. 서론

## 1.1 연구의 배경 및 목적

컴퓨터 주변기기 중 성능과 기능적으로 큰 변화 없이 개선되지 않은 것을 꼽으라 하면 단연코 키보드일 것이다. CPU, 그래픽 카드 등이 보다 빠른 속도와 성능 향상을 위해 꾸준히 개선되고 발전되어 온 반면 키보드의 경우에는 특별히 성능이나 기능이 개선되거나 발전하지 않았다[2].

더구나 한글의 특성상 한글 자판의 구성에 관해서는 계속 해결해야 할 문제로 남아 있다. 영어를 포함한 라틴 계열의 문자는 하나의 자소가 도형으로 표현될 때 일정한 크기로 쓰여지는 반면 한글은 2 개 또는 3 개, 경우에 따라서는 5 개까지의 자소들로 구성된다. 즉 영어를 기계화 할 때는 입력, 출력, 처리의 단위가 모두 같아 컴퓨터에서 사용하는 코드는 자소 단위로 코드를 부여하면 된다. 그러나 한글은 자소 단위로 입력하고 음절 단위로 모아서 출력하도록 되어 있다. 결국 한글 코드는 자소 단위와 음절 단위의 의미를 동시에 갖고 있어야 입출력 및 처리가 효율적인 코드가 된다[1].

한글 키보드에서 한글 자소의 종류 문제는 한글을 구성하는 자소 수를 적게 분해하면 키보드상의 자소 키 수가 많아지고 분해하는 수가 많아지면 결합 논리상에 문제가 생긴다. 한글의 기본 자수는 24 개지만 한 문자가 분해하여 이룬 자소군이 다시 동일 문자를 재현할 수 있어야 한다. 따라서 한글 자판의 구성 문제는 자소의 빈도수에 따른 키의 배열 문제, 입력 자모의 수를 몇 개로 할 것인가의 문제, 빈도수가 높은 복모음과 복자음을 하나의 키로 할 것인가에 관한 문제를 모두 고려해야 한다[3].

최근의 키보드는 다양한 종류가 출시되고 있다. 그것은 무엇보다도 PDA

와 휴대폰의 대중화 때문이다. PDA, 휴대폰과 같은 기기는 휴대하면서 사용하고 크기가 작기 때문에 기존에 일반 컴퓨터에서 사용하던 키보드를 그대로 사용할 수 없다. 그래서 이들 기기는 자체적으로 문자를 입력할 수 있는 시스템이 내장되어 있다. 그러나 기기의 크기가 작기 때문에 내장된 입력 시스템이 효율적이지 않은 것이 사실이다[2].

현대 정보화 사회에서 처리해야 할 정보의 양이 기하급수적으로 증가함에 따라서 필요한 정보를 효과적으로 입력 및 출력하고 분석 처리하는 능력이 요구되고 있다[4]. 더구나 정보 처리의 속도는 CPU 속도가 아닌 사용자의 입력 작업 속도에 달려 있다. 많은 양의 정보를 빠른 시간 내에 입력하기 위해서는 인체 공학적인 디자인 측면과 키보드를 두드릴 때의 매커니즘에 대한 개선이 이루어지는 하드웨어 키보드에 대한 연구뿐만 아니라 소형화 되어가는 정보 기기에 적합하게 최소의 키 버튼을 제공하고, 키 버튼 선택 횟수를 줄일 수 있는 지능적인 소프트웨어 키보드에 대한 연구가 필요하다.

## 1.2 연구 내용

본 논문은 하드웨어 키보드를 사용할 수 없는 정보 기기에서 한글을 입력할 때 기존의 PC 키보드를 그대로 화면에 도시하는 소프트웨어 키보드가 가지는 키 버튼 선택 영역면과 키 선택 횟수면에서의 단점을 보완하기 위하여 첫째, 키 버튼 선택 영역을 고려하여 입력하고자 하는 단어의 집합이 주어졌을 때 주어진 단어 입력에 필요한 최소한의 키 버튼만을 도시하는 기법을 제안한다. 둘째, 단어를 입력함에 있어서의 신속성을 위해 키 버튼 선택 횟수를 최소화하는 기법을 제안한다. 주어진 단어 집합을 입력할 때 필요한 키 버튼 수의 감소로 인한 키 버튼 선택 영역의 축소와 키 버튼 선택 횟수의 감소로 인한 키 입력 시간 단축의 효과를 기대할 수 있다. 또한 본 논문에서 제안하는 입력 방식의 효율성을 검증하기 위하여 2000 개의 샘플 단어 집합을 추출

하고 단어 집합의 크기를 변경하면서 기존의 입력 방식과 본 논문에서 제안하는 데이터베이스에 근거한 지능형 키보드 알고리즘의 성능을 평가하였다.

본 논문의 구성은 다음과 같다. II장은 한글 코드에 관한 기존의 연구와 여러 가지 정보 기기에서의 한글 입력 방식에 대해 기술한다. III장에서는 주어진 단어 집합을 효율적으로 입력할 수 있는 데이터베이스에 근거한 지능형 키보드 방식을 제안하고, IV장에서는 제안한 입력 방식에 대한 성능 평가를 수행하며, 마지막으로 V장에서 결론과 함께 앞으로의 연구 방안에 대해 기술한다.

## II. 관련 기술 및 연구 동향

본 장에서는 컴퓨터에서 한글을 처리하는 방법에 대한 연구를 통해 한글 코드의 종류와 특성을 알아 보고, 일반 컴퓨터에서 사용하는 하드웨어 키보드 형태가 아닌 한글 입력 형태와 모바일 정보 통신 기기에서의 한글 입력 방식에 대한 고찰을 통해 현재까지의 연구 동향과 기반 기술에 대해 살펴보고록 한다.

### 2.1 한글 코드 처리 기법

컴퓨터는 영문을 사용하는 문화권에서 개발되어 영문자 하나를 표현하는 단위를 바이트로 구분하고 정보를 처리할 수 있는 최소 단위로 만들었다. 한글은 이와 다르게 초성, 중성, 종성 세 개의 글자가 2 바이트 속에 스며들어 있으므로 글자를 처리할 때 각 자소 단위로 처리할 수 있는 것이 아니라 2 바이트 단위로 처리를 해야 한다. 또한 통신을 위해서 사용되는 패리티 체크를 다루는 공간도 한글 처리와는 별도로 고려해야 한다[5].

현대 국어에서 사용되는 음소는 자음과 모음 그리고 복자음, 복모음, 쌍자음이 있고 초성에 사용되는 자음은 19 가지, 중성에 사용되는 모음은 21 가지, 그리고 종성에 사용되는 자음은 27 가지가 있는데 이를 표로 정리하면 표 2.1 과 같다.

[표 2.1] 한글 음소

구분	내용
초성	ㄱ, ㅋ, ㆁ, ㄷ, ㅌ, ㄹ, ㅁ, ㅂ, ㅅ, ㅆ, ㅇ, ㅈ, ㅊ, ㅊ, ㅋ, ㆁ, ㅍ, ㅎ
중성	ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ
종성	ㄱ, ㅋ, ㆁ, ㄷ, ㅌ, ㄹ, ㅁ, ㅂ, ㅅ, ㅆ, ㅇ, ㅈ, ㅊ, ㅊ, ㅋ, ㆁ, ㅍ, ㅎ

표 2.1 에서 나열한 한글 음소를 사용하여 음절을 구성하고 이러한 한글을 컴퓨터에서 처리하는 방식에 대한 표준화의 지연으로 여러 종류의 한글 코드들이 사용되어져 왔다. 한글 1 음절을 표현하는데 필요한 바이트의 수와 조합 원리, 8 번째 비트의 사용 여부에 따라 분류한 한글 코드의 종류와 특징은 표 2.2 와 같고[5,6], 2 바이트 완성형 코드를 기본으로 완성형 코드에서 표현할 수 없었던 8,822 자를 추가한 확장 완성형 코드가 현재의 표준으로 제정되어 사용되고 있다.

[표 2.2] 한글 코드 분류

바이트수	한글 코드	특징
1 바이트	7 비트 완성형	사용 가능 한글 1500 자, 연구 부족
2 바이트	2 바이트 조합형	국제 표준 규약 준수 곤란
	2 바이트 완성형	모든 한글 표현 곤란
	2 바이트 확장 완성형	현재의 표준
	유니 완성형	자소 정보 분석 가능
3 바이트	3 바이트 조합형	필요 이상의 종성 바이트 사용, 기억 장소 낭비
N 바이트	N 바이트 조합형	문자 길이 가변적, 영어와 충돌
	유니 조합형	문자 길이 가변적

영문자와 다르게 한글은 초성, 중성, 종성으로 한 음절을 구성한다. 따라서 키보드를 사용해서 한글을 입력시키는 알고리즘의 경우 입력되는 문자열

을 모아 쓰기로 표현된 한 글자로 인식을 결정하기 위해서 복잡한 오토마타를 만족시키는 알고리즘이 개발되었다[5]. 또한 한 음절로부터 이를 구성하는 초성, 중성, 종성 음소를 분석하는 알고리즘도 개발되었다. 이러한 알고리즘에는 한글 코드 변환 알고리즘이 사용되는데, 한글 코드 변환은 여러 가지 한글 코드간의 대응 문자로 변환하는 것으로 논리적 특성에 따라 아스키 문자, 한글 문자, 한자, 특수 문자 등으로 그룹화한 후 그룹화 된 문자를 각각의 테이블로 구성하여 처리한다. 즉, 순차적으로 구성된 테이블을 이진 탐색 기법으로 탐색하여 대응 코드로 변환한다[6].

## 2.2 Mobile 기기에서의 한글 입력 방식

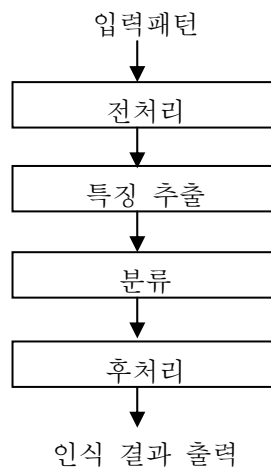
최근 정보 통신 기술과 무선 인터넷 기술의 발달로 휴대폰과 PDA(Personal Digital Assistant)에서의 한글 입력이 급속히 늘어나고 있다. 휴대폰과 PDA 와 같은 모바일 정보 단말기에서 한글 입력 방식은 전화기 자판을 통한 입력 방식과 화면 키보드 방식, 그리고 펜을 통한 입력 방식이 대표적이다.

전화기 자판을 통한 입력 방식은 휴대폰에 있는 0~9 까지의 10 개의 숫자 버튼과 2 개의 특수 버튼(\*,#)을 포함하는 12 개의 버튼을 이용해서 쉽고 빠르게 한글을 입력할 수 있는 방식이다. 이 12 개의 버튼으로 67 개의 한글 자모를 입력하기 위해서는 동일한 버튼에 여러 개의 자음과 모음을 배정해야 한다. 이때 쌍자음과 격자음과 같이 가족 관계에 있는 자모를 하나의 버튼에 배치하는 게 일반적이다. 예를 들면, 「ㄱ, ㄲ, ㅋ」 과 같이 가족 관계에 있는 자모를 하나의 버튼을 통해 입력한다. 이와 같은 대표적인 방식은 삼성전자의 천지인 방식과 LG 전자의 나랏글 2000 이 있다. 천지인 방식은 단자음을 세 번 눌러서 쌍자음을 입력하고, 격자음은 두 번 눌러서 입력한다. 중성은 「·, -, |」를 합성해서 입력하고 다음 글자를 입력하기 위해

서는 2초동안 기다려야 한다. 나랏글 2000의 경우는 쌍자음 변형을 위해서는 「#」 버튼을, 격자음 변형을 위해서는 「\*」 버튼을 이용한다. 모음을 입력하는 방식은 「ㅏ」 「ㅑ」 「ㅓ」 「ㅕ」를 각각 동일한 버튼에 배정하고 복모음은 「\*」 버튼을 이용해 입력한다[8].

화면 키보드는 일반 PC에서 사용하는 키보드와 동일한 키보드를 화면에 나타내어 화면의 키보드를 치면 실제 키보드를 친 것과 같은 효과를 내주는 소프트웨어 키보드 방식이다.

펜을 통한 입력 방식은 LCD와 입력 장치인 태블릿(Tablet)을 결합한 디스플레이 장치가 있어서 태블릿에 펜으로 글씨를 쓰면 태블릿에서 펜의 위치를 인식한 다음 운영 체제가 펜의 움직인 자취를 화면에 보이고, 동시에 이 자취를 가지고 필기된 글씨를 인식하여 문자로 표시하는 방식이다[9]. 문자 인식 과정은 입력 과정에서 발생한 시간적, 공간적 정보와 획순, 획수, 필기 방향이나 속도 등의 부가적인 정보로 입력 데이터에서 특징을 추출하고, 미리 정해진 여러 모델 중의 한 모델로 분류하는 과정으로 구성되고 그림 2.1은 문자 인식 과정을 보여 준다[10].



[그림 2.1] 문자 인식 과정

## 2.3 구조적 가상 키보드에 관한 연구

일반적인 PC 키보드를 화면에 표현할 경우 한글 입력을 위한 키는 Shift 키를 눌러야 하는 쌍자음(ㄱ, ㅋ, ㆁ, ...)과 복자음(ㄷ, ㅌ, ...), 이중 모음을 포함해서 33 개이다. 이 33 개의 키를 모두 버튼으로 만들어 화면에 표시한다면 입력 창의 크기가 최소화되어야 할 경우 상대적으로 버튼의 크기가 작아질 수 밖에 없다. 이러한 단점을 해결하기 위해 제안한 소프트웨어 키보드가 구조적 가상 키보드이다[7].

구조적 가상 키보드는 입력 시간의 단축을 위해 초성, 중성, 종성용 키보드를 분리하는 구조형을 지향하고 초성, 중성, 종성의 입력 창이 자동 전환된다. 즉, 초성을 입력하면 중성 입력창이 활성화되고 중성을 입력하고 나면 종성 입력창이 활성화 된다. 따라서 전체 선택 영역을 축소시켜 자소 선택의 간편성과 신속성을 높여 준다[7].

이 기법에서는 초성과 종성용 키보드를 별개의 키보드 입력 창으로 분리시켰고, 쌍자음과 복자음, 이중 모음을 각각 다른 키 버튼으로 할당하였기 때문에 초성 키보드의 버튼 수는 19 개, 중성 키보드는 21 개, 종성 키보드는 21 개이다. 따라서 일반 PC 에서 사용하는 하드웨어 키보드의 한글 입력을 위한 키 버튼 수인 26 개와 비교했을 때는 효율성이 그다지 높지 않으므로 키보드의 버튼 수를 줄일 수 있는 알고리즘이 필요하다.



### III. 데이터베이스에 근거한 지능형 키보드

본 장에서는 데이터베이스에 근거한 지능형 키보드에 대해 정의하고, 기능을 소개하며, 이 방법을 통하여 주어진 단어들의 집합을 최소한의 키 집합으로 입력할 수 있는 방법과 키 선택 횟수를 최소화할 수 있는 방법을 제안한다.

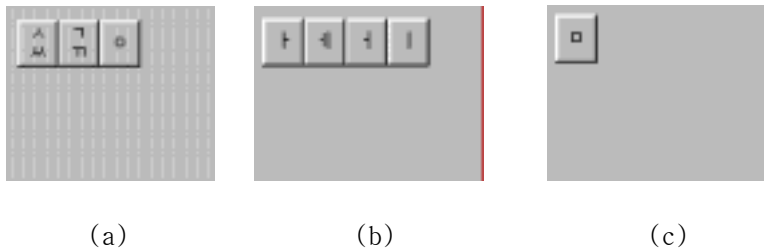
#### 3.1 데이터베이스에 근거한 지능형 키보드 정의

데이터베이스에 근거한 지능형 키보드란 모바일 기기와 같이 하드웨어 키보드나 키패드가 없는 단말기에서 데이터베이스나 파일 형태로 저장되어 있는 단어들의 집합을 입력할 때 필요한 키만을 보여 주는 지능형 소프트웨어 키보드이다. 즉, 입력하고자 하는 단어의 초성에 해당하는 키를 선택하면 전체 단어들의 집합에서 선택된 초성 다음에 나올 수 있는 중성 키만을 보여 주게 된다. 그리고 중성 키를 선택하면 전체 단어들의 집합에서 지금까지 선택된 초성과 중성 다음에 올 수 있는 중성 키만을 보여 준다. 만약 중성이 없다면 다음 음절의 초성 키를 보여 준다.

만약 표 3.1 과 같은 단어 집합이 주어진다면 초기에 보여지는 초성 키 보드는 그림 3.1 (a)와 같이 모든 단어의 첫 음절의 초성에 해당하는 자음을 중복 없이 도시한다. 이 상태에서 ‘ㅅ’ 키 버튼을 선택했을 때는 그림 3.1 의 (b)와 같이 ‘ㅅ’ 을 초성으로 가진 단어의 중성에 해당하는 모음 키들이 도시 된다. 이때 ‘ㅓ’ 키를 선택하면 그림 3.1 의 (c)와 같이 초성이 ‘ㅅ’ 이고 중성이 ‘ㅓ’ 인 음절의 중성에 해당하는 자음 키 버튼이 도시된다.

[표 3.1] 그림 3.1의 입력 단어 집합

삼도물산	ㅅ	ㅊ	ㅍ
경남모직우	ㄱ	ㅋ	ㅇ
인지컨트롤	ㅇ	ㅣ	ㄴ
삼성전자우	ㅅ	ㅊ	ㅍ
삼성증권우	ㅅ	ㅊ	ㅍ
경동보일러	ㄱ	ㅋ	ㅇ
엘지애드	ㅇ	ㅋ	ㄴ
센추리	ㅅ	ㅋ	ㄴ
성지건설	ㅅ	ㅋ	ㅇ
신평제지	ㅅ	ㅣ	ㄴ



[그림 3.1] 표 3.1과 같은 단어 집합이 주어졌을 때 키보드 변화 :

- (a) 초성 선택을 위한 키보드
- (b) (a)상태에서 ‘ㅅ’ 을 선택했을 때 키보드
- (c) (b)상태에서 ‘ㅊ’ 를 선택했을 때 키보드

### 3.2 데이터베이스에 근거한 지능형 키보드의 설계

주어진 단어들의 집합을 효율적으로 입력하기 위한 지능형 키보드의 설계 사양을 다음과 같이 정하였다.

- (1) 한글 입력에 있어서 기존의 하드웨어 키보드를 소프트웨어 키보드

로 대체

- (2) 초성과 종성 입력을 위한 자음 키보드와 중성 입력을 위한 모음 키보드로 분리
- (3) 주어진 단어들의 집합에 해당하는 자음과 모음 키 버튼만을 지능적으로 도시
- (4) 키 버튼 선택 횟수 최소화를 위한 후입력 자동 완성 기능

위와 같은 기능 구현을 위한 자음 키보드와 모음 키보드의 구성과 지능형 키 도시 기능은 다음과 같다.

### 3.2.1 자모의 구성

데이터베이스에 근거한 지능형 키보드에서 초성과 종성 입력을 위해서는 그림 3.2 의 (a)와 같은 자음 키보드를 사용한다. 일반 PC 의 키보드와 개수와 비교할 때 빈칸 키가 추가되는데 이는 종성이 없는 단어와 종성이 있는 단어를 선택할 때 사용된다.

초성에는 이중 자음이 존재하지 않으므로 쌍자음을 입력할 경우는 더블 클릭과 같은 이벤트로 구현하여 단자음과 구분할 수 있다. 종성에는 쌍자음과 이중 자음이 모두 존재할 수 있으므로 쌍자음은 초성과 마찬가지로 처리하고, 이중 자음의 경우에는 하나의 자음 선택 후 다시 종성을 위한 키 버튼들이 도시될 때 연속적으로 선택한다.

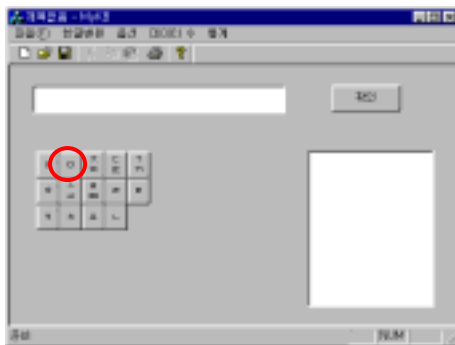
중성 입력을 위해서는 그림 3.2 의 (b)와 같은 키보드를 사용하는데 일반 PC 키보드와 비교해보면 ‘ㅈ’ 와 ‘ㅉ’ 키가 별개의 버튼으로 구분되었다. 이는 키 버튼 선택 횟수 최소화를 위한 방안으로 Shift 키 없이 선택이 가능하도록 하기 위함이다.



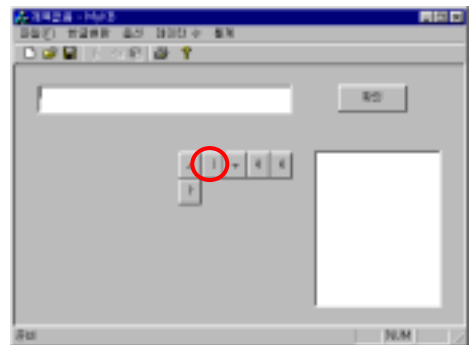
고 단자음일 경우 해당 자음 키 버튼을 도시한다.

- (g) 종성 입력을 위해 자음 키 버튼을 선택했을 때, 단자음일 경우는 다음 음절의 초성에 해당하는 자음 키 버튼을, 이중 자음일 경우는 두번째 종성에 해당하는 자음 키 버튼을 도시한다.

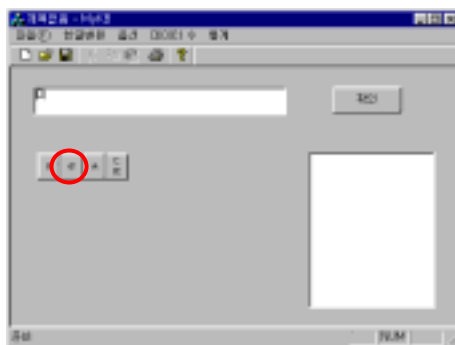
그림 3.3 은 현재 등록된 1000 개의 주식 거래소 이름을 샘플 집합으로 구성하였을 때 ‘미래와사람’ 이라는 주식 명을 입력하는 과정이다. 다섯 음절로 구성되었지만 입력에 필요한 키 개수는 첫 음절의 초성 키를 제외했을 때 12 개로 일반 키보드에 비하여 월등히 적다.



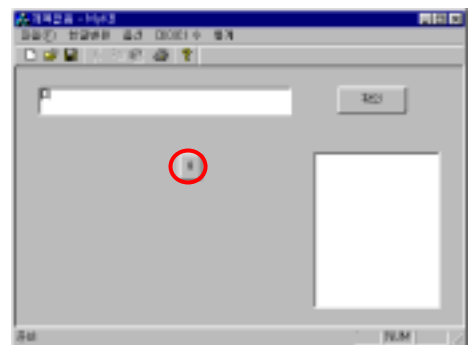
(a) 초성 선택 화면



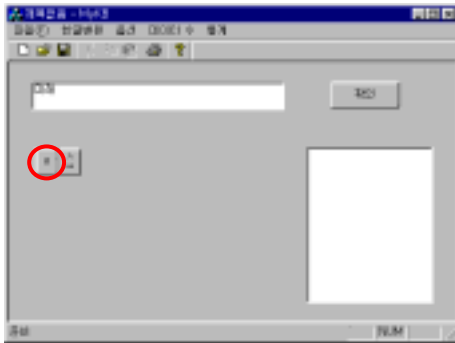
(b) (a)에서 ‘ㄱ’ 선택 후



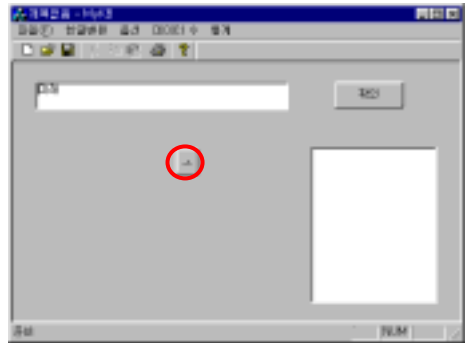
(c) (b)에서 ‘ㅋ’ 선택 후



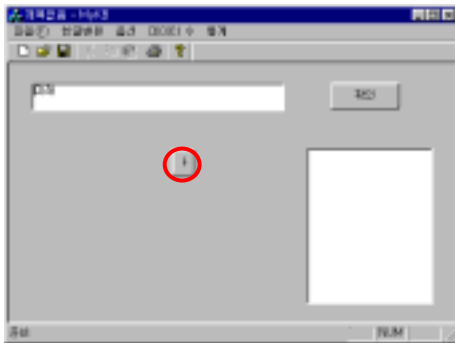
(d) (c)에서 ‘ㆁ’ 선택 후



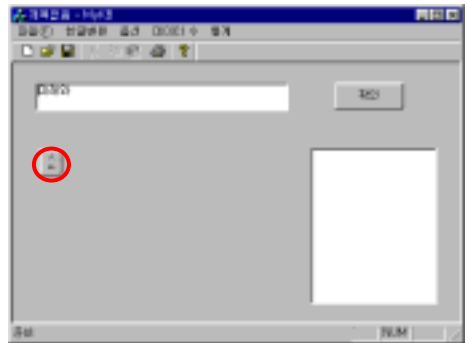
(e) (d)에서 '가' 선택 후



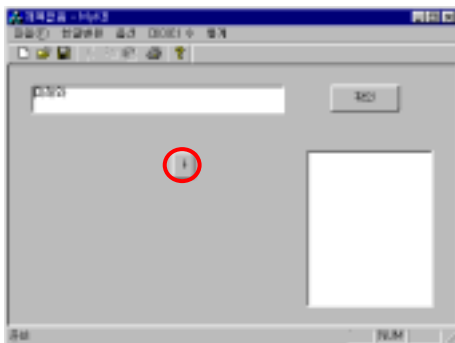
(f) (e)에서 '나' 선택 후



(g) (f)에서 '다' 선택 후



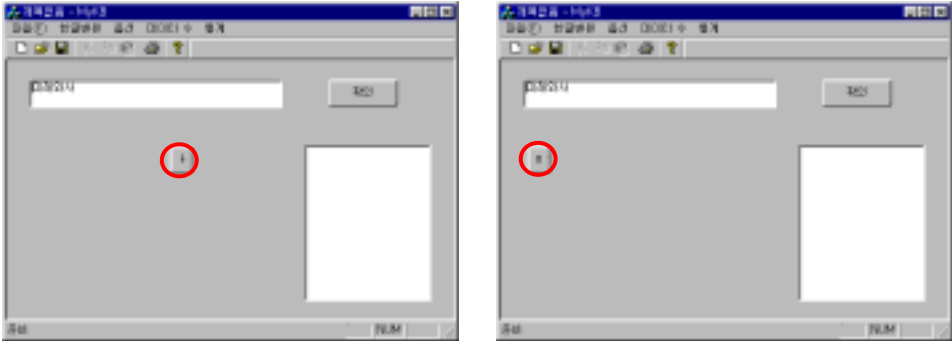
(h) (g)에서 '라' 선택 후



(i) (h)에서 '사' 선택 후



(j) (i)에서 '아' 선택 후



(k) (j)에서 ‘르’ 선택 후

(l) (k)에서 ‘라’ 선택 후

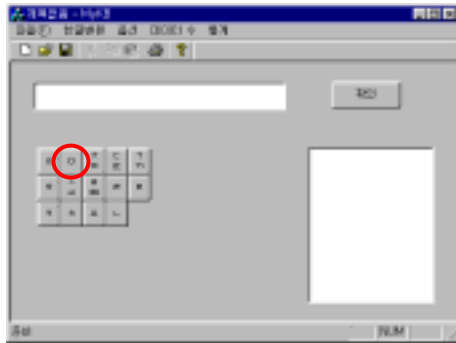
[그림 3.3] 지능형 키보드 입력 과정 예

### 3.2.3 후입력 자동 완성 기능

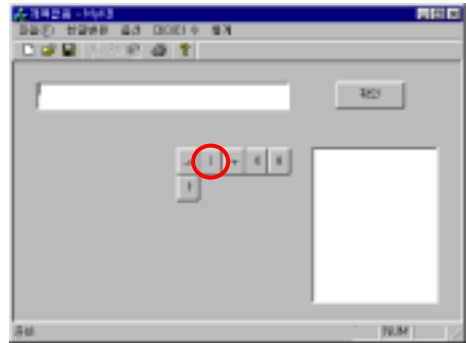
데이터베이스에 근거한 지능형 키보드는 주어진 단어들의 집합을 효율적으로 입력하기 위한 소프트 키보드이므로 초성, 중성, 종성을 선택해 감에 따라 입력 가능한 단어들의 수는 감소하게 된다. 따라서 남아 있는 단어가 한 개일 경우는 자동으로 나머지 음절들을 채움으로써 키 선택 횟수를 줄일 수 있고, 화면에 공간적 제약이 없을 경우는 리스트 박스와 같은 컨트롤을 사용하여 입력 가능한 단어들을 나열하고 그 중 하나의 단어를 선택할 수 있다.

표 3.1 을 예로 들어 설명하면 ‘스’ 과 ‘ㅏ’, ‘ㅓ’ 을 선택했을 때 입력 가능한 단어들은 ‘삼성전자우’ 와 ‘삼성증권우’ 두개로 한정된다. 이 때 한번의 키 선택으로 두 단어 중 하나를 입력할 수 있고, 리스트 박스를 사용하여 입력 가능한 단어들이 한정적이 되었을 때 선택할 수 있다.

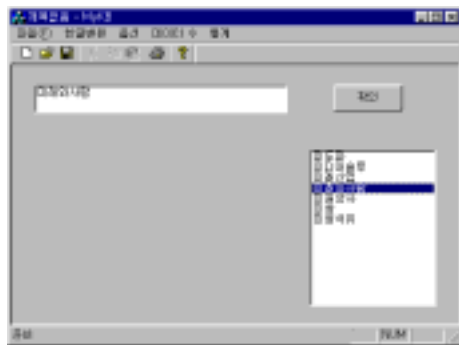
그림 3.4 는 앞 절의 그림 3.3 에서 입력의 예로 제시했던 동일한 샘플 집합의 크기 1000 에서 후입력 자동완성 리스트박스의 크기가 10 일때의 예이다. 동일한 ‘미래와사람’ 이라는 단어를 입력하기 위하여 ‘ㅓ’ 과 ‘ㅓ’ 를 선택하면 리스트박스에 나머지 입력 가능한 단어들이 도시 되고, 여기서 선택 하면 전체 3번의 선택으로 입력을 완성할 수 있다.



(a) 초성 선택 화면



(b) (a)에서 'ㅇ' 선택 후



(c) (b)에서 'ㅣ' 선택 후

[그림 3.4] 후입력 자동 완성 기능 예

### 3.3 시스템 구현 환경

본 논문에서 제안한 데이터베이스에 근거한 지능형 키보드를 구현하기 위한 시스템의 환경은 표 3.2와 같다.

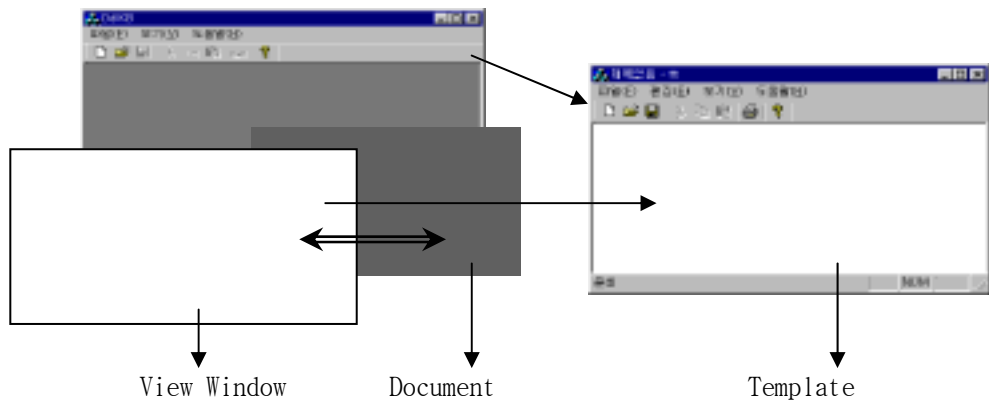
[표 3.2] 시스템 구현 환경

운영 체제	Microsoft사의 Windows 98
개발 도구 및 언어	Microsoft Visual Studio 6.0 Microsoft Visual C++ Compiler



프로그램은 PC(Personal Computer)상에서 Microsoft 사의 Visual C++ 6.0 컴파일러를 사용했고, MFC(Microsoft Foundation Class) 환경에서 프로그래밍하였다.

본 논문에서 구현한 SDI(Single Document Interface)형태의 MFC 프로그램 구조는 그림 3.5 와 같다.



[그림 3.5] SDI 형 MFC 프로그램 구조

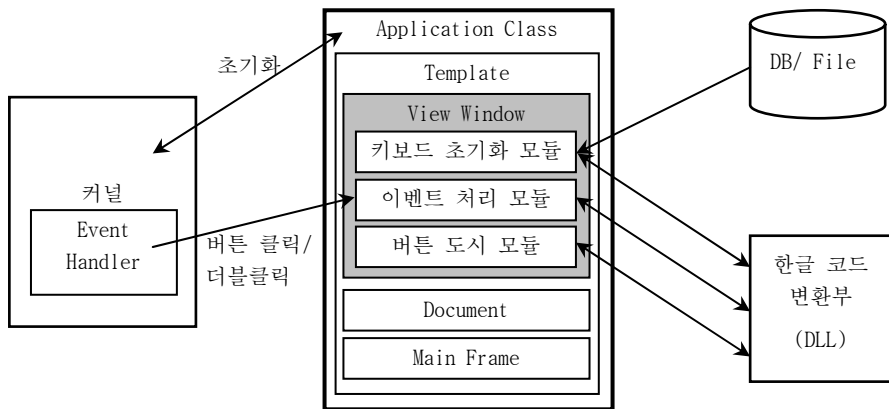
### 3.4 데이터베이스에 근거한 지능형 키보드 알고리즘

본 절에서는 데이터베이스에 근거한 지능형 키보드의 전체 구조를 블록 다이어그램을 통해 살펴보고, 지능형 키보드의 알고리즘을 순서도로 설명한다.

#### 3.4.1 Overall Architecture

데이터베이스에 근거한 지능형 키보드의 전체적인 구조는 그림 3.6 과

같다. 프로그램은 커널과의 초기화로 시작되고 데이터베이스나 파일로부터 주어진 단어 집합을 읽어 들여 키보드 초기화 모듈을 수행한다. 그 후 사용자의 이벤트가 발생하면 이벤트 처리 모듈이 이벤트의 종류와 이벤트가 발생한 버튼에 따라 해당 이벤트 처리 하여 다음 입력에 필요한 키 버튼들을 도시한다. 이 과정에서 한글 완성형 코드와 조합형 코드간의 코드 변환에 관련된 함수들을 사용하는데, 이 함수들로 구성된 DLL(Dynamic Linked Library) 파일은 한국어 정보처리 연구소에서 제공하는 소스를 바탕으로 만들었다.



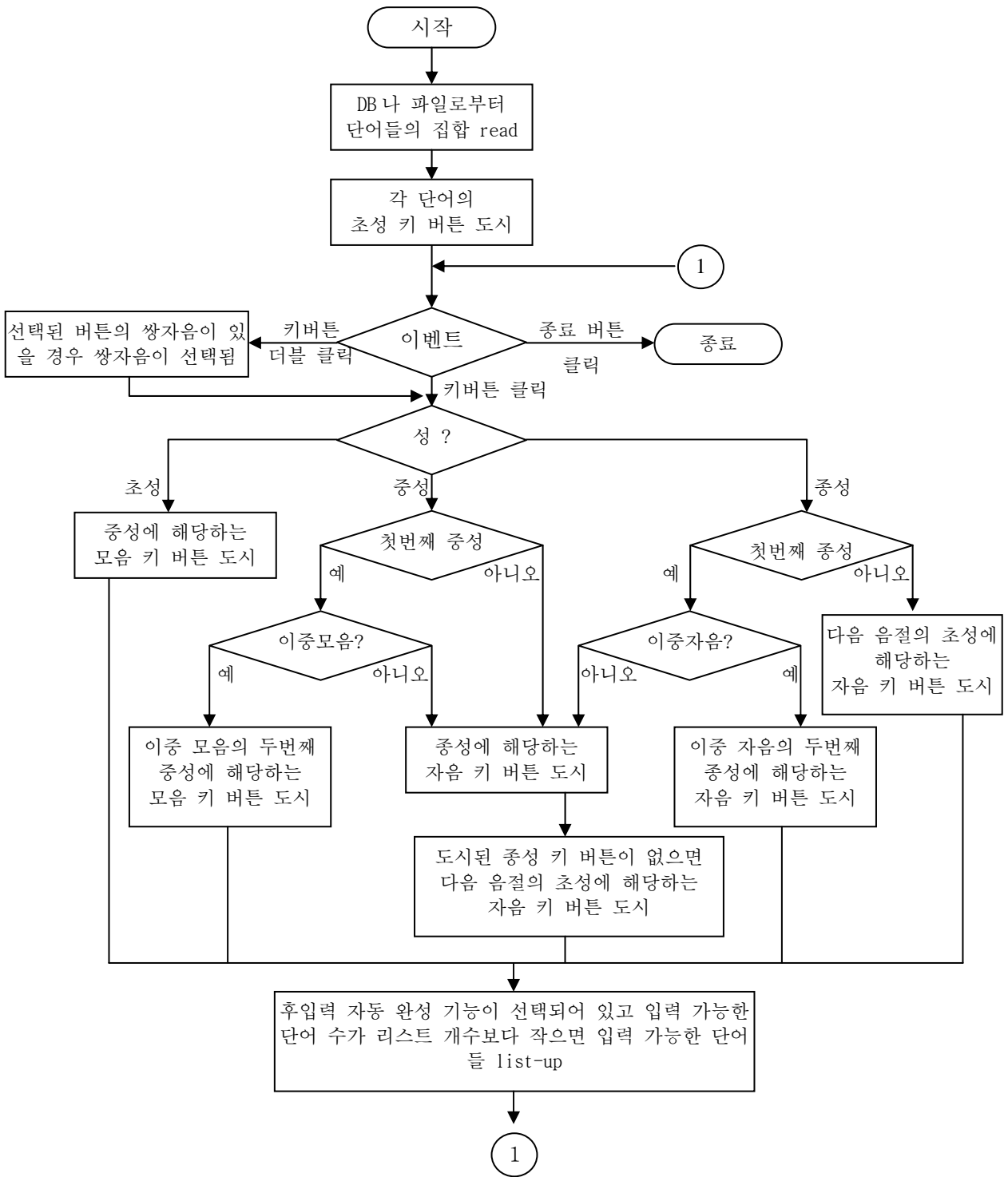
[그림 3.6] 데이터베이스에 근거한 지능형 키보드 블록 다이어그램

### 3.4.2 프로그램 순서도

데이터베이스에 근거한 지능형 키보드의 처리 과정은 데이터베이스나 파일로부터 주어진 단어의 집합을 읽어 조합형 코드로 변환해서 내부 자료 구조에 저장한 후 각 단어의 첫 음절의 초성에 해당하는 자음 키 버튼을 중복 없이 도시한다. 버튼 클릭 이벤트가 발생하면 선택된 버튼을 초성으로 하는 단어의 중성 키 버튼을 도시한다. 중성 키 버튼이 선택되었다면 이중 모음일

경우에는 두 번째 모음 키 버튼을, 단모음일 경우에는 종성에 해당하는 자음 키 버튼을 도시한다. 종성의 경우에도 이중 자음이 올 수 있으므로 첫 번째 종성일 때는 두 번째 종성에 해당하는 자음 키 버튼을, 아닐 경우에는 다음 음절의 초성에 해당하는 자음 키 버튼을 도시한다. 만약 입력 가능한 단어들의 해당 음절에 종성이 하나도 없다면 다음 음절의 초성에 해당하는 자음 키 버튼들을 도시한다. 후입력 자동 완성 기능이 선택되어 있다면 입력 가능한 단어들의 수를 계산하여 정의된 한정적 수보다 작을 경우 리스트 박스에 나열한다.

위 과정을 순서도를 통해 표현하면 그림 3.7 과 같다



[그림 3.7] 데이터베이스에 근거한 지능형 키보드의 순서도

### 3.4.3 입력 파라미터

데이터베이스에 근거한 지능형 키보드의 키 선택 횟수를 최소화하기 위한 기능에 사용한 파라미터들은 표 3.3 과 같다.

[표 3.3] 입력 파라미터

파라미터	값	설명
nCount	1 ~ 2000	주어진 단어 집합의 크기
bCompList	TRUE / FALSE	후입력 자동 완성 리스트 기능의 실행여부
nListCount	1 / 10	후입력 자동 완성 리스트의 크기 1 : 대응되는 단어가 하나일 때 자동완성 10 : 대응되는 단어가 10 개일 때 자동 완성 리스트를 통해 나열

본 논문에서 구현한 데이터베이스에 근거한 지능형 키보드에 데이터베이스나 파일 형태로 주어지는 입력 단어 집합의 개수를 nCount 로 나타내는데 프로그램내에서 배열의 크기를 결정하는 변수여서 최대값을 2000 으로 정의하였다. bCompList 값이 TRUE 이면 후입력 자동 완성 기능이 실행이 되고 이때 대응되는 단어의 개수는 nListCount 에 의해 결정된다. 버튼이 클릭 될때마다 초성, 중성, 종성에 무관하게 대응되는 단어의 개수가 nListCount 보다 작으면 대응되는 단어들을 리스트 박스에 나열한다. bCompList 값이 FALSE 이면 단어의 마지막 음절에 해당하는 중성이나 종성 버튼을 선택했을 때 단어 입력이 완성된다.

## 3.5 데이터베이스에 근거한 지능형 키보드의 특징

본 논문에서 제안하는 주어진 단어 집합을 효율적으로 입력하기 위한 데이터베이스에 근거한 지능형 키보드의 특징은 다음과 같다.

- 기존의 하드웨어로 구현된 키보드나 현재 모바일 단말기에서 사용되고 있는 소프트 키보드는 한글 입력시 초성, 중성, 종성에 무관하게 항상 고정된 키 집합을 제공하는 반면 데이터베이스에 근거한 지능형 키보드는 주어진 단어 집합을 분석하여 초성, 중성, 종성 입력시 필요한 키 버튼만을 도시해 준다. 따라서 입력시 도시 되는 버튼 경우의 수를 크게 줄일 수 있다. 이는 모바일 단말기와 같이 화면이 제한적이고 하드 키보드가 없는 경우 유용하다.
- 주어진 단어 집합을 분석하여 입력할 수 있는 키 버튼만을 도시하므로 사용자의 입력 오류를 원천적으로 방지할 수 있다. 따라서 단어 집합을 사전으로 확장하면 단어의 맞춤법 오류를 입력 단계에서 원천적으로 예방할 수 있다.
- 데이터베이스에 근거한 지능형 키보드는 입력 하고자 하는 단어들의 집합이 주어지므로 소수의 음절만 입력하여도 입력 가능 상태인 단어들의 숫자가 현저히 줄어 든다. 따라서 화면의 제한이 있는 경우에는 입력 가능 상태인 단어가 한 개일 때 나머지 음절을 자동적으로 채워 주고, 화면의 제한이 없는 경우에는 입력 가능 상태인 단어가 충분히 줄었을 때 선택 리스트 박스를 통하여 선택하게 하면 단어 입력을 위한 키 선택 횟수를 최소화할 수 있다.

이와 같은 특징들을 바탕으로 구현된 데이터베이스에 근거한 지능형 키보드 가 이전 방식들과 비교했을 때 어느 정도의 성능 향상을 가져오는지 검

증하기 위하여 데이터베이스에 근거한 지능형 키보드 내에 성능 평가 모듈을 추가하였고, 다음 장에서는 여러 종류의 단어 집합에 대해 기존 입력 방식과의 비교를 통해 본 논문에서 제안한 알고리즘의 성능을 평가해 본다.

## VI. 성능 평가

본 장에서는 논문에서 제안한 데이터베이스에 근거한 지능형 키보드의 성능 평가를 위하여 기존에 존재하는 일반 키보드 방식과 비교하여 다음과 같은 성능 평가를 수행한다. 첫째, 동일한 단어 집합에 대해 일반 키보드 방식과 데이터베이스에 근거한 지능형 키보드 방식의 키 개수를 비교한다. 둘째, 동일한 단어 집합에 대하여 일반 키보드 방식과 데이터베이스에 근거한 지능형 키보드 방식의 키 선택 횟수를 비교한다. 이때 데이터베이스에 근거한 지능형 키보드의 후입력 자동 완성 기능이 있을 때와 없을 때를 구분하여 살펴본다. 이를 통해 다양한 형태의 단어 집합이 주어질 때 어떤 방법이 가장 효율적인 입력 방법인지에 대한 기준을 제안한다.

### 4.1 성능 평가 프로그램 알고리즘

성능 평가를 위한 프로그램은 데이터베이스에 근거한 지능형 키보드 프로그램에 하나의 모듈로 추가하였다. 이는 각각의 단어에 대하여 사용자가 키버튼을 선택하는 과정을 성능 평가 프로그램 내에서 동일한 이벤트로 발생시켜 데이터베이스에 근거한 지능형 키보드의 함수들을 공유하지 위해서 이다.

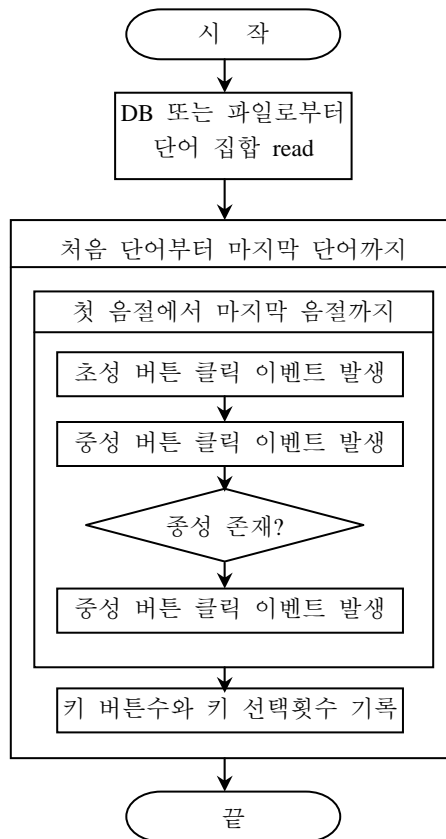
#### 4.2.1 프로그램 순서도

성능 평가 프로그램은 일단 주어진 단어들의 집합을 읽어 들여 분석하는 과정으로 시작한다. 처음 단어에서 마지막 단어까지 각각의 단어에 대하여 각 음절의 초성 버튼 클릭 이벤트를 발생시키고, 중성 버튼 클릭 이벤트를 받



생시킨다. 중성이 있으면 중성 버튼 클릭 이벤트를 발생시키고 중성이 없으면 다음 음절의 초성 버튼 클릭 이벤트를 발생시킨다. 한 단어에 대한 처리가 끝나면 한 단어 입력 과정에서 도시 되었던 키 버튼의 개수와 키 선택 횟수를 특정 파일에 기록한다. 후입력 자동 완성 기능이 실행 중일 경우는 자동 완성 조건을 만족했을 때 버튼 클릭 이벤트 발생을 중지하고 파일에 결과를 기록한 후 다음 단어에 대한 처리를 계속한다. 이 과정을 순서도로 나타내면 그림 4.1과 같다.

[그림 4.1] 성능 평가 프로그램 순서도



#### 4.2.2 입력 파라미터

데이터베이스에 근거한 지능형 키보드의 성능 평가를 위해 사용한 파라미터들은 표 4.1 과 같다. 표 3.3 의 데이터베이스에 근거한 지능형 키보드 프로그램의 파라미터와 비교했을 때 추가되는 단어 집합의 크기인 nCount 값은 단어 집합 크기의 비교 평가를 위해 100/500/1000/1500/2000 중에 하나를 가지도록 한다. 그리고 sFileName 은 각 단어를 입력하는 데 필요한 키 버튼의 수와 키 버튼 선택 횟수를 기록할 출력 파일의 이름이다.

[표 4.1] 성능 평가 프로그램 입력 파라미터

파라미터	값	설명
nCount	100/500/1000/1500/2000	주어진 단어 집합의 크기
bCompList	TRUE / FALSE	후입력 자동 완성 리스트 기능의 실행여부
nListCount	1 / 10	자동 완성 리스트의 크기 1 : 대응되는 단어가 하나일 때 후입력 자동 완성 10 : 대응되는 단어가 10 개일 때 자동 완성 리스트를 통해 나열
sFileName	문자열	출력 파일의 이름

#### 4.3 샘플 단어 집합

본 장의 성능 평가에 사용된 단어 집합은 최대 2000 개이고, 1000 개는 현 주식 시장의 거래소와 코스닥에 상장된 주식 이름에서, 나머지 1000 개는 야후 사전(<http://kr.alldic.yahoo.com>)에 등록된 단어에서 발췌하였다.

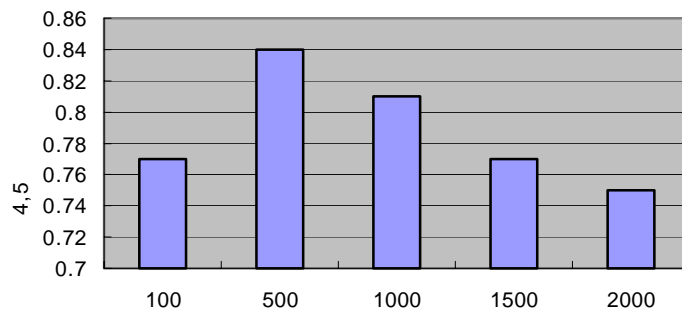
성능 평가에 사용된 단어 집합의 크기는 100 개, 500 개, 1000 개, 1500 개, 2000 개이고 단어의 최소 음절은 2 음절, 최대 음절은 6 음절이다. 각 단

어 집합 크기별 단어의 음절 수는 표 4.2와 같다.

[표 4.2] 단어 집합 크기별 음절 수 비교

	단어 집합의 크기				
	100	500	1000	1500	2000
2 음절	10	36	87	135	170
3 음절	13	40	97	206	310
4 음절	41	259	467	665	864
5 음절	36	163	347	492	653
6 음절	0	2	2	2	3

표 4.2를 살펴 보면 단어 집합의 크기가 500 일 때와 1000 일 때 단어의 길이가 4,5 음절인 단어의 비율이 상대적으로 높다. 이는 앞으로 수행할 성능 평가 과정에서 나타나는 그래프에서 단어 집합의 크기가 500 이나 1000 일 때 그래프 변화가 확연히 나타남의 요인이 된다. 그림 4.2는 각 단어 집합의 크기에서 단어의 길이가 4 음절이거나 5 음절인 단어들이 차지하는 비율을 그래프로 도시한 것이다.

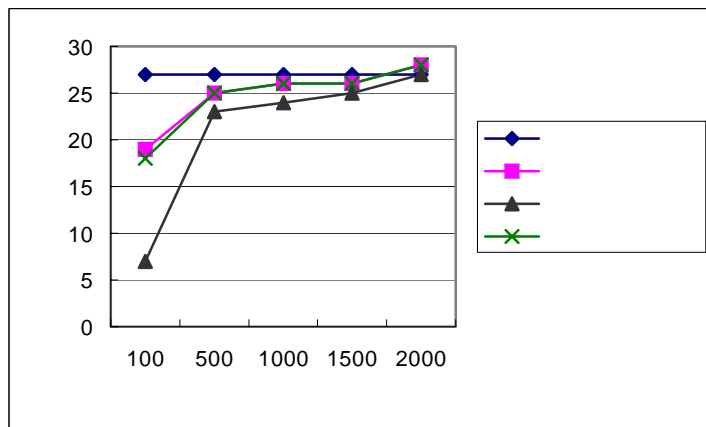


[그림 4.2] 단어 집합의 크기에 따른 4,5 음절 단어 비율

## 4.4 성능 평가 결과

### 4.3.1 일반 키보드 방식과 데이터베이스에 근거한 지능형 키보드 방식의 키 개수 비교

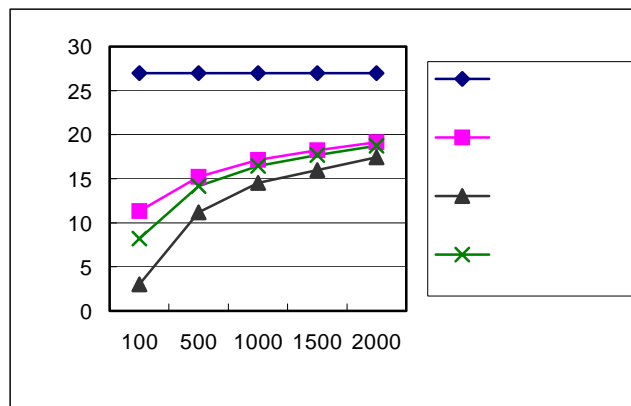
PC 에서 사용하는 일반 키보드 방식과 본 논문에서 제안하는 데이터베이스에 근거한 지능형 키보드 방식의 키 개수면에서의 성능을 평가하기 위하여 각 단어 집합에 대해 입력에 필요한 키 버튼 수의 최대값과 평균값으로 비교하였다. 그림 4.3 은 단어 집합의 크기를 100 개, 500 개, 1000 개, 1500 개, 2000 개의 다섯 가지 경우로 변경하면서 각 키보드 방식별 키 버튼 수의 최대값을 구한 결과이다. 이 때 후입력 자동 완성 리스트는 데이터베이스에 근거한 지능형 키보드에 후입력 자동 완성 리스트 박스를 추가한 기능이고 리스트 박스의 크기는 10 이다. 후입력 자동 완성은 데이터베이스에 근거한 지능형 키보드에서 대응되는 단어가 한 개일 때 자동 완성시켜 주는 경우이다.



[그림 4.3] 단어 집합 크기 변화에 따른 키 버튼 개수 최대값 비교

그림 4.3 은 단어 입력을 위해 필요한 키 버튼 수의 최대값이 후입력 자동 완성 리스트 박스 기능을 가진 데이터베이스에 근거한 지능형 키보드 방식이 가장 작게 나타나는 결과를 보여 준다. 또한 일반 키보드 방식은 단어 집합의 크기에 무관하게 항상 27 개의 키 버튼이 필요하지만 데이터베이스에 근거한 지능형 키보드 방식에서는 키 버튼의 최대값으로 비교해 봤을 때 500 개 이하의 단어 집합에서 지능형 키보드의 모든 방식이 효율적임을 나타내고 있다.

그림 4.4 는 단어 집합 크기 변화에 따른 각 키보드 방식별 키 버튼 수의 평균값을 구한 결과이다.



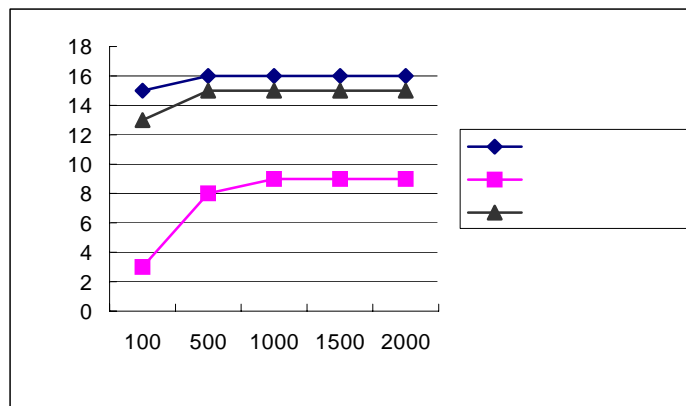
[그림 4.4] 단어 집합 크기 변화에 따른 키 버튼 개수 평균값 비교

위의 그림 4.4 는 단어 입력을 위해 필요한 키 버튼 수의 평균값이 단어 집합의 모든 크기에 대해서 일반 키보드보다 전반적으로 작게 나타나는 결과를 보여 준다. 일반 키보드 방식은 단어 집합의 크기에 무관하게 항상 27 개의 키 버튼이 필요하지만 데이터베이스에 근거한 지능형 키보드의 경우는 2000 개의 단어 집합에 대해서도 20 개 미만의 키 버튼만이 필요함을 나타낸다.

또한 지능형 키보드의 모든 방식이 단어 집합의 크기가 증가하면서 키 버튼 수의 증가 비율이 점차 줄어드는 결과를 보인다.

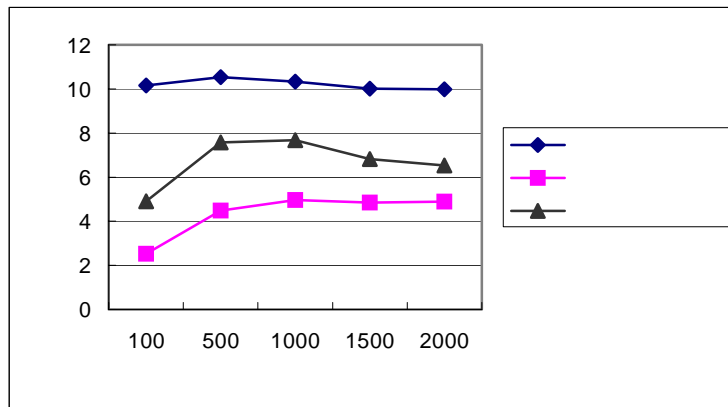
#### 4.3.2 일반 키보드 방식과 데이터베이스에 근거한 지능형 키보드 방식의 키 선택 횟수 비교

이 절에서는 PC 에서 사용하는 일반 키보드 방식과 데이터베이스에 근거한 지능형 키보드 방식의 키 버튼 선택 횟수의 성능 평가를 위하여 각 단어의 입력에 필요한 키 선택 횟수의 최대값과 평균값으로 비교하였다. 데이터베이스에 근거한 지능형 키보드 방식 중 후입력 자동 완성 기능이 없는 경우는 일반 키보드와 키 선택 횟수에 있어서 동일하므로 리스트 박스의 크기가 10인 후입력 자동 완성 리스트 박스 방식과 후입력 자동 완성 방식으로만 성능 평가를 하였다. 그림 4.5 은 단어 집합의 크기를 변경하면서 단어 입력 시 필요한 키 선택 횟수의 최대값을 구한 결과이다.



[그림 4.5] 단어 집합 크기 변화에 따른 키 선택 횟수 최대값 비교

그림 4.5 는 단어 입력을 위해 필요한 키 선택 횟수의 최대값이 후입력 자동 완성 리스트 기능을 가진 데이터베이스에 근거한 지능형 키보드의 경우가 가장 작게 나타나는 결과를 보여 준다. 그리고 세 가지 입력 방식 모두 500 개 이상의 단어 집합에 대해서는 키 선택 횟수의 최대값이 수렴됨을 나타내고 있다. 일반 키보드 방식에서 최대 키 선택 횟수가 16 번으로 나타나는 경우는 단어의 길이가 6 음절로 본 논문에서 성능 평가 시에 사용한 단어들 중에서 음절 수가 가장 많은 경우이다. 후입력 자동 완성 리스트 방식이나 후입력 자동 완성 방식에서 선택 횟수가 최대값으로 나오는 경우는 마지막 한 음절이 다른 단어가 존재할 때이다. 예를 들어 ‘가나다라’ 와 ‘가나다라마’ 와 같이 마지막 한 음절만 다른 단어가 존재한다면 이 경우가 키 선택 횟수의 최대값을 가지는 경우이다.



[그림 4.6] 단어 집합 크기 변화에 따른 키 선택 횟수 평균값 비교

그림 4.6 은 단어 입력 시 필요한 키 선택 횟수의 평균값으로 지능형 키보드 입력 방식의 성능을 평가해 본 결과이다. 키 선택 횟수의 평균값으로 비교했을 때 후입력 자동 완성 리스트 기능이 키 선택 횟수가 가장 적는데 일반 키보드 입력 방식에 비해 50% 정도 선택 횟수가 감소하였다. 후입력 자동

완성 방식은 일반 키보드 방식에 비해 25% 정도 키 선택 횟수가 감소됨을 알 수 있다. 단어 집합의 크기가 500 과 1000 일 때 세 가지 방식 모두 키 선택 횟수가 가장 크게 나타나는데 이는 그림 4.2 에서 나타나듯이 성능 평가에 사용된 단어 집합에서 단어 집합의 크기가 500 이거나 1000 일 때 단어의 길이가 4 음절이거나 5 음절인 단어가 차지하는 비율이 상대적으로 높기 때문이다.

#### 4.3.3 데이터베이스에 근거한 지능형 키보드에서 이중 모음과 이중 자음 분리 시 성능 평가

본 논문에서 제안하는 데이터베이스에 근거한 지능형 키보드는 단어 입력 시 주어진 단어 집합을 분석하여 입력 가능한 키 버튼만을 도시해 주기 때문에 일반 PC 키보드와 다르게 쌍자음이나 이중 자음, 이중 모음을 별개의 키 버튼으로 분리하여 구현할 수도 있다. 즉, 한글의 초성과 종성에 올 수 있는 30 개의 자음과 중성에 올 수 있는 21 개의 모음을 각각 별개의 키 버튼으로 구현할 수 있다. 그림 4.7 은 이중 자음과 이중 모음을 분리했을 경우 사용되는 자음과 모음 키보드이다.



(a) 자음 키보드

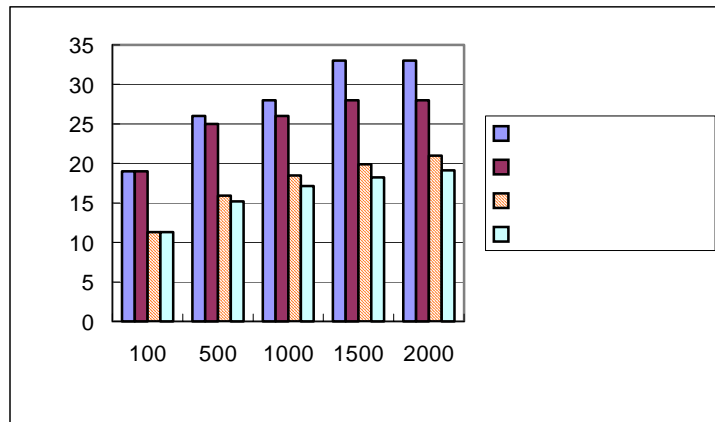


(b) 모음 키보드

[그림 4.7] 이중 자음과 이중 모음 분리 시 키보드 형태



실지로 이중 자음과 이중 모음을 별개의 키 버튼으로 분리했을 경우의 성능 평가를 위해 그림 4.7 과 같은 형태의 키보드로 데이터베이스에 근거한 지능형 키보드를 구현해 보았다. 그림 4.8 은 이중 모음과 이중 자음을 분리했을 경우와 분리하지 않았을 경우 단어 입력에 필요한 키 버튼의 최대값과 평균값을 비교한 그래프이다. 이 비교 그래프를 통하여 1000 개 이하의 단어 집합에서는 이중 모음과 이중 자음을 분리했을 경우와 분리하지 않았을 경우 키 개수의 최대값과 평균값 모두 별다른 차이가 없다는 점과 1000 개보다 큰 단어 집합에서는 평균값보다 최대값에서 두 방법의 성능 차이가 크다는 점을 알 수 있다.



[그림 4.8] 이중 모음과 이중 자음 분리 시 키 개수 비교

본 논문에서 수행한 세 가지 성능 평가의 수행 결과, 데이터베이스에 근거한 지능형 키보드 방식이 입력 시 필요한 키 개수면에서나 키 선택 횟수면에서 일반 PC 키보드 입력 방식보다 효율적임을 나타내었다. 키 개수면에서는 데이터베이스에 근거한 지능형 키보드의 모든 방식이 일반 키보드 방식과 비교했을 때 높은 성능을 나타내었고, 키 선택 횟수면에서는 데이터베이스

에 근거한 지능형 키보드 방식 중 후입력 자동 완성 리스트 박스 방식이 가장 높은 성능을 나타내었다. 또한, 단어 집합의 크기가 1000 개 이하일 경우 데이터베이스에 근거한 지능형 키보드의 자음과 모음 키보드를 이중 자음과 이중 모음을 별개의 키 버튼으로 구현하는 방법과 단자음과 단모음 키 버튼을 통하여 입력하는 방법에 별다른 차이가 없다는 것을 알 수 있었다.

이러한 결과를 바탕으로 다양한 단어의 집합의 크기와 입력 창 크기의 제한 여부에 따라 지능형인 소프트웨어 키보드 입력 방식을 선택적으로 수행할 수 있다. 입력 창 크기에 제한이 클 경우는 대응되는 단어가 하나일 때만 자동 선택을 실행하는 음절 완성 기능이 적합하고 그렇지 않을 경우는 음절 완성 리스트 박스의 크기를 조절해 가면서 효율적인 입력 방식을 선택할 수 있다.

## V. 결 론

### 5.1 연구 결과 및 의의

최근 무선 인터넷과 정보 통신 기술의 발달로 정보 기기와 모바일 단말기의 크기가 점차 소형화 되어감에 따라 부피가 크고 이동성에 문제가 있는 하드웨어 키보드를 그대로 사용할 수 없는 문제점이 있다. 본 논문에서는 모바일 기기와 같이 하드웨어 키보드가 없는 단말기에서 기존의 하드웨어 키보드 대신에 사용할 수 있는 소프트웨어 키보드 형태에 관해 살펴 보았고, 기존의 방법들이 단어 입력 창의 크기면에서나 입력 속도면에서 많은 단점을 가지고 있음을 지적하였다. 일반 하드웨어 키보드를 그대로 화면에 보여 주는 화면 키보드 형태는 입력 창의 크기 문제와 입력 속도에 문제가 있고, 펜을 통해 입력하는 방식은 정자체가 아닌 필기체의 경우에 아직까지도 오류율이 높고 문자 인식 속도도 만족스럽지 못한 상태이다.

본 논문에서는 소프트웨어 키보드를 구현할 때 주어진 단어 집합을 최소한의 키 버튼과 최소한의 키 선택 횟수를 통해 입력할 수 있는 데이터베이스에 근거한 지능형 키보드를 제안하였다. 데이터베이스에 근거한 지능형 키보드에서 키 버튼 개수를 줄이는 방법으로 초성과 중성 입력을 위한 자음 키보드와 중성 입력을 위한 모음 키보드를 가지는 구조적 키보드 형태를 제안하였는데 자음과 모음 키보드는 쌍자음과 이중 자음, 이중 모음을 위한 별개의 키 버튼이 존재하는 것이 아니라 단자음이나 단모음 버튼을 통해 입력하므로 키 버튼이 최대한 사용되더라도 일반 하드웨어 키보드의 키 버튼 수를 넘지 않는다. 또한, 초성을 선택하면 입력한 초성 다음에 나올 수 있는 중성을 선택하기 위한 모음 키보드로, 중성을 입력하면 중성이나 다음 음절의 초성을 선택할 수 있는 자음 키보드로 자동 전환된다. 데이터베이스에 근거한 지능형

키보드에서 키 버튼 선택 횟수를 최소화하는 방법으로 후입력 자동 완성 리스트 박스 기능을 제안하였는데 단어의 초성, 중성, 종성 버튼 선택시마다 입력 가능한 단어들의 개수를 세어서 단어의 개수가 후입력 자동 완성 리스트 박스의 크기 이하가 될 때 리스트 박스에 단어들을 나열해 주고 한번의 클릭을 통하여 나머지 입력을 완성하는 기능이다. 후입력 자동 완성 리스트 박스의 크기는 입력 창의 크기 제한에 따라 조절하여 사용할 수 있다.

제안된 방법들과 일반 키보드 방식의 성능을 비교, 평가하기 위하여 2000 개의 샘플 단어 집합을 구성하고, 단어 집합의 크기를 100 개, 500 개, 1000 개, 1500 개, 2000 개로 변경해 가면서 세 가지 성능 평가를 수행하였다. 성능 평가 결과 첫째, 단어 입력시 필요한 키 버튼 개수면에서 데이터베이스에 근거한 지능형 키보드 방식이 모든 단어 집합 크기에서 평균적으로 높은 성능을 보였다. 또한 단어 집합의 크기가 증가할수록 키 버튼 수의 증가율은 감소함을 알 수 있었다. 둘째, 단어 입력시 필요한 키 선택 횟수면에서 데이터베이스에 근거한 지능형 키보드의 후입력 자동 완성 리스트 박스 방식이 키 선택 횟수의 최대값이나 평균값에서 모두 효율적으로 나타났다. 후입력 자동 완성 리스트 박스의 크기가 1 일 때와 10 일 때를 비교해 보면 크기가 10 일 때가 평균적으로 35%정도 더 선택 횟수를 줄일 수 있는 것으로 나타났다. 이러한 성능 평가 결과를 통하여 단어 집합의 크기와 입력 창의 크기 제한 여부에 따라 데이터베이스에 근거한 지능형 키보드 방식을 선택적으로 수행할 수 있다.

주어진 단어 집합을 입력할 때 지능적으로 키 버튼을 도사하는 데이터베이스에 근거한 지능형 키보드 방식에 대한 연구를 통하여 입력에 필요한 키 버튼 선택 영역의 최소화로 입력 창의 크기를 줄일 수 있고, 최소의 키 선택으로 입력의 신속성을 높여 전체 입력 시스템의 성능을 향상시킬 수 있다.

## 5.2 향후 연구 과제

본 논문에서는 제안한 방법의 성능 평가 시 단어 집합의 크기는 최대 2000 개였다. 단어 집합을 사전으로 확대하면 모든 단어에 대한 입력의 오류를 방지할 수 있는 지능형 키보드가 가능할 것이다. 따라서 단어 집합을 사전으로 확대했을 때 프로그램의 성능 향상을 위한 알고리즘과 탐색 기법에 대한 연구가 필요하다.

또한 데이터베이스 기반 지능형 키보드 방식은 키 버튼의 종류와 위치가 매번 변화되므로 입력 시 혼란을 줄 수 있다. 입력시마다 도시 되는 키 버튼이 가변적일 때 키 버튼 인지와 선택에 좀더 효율적일 수 있도록 키 버튼의 배열에 대한 고려가 필요하다.

## 참 고 문 헌

- [1] 박동순, 이재현, 최은, 강석, “컴퓨터 한글 코드의 사용과 표준화”, 정보 과학회지 제 6 권 제 1 호(' 88.2), pages 69-74, 1988
- [2] 김지현, “휴대용 키보드”, <http://www.i-biznet.com> [김지현의 디지털 월드]
- [3] 고희대, “컴퓨터에 의한 한글 입출력 처리에 관한 연구”, 전남대 대학원 석사 학위 논문, pages 8-9, 1984
- [4] 양명섭, “구조적 형태분석에 의한 한글 자모분리기의 설계 및 구현”, 전북대 대학원, pages 1-9, 1995
- [5] 이 준, “고정어 풀어쓰기로 표현된 한글 처리에 관한 연구”, 공군사관 학교 논문집 41(' 98.2), pages 129-149, 1998
- [6] 한국어 정보처리 연구소, “C 로 구현한 한글 코드 시스템 프로그래밍 가이드”, 한국어 정보처리 연구소, 2001
- [7] 이재혁, “한글입력을 위한 구조형 가상 키보드의 개발”, 군산대학논문집 9(' 84.12), pages 281-296, 1984
- [8] 구민모, 이만영, “전화기 자판의 한글 입력 효율성 평가모형”, 정보처리학회 논문지 D 제 8-D 권 제 3 호(' 02.6), pages 295-304, 2001

- [9] 이현주, 조문증, “Pen Windows 상에서의 On-line 필기인식 기술 현황” ,  
정보과학회지제 11 권 제 5 호(’ 93.10), pages 104-114, 1993
- [10] 안 창, “한글 인식 기술의 현황과 문제점” , 한일장신대신학과사회 14,  
pages 419-430, 2000
- [11] 이상영, Visual C++ Programming Bible ver. 6.x, 영진출판사, 2002

## ABSTRACT

### Research On the Intelligent Software Keyboard based on Database

*Department of Electronic Commerce  
Ewha Graduate School of Information Science  
Lee Kye Suk*

These days, due to the development of wireless internet and information & communication technologies, the size of mobile terminal and information device is getting smaller. Especially, since the popularization of PDA(Personal Digital Assistant) and cellular phone, the number of data management using small-sized terminal is on the increasing trend. However, such mobile terminals, mostly carried for portability, are not possible to go with the existing keyboard connected to desktop PC and although it has built-in character input system, it has an inefficiency of input due to the limitation of button size and the number of buttons.

This thesis has suggested new software-driven keyboard input method which is efficient in the frequency of key selection and the size of area on which to select the key button in inputting the items stored in mobile devices in the form of database incompatible with generic keyboard for PC.



When given the set of words to input, by showing only the minimum set of buttons needed for inputting given words, it is possible to reduce the selection area of key button and to minimize the number of key button selection by providing the automatic command completion functionality when the number of the words which may be input is limited after the key buttons are selected for the promptness in inputting words.

After evaluating the performance on the frequency of key button selection and key button number in the set of words in various sizes by the comparison of the method of intelligent input method suggested herein with existing one, it proved that the frequency of key button selection and the number of key button has decreased in case of intelligent input method suggested.

Through the result of this research, we may expect the improvement of performance of input system in mobile information terminal on which to input the set of stored words in the form of database as a software-driven keyboard form.