



저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



동일조건변경허락. 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

碩士學位 論文

휴대용 터치스크린 장치를 위한
한글 자소 배치

A model of Korean grapheme for
portable touch screen device

2010 年 12月

崇實大學校 大學院

컴퓨터學科

鄭 賢 基

碩士學位 論文

휴대용 터치스크린 장치를 위한
한글 자소 배치

A model of Korean grapheme
for portable touch screen device

2010 年 12月

崇實大學校 大學院

컴퓨터學科

鄭 賢 基

碩士學位 論文

휴대용 터치스크린 장치를 위한
한글 자소 배치

指導教授

이 論文을 碩士學位 論文으로 제출함

2010 年 12月

崇實大學校 大學院

컴퓨터學科

鄭 賢 基

鄭賢基의 碩士學位論文을 認准함

審 查 委 員 長 印

審 查 委 員 印

審 查 委 員 印

2010 年 12月

崇實大學校 大學院

感謝의 글

가장 먼저 지금까지 저를 지도해주시고 이끌어 주시고 가르쳐 주신 양승민 교수님께 큰 감사를 드립니다. 방황하던 저를 기다려주시고 논문을 완성하기까지 품어주셔서 이렇게 제가 논문을 마칠 수 있게 되었습니다. 정말 교수님께는 감사하다는 말도 너무나 부족하여 죄송한 마음을 항상 품고 있습니다.

대학원에서 목표를 갖게 해주고 진심어린 조언을 해주셨던 현희형에게도 감사를 드립니다. 여러 조언을 해주셨는데 기대에 미치지 못해서 죄송했었습니다.

항상 제게 생각할 거리를 만들어 주고 새로운 관점을 생각할 수 있게 해 준 용수형, 묵묵히 격려해 준 유진이 형, 항상 유쾌함으로 분위기를 밝게 해 준 명오 형, 친구이기 전에 선배로서 관심을 갖고 많은 부분을 알려주고 일깨워준 도형이, 부족한 신입 총무에게 많은 걸 알려주고 잘 대해준 미영이 누나, 뭐든 물어보면 꼼꼼하게 답변해준 진오형, 재밌는 소재화 토론거리를 항상 제공해주고 이 논문의 기초를 닦아놓은 종화형, 항상 먼저 다가와서 처진 저를 멀쩡하게 바꿔준 정균 선배에게 고마운 마음을 전합니다.

제 유일한 동기이자 인생 선배이자 든든한 형으로써 항상 철없는 동생을 일깨워 주시고 챙겨주시느라 고생 많으셨던 성국이 형에게도 큰 감사를 드립니다. 형 덕분에 연구실 생활 잘 견뎌나갈 수 있었습니다.

짧은 기간 밖에 함께 하지 못해서 더욱 미안하고 애똥한 춘기와 종하, 뒤늦게 연구실에 들어와서 잘해주지도 많은 것을 전해주지도 못하고 제가 도움만 받은 든든하고 잘 따라주고 활기가 넘쳤던 은혁이, 영석이, 창훈이에게도 고마움을 전합니다.

오랫동안 함께 하지 못했지만 제게 정과 지식을 함께 느끼게 해주었던 정보대 학술 소모임 오즈의 선배들에게도 감사 인사를 드립니다. 배우기만 하고 제대로 전하질 못해서 활동하지 않은 지금까지도 항상 미안한 마음이 듭니다.

어려버리한 사회 초년생을 잘 다독이시고 이끌어주신 엠스톤의 모든 동료분들에게도 감사함을 전합니다. 특히 저 때문에 고생하신 소장님께 큰 감사를 드립니다.

학부 다니는 동안 재미없는 친구를 잘 챙기고 놀아주고 함께 해 준 옛취 모임 친구들에게도 고마움을 전합니다.

초등학교 때부터 지금까지 불 때마다 정을 잊지않고 격려해준 사거리 및 장성 고등학교 친구들에게도 감사함을 전합니다.

입학했을 때부터 지금까지 활동이 뜸해진 지금까지도 잊지않고 불러주고 함께 살기도 하고 고민도 들어주고 힘든 일도 함께 해주고 같이 웃고 울고 놀고 얘기했던 제 대학 인생의 대부분을 차지한 전통 무예 동아리 송실경당 빛살가름과 관련한 모든 이들에게도 감사한 마음을 드립니다. 다 언급하고 싶지만 너무 많아서 마음만 전달합니다.

부족한 아들, 손자가 이런 저런 시행착오를 겪으면서 집에 폐만 끼치고 있음에도 격려해주시고 지원해주시고 꾸짖어주신 그리고 지금까지 고생하시며 길러주신 부모님과 할머니가 계셔서 제가 졸업 논문을 마칠 수 있는 날이 온 것 같습니다. 감사하고 사랑합니다.

해주는 것도 별로 없는 오빠를 위해 항상 전화해주고 챙겨주고 친구가 되어주었던 동생에게도 고마운 마음을 전합니다.

항상 뒤에서 양으로 음으로 챙겨주시고 묵묵히 기다려주시고 앞에서는 또 응원해주신 우리 첫째, 둘째, 막내 작은 아버지, 어머니와 사랑스런 사촌들 그리고 그 외 친척분들에게도 감사하고 사랑하는 마음을 전합니다.

끝으로 지금 당장 떠오르지 않은 모든 이들과 함께 그동안 저와 함께 해주었던 팔일당 친구들에게 고마움을 전하며 이 모든 기쁨과 영광을 이 모든 일과 사람들을 준비해주시고 예비해주시고 지금도 저와 함께해 주시는 하나님께 돌립니다.

목 차

國文抄錄	v
英文抄錄	vi
제 1 장 서론	1
1.1. 연구목적	1
1.2. 기존의 방법들	1
1.3. 연구방법	3
제 2 장 휴대 전화 및 PDA 한글 입력방식	4
2.1. 버튼 누름 방식에서의 배열	4
2.1.1. A형 방식	4
2.1.2. B형 방식	4
2.1.3. C형 방식	5
2.1.4. 쿼티(qwerty) 방식	6
2.2. 터치 패드(touch pad) 방식에서의 배열	7
2.2.1. 쿼티(qwerty) 방식	7
2.2.2. cube 방식	7
제 3 장 cube 방식 개선 방안	9
3.1. 평균 타수 문제	9
3.2. 3×3 배열	10
3.2.1. cube-3A 방식	10
3.2.2. cube-3B 방식	11
3.2.3. cube-3C 방식	13
3.3. 4×3 배열	14
3.3.1. cube-4A 방식	14
3.3.2. cube-4B 방식	16
제 4 장 구현	18
4.1. 개발 환경	18
4.2. 프로그램 구성	18
4.3. 한글 자소 조합 알고리즘(Hangul Automata)	19
4.4. 실행 결과	21
제 5 장 평가	23
5.1. 자소당 키 입력 타수 비교	24

5.2. Hick's Law	25
5.3. KLM-GOMS Model	26
5.4. 분석결과	28
제 6 장 결론	30
 참고문헌	32

그 립 목 차

[그림 2-1] 두벌식 한글 자판 배치	7
[그림 4-1] 프로그램 구성	19
[그림 4-2] 한글 상태 전이도	20
[그림 4-3] 시뮬레이터에 실행된 화면	21
[그림 4-4] 안드로이드 기기에서 구동시킨 화면	22

표 목 차

[표 1-1] 분산형 한글 배치[7,8]	2
[표 1-2] 획조합 방식[10]	3
[표 2-1] A형 방식	4
[표 2-2] B형 방식	5
[표 2-3] C형 방식	6
[표 2-4] cube 방식의 자소 구성	7
[표 2-5] cube 방식[6]	8
[표 3-1] 한글 40개 자소	9
[표 3-2] 자소당 입력 타수(회) (A, B, C, cube)	9
[표 3-3] cube-3A 방식의 자소 구성	10
[표 3-4] cube-3A 방식	11
[표 3-5] cube 방식과 cube-3A 방식 입력 타수(회)	11
[표 3-6] cube-3B 방식의 자소 구성	12
[표 3-7] cube-3B 방식	12
[표 3-8] cube-3C 방식의 자소 구성	13
[표 3-9] cube-3C 방식	13
[표 3-10] cube-4A 방식의 자소 구성	14
[표 3-11] cube-4A 방식	15
[표 3-12] cube-4B 방식의 자소 구성	16
[표 3-13] cube-4B 방식	17
[표 4-1] 한글 입력 상태 설명	21
[표 5-1] 자소당 입력 타수(회) cube, cube-3, cube-4	25
[표 5-2] cube 방식과 cube-4B 방식 평균 반응 시간(초)	26
[표 5-3] 큐브 입력기 사용자의 GOMS 구성 요소	26
[표 5-4] 큐브 입력기 사용자의 조작 시간(초)	27
[표 5-5] cube 방식과 cube-4B 방식 GOMS 모델 비교	28
[표 부-1] 한글 자소 사용 빈도	34

국문초록

휴대용 터치스크린 장치를 위한 한글 자소 배치

컴퓨터 학과 鄭賢基

指導教授 梁承民

스마트 폰이 널리 보급됨에 따라 터치 스크린에서 한글을 효율적으로 입력하는 방법의 연구가 필요하다. 아직도 기존 전화기에서 사용되는 방법이나 컴퓨터에서 사용되는 방법이 주로 사용된다. 본 논문에서는 기존에 제안된 큐브(cube) 방식을 개선한 3×3 배열 방식과 4×3 배열 방식을 제안한다. 이는 입력 타수를 줄이고 사용자가 인식을 쉽게 하여 편리성을 증가시키기 위함이다. 성능 평가를 위해 안드로이드가 탑재된 넥서스원(Nexus One)에서 구현하여 시험하였다. 본 논문의 결과는 향후 터치 스크린에서의 한글 입력 방법을 개선하는데 큰 도움이 될 것이다.

ABSTRACT

A model of Korean grapheme for portable touch screen device

Jung, Hyun Gi
Department of Computing
Graduate School
Soongsil University

As the smart phones are used widely, the research on efficient Korean character input method for touch screen is needed. The methods used in traditional mobile phones or computer terminals are still.

In this paper, we propose 5 different dispositions that are based on 3×3 and 4×3 arrays. The goal is to minimize the number of typing for Korean character input and to improve the cognition of graphemes. For performance analysis, the proposed dispositions were implemented on Nexus One with Android. The result of this paper contributes to improvement of Korean character input methods for touch screen.

제 1 장 서 론

전화기로 통화 이상의 기능을 바라는 수요가 늘면서 휴대 가능한 전화기에 PC에서 제공되는 기능을 강화한 스마트폰[1]이 등장해 널리 퍼지고, 국내에도 아이폰(iphone)과 안드로이드폰(android phone)이 등장함에 따라 스마트폰에 대한 관심이 높아져 가고 있다[2,3,4].

스마트폰에서 주로 터치스크린을 입력 인터페이스로 사용하게 되면서 터치스크린 상에서의 한글 입력 방식이 절실히 필요하게 되었다.

국내 시장에서의 한글 입력 방식은 기존 버튼 휴대전화에서 사용하던 방식이나 PC에서 사용하던 방식을 그대로 채용하고 있어 각각의 장단점을 모두 갖고 있는 상태이다[5].

이에 본 논문에서는 터치스크린 방식에서 새롭게 제시된 cube 방식[6]을 선택하여 cube 방식에서 안고 있는 단점을 개선할 것이다.

cube 방식에서 제안된 터치스크린의 특징은 살리면서 입력 성능은 높이는 배치를 제안하고 이에 대한 분석을 제시한다.

1.1. 연구목적

기존 휴대 기기상에서 한글 입력 방식의 단점인 타수나 종성과 초성의 충돌 문제를 극복하는 방법인 cube 방식을 제시하려고 한다. 새로운 인터페이스인 터치스크린 방식의 장점을 활용하여 제시된 cube 방식의 장단점을 분석하고 단점을 보완한 방식을 제시하는게 본 논문의 목적이다.

1.2. 기존의 방법들

분산형 한글 자소배치[7], 또는 대화형 한글 입력 방식[8]을 제시하는 논문에서는 <표 1-1>과 같은 4x3 크기의 자판에서 기존[9]과는 양상이 다른 자소 배치를 제시한다.

표 1-1 분산형 한글
배치[7,8]

ㄱ ㅋ	ㄴ ㄸ	ㄷ ㅌ
ㅊ		ㅊ
ㄹ ㄺ	ㅇ ㅅ	ㅂ ㅍ
		ㅈ
ㅅ ㅈ	ㅇ ㅈ	ㅈ ㅈ
다음	ㅎ	선택

기존에는 모음은 모음만 있는 칸에, 자음은 자음만 있는 칸에 자소들이 있었지만 분산형 배치 방식은 자음과 모음의 위치를 제한하지 않아서 칸 하나에 자소가 3개까지 뭉쳐있는 경우는 없어지게 되었다. 글자를 조합하는 방식도 독특하다. 누른 횟수에 따라서 나올 수 있는 글자들을 제시하는데 사용 빈도에 따라 가장 많이 나오는 문자를 후보 중 가장 앞 쪽에 배치하게 된다. 예를 들어, 왼쪽 맨 윗 칸을 'ㄱ'이라고 하자. 'ㄱ'을 한 번 누르면 후보 화면에 'ㄱ', 'ㅋ', 'ㅊ'가 보여지고 'ㄱ'이 입력된다. 여기서 '다음'을 누르면 'ㅋ'에 초점이 가고 '선택'을 누르면 'ㅋ'이 입력된다.

'ㄱ'을 두 번 누르면 '가', '카', 'ㄱ' 이 후보로 나오고, 'ㄱ'을 세 번 누르면 '까', '각', '깅' 이 후보로 나오고, 'ㄱ'을 네 번 누르면 '깅'이 후보로 나오게 된다.

이 방식은 사용 빈도에 따라 맨 처음에 선택되는 자음들이 바로 출력되어 입력 속도가 빨라지는 장점이 있는 반면 사용빈도가 낮은 자소들이 연속될 때 '다음'을 여러 번 눌러야하고, 추가적인 기억장소가 필요하다는 단점이 있다.

각 자소의 배치보다 자소를 이루는 획의 조합에 중점을 둔 논문[10]에서

는 각 자소를 이루는 획을 크게 3가지로 분해하여 초성, 중성, 종성의 순서로 배치하고, 획수가 많은 'ㄷ', 'ㄱ'과 분해가 안되는 'ㅇ'을 따로 입력할 수 있게 배치한 방법을 쓴다.<표 1-2>

표 1-2 획조합 방식[10]

1 —	2	3 人
4 —	5 •	6
7 —	8	9 人
* ㄷ	0 ㅇ	# ㄱ

입력하는 방법은 'ㄱ'이라면 '12'를 누르고 'ㅇ'이라면 '0'을, 'ㅏ'라면 '65'를 누르는 방식이다. 단, 쌍자음인 경우 홀자음을 입력하고 '0'을 누르면 쌍자음이 되는 방식이다. 'ㄱ'은 '212'까지만 입력해도 'ㄱ'으로 인식되고 '#'을 눌러도 'ㄱ'이 입력된다.

이 방식은 자판의 위치를 외우는 부담이 줄어들고, 획이 조합되는 방식이라 직관적으로 초보자도 접근하기 쉬운 장점이 있다. 이 논문에 의하면 중국어, 일어에도 적용할 수 있다고 한다.

하지만 한 글자를 입력하기 위해 자판을 누르는 횟수가 기존에 비해 많이 필요해서 속도면에서는 불리한 단점이 있다.

1.3. 연구방법

기존 휴대전화에서 많이 쓰였던 3가지 방식[9]의 장단점에 대한 자료를 수집해서 이와 연관지어 cube 방식에서 발생할 수 있는 문제점을 분석한다. 이러한 문제점을 최소화하는 3×3 배치와 4×3 배치를 제시하고 이를 실제로 평가해볼 수 있도록 안드로이드 플랫폼 상에서 구현하였다.

제 2 장 휴대 전화 및 PDA 한글 입력방식

한글 입력 가능한 휴대 기기중 휴대전화, PDA 등에서 주로 사용하는 한글 입력 방법은 입력을 인식하는 방법에 따라 크게 버튼 누름 방식과 터치 패드(touch pad) 방식이 있고 자판 배열 종류에 따라 4×3 방식, 데스크탑 IBM PC 호환 쿼티(qwerty) 방식 등으로 나뉜다.

2.1. 버튼 누름 방식에서의 배열

2.1.1. A형 방식

<표 2-1>을 보면 자음은 좌측 및 중앙 열에, 모음은 우측 열 및 하단에 배치되어 있다. 자음은 'ㄱ','ㄴ','ㄷ','ㄹ','ㅁ','ㅂ','ㅅ','ㅇ'의 기본 자음만 배치되어 있고 나머지 자음들은 기본 자음에 획 추가 버튼을 후속 입력하여 다른 자음 조합하는 방식으로 되어 있다. 쌍자음은 기본 자음에 쌍자음 버튼을 후속 입력하여 입력한다.

복모음은 배치된 모음들을 연달아 눌러 조합한다.

자음 입력과 'ㅈ','ㅊ','ㅊ','ㅊ' 모음 입력이 직관적인 장점을 갖고 있다.

표 2-1 A형 방식

ㄱ	ㄴ	ㅏ ㅑ
ㄷ	ㄹ	ㅓ ㅕ
ㅁ	ㅂ	ㅗ
ㅅ	ㅇ	ㅜ
획추가	—	쌍자음

2.1.2. B형 방식

<표 2-2>를 보면 모음이 첫 번째 행에 자음이 그 아래에 배치되어 있

다. 자음은 비슷한 발음을 갖는 모임끼리 배치되어 있고, 쌍자음은 해당 자음의 홀자음이 존재하는 버튼을 3번 누름으로 입력할 수 있다. 'ㅎ' 과 'ㅅ' 이, 'ㄱ' 과 'ㅇ' 이 같이 배치되어 있는 점과 ㄱㄴㄷ 순의 배치가 특징이다. A형 방식에 비해서 쌍자음 입력이 직관적이다.

모음은 'ㅣ', '·', 'ㅡ' 의 조합으로 입력하여 직관적인 모음 입력을 할 수 있다. 할당된 버튼이 3개 밖에 안되는 장점도 있으나 모음이 복잡하면 타수가 늘어나는 단점이 있다. 단어입력시 앞 글자의 종성과 뒷 글자의 종성이 같은 경우 추가 처리가 필요한 단점도 있다.

표 2-2 B형 방식

ㅣ	·	ㅡ
ㄱ ㅋ ㄴ	ㄴ ㄹ	ㄷ ㅌ ㄸ
ㅂ ㅃ ㅍ	ㅅ ㅎ ㅆ	ㅈ ㅊ ㅊ
	ㅇ ㅁ	

2.1.3. C형 방식

<표 2-3>를 보면 자음은 좌측 및 중앙 열에 배치되어 있고 모음은 우측 열과 상단 행에 배치되어 있다.

B형 방식과 유사하게 발음이 비슷한 자음끼리 배치되어 있고, 쌍자음은 해당 자음의 홀자음이 존재하는 버튼을 3번 누름으로 입력할 수 있다.

'ㅎ' 과 'ㅇ' 이, 'ㄱ' 과 'ㅅ' 이 같이 배치되어 있는 점과 'ㄱ', 'ㄷ', 'ㄱ', 'ㅈ' 이 좌측 열에, 'ㄴ', 'ㅂ', 'ㅇ' 이 중앙 열에 배치되어 있는 점이 특징이다.

B형 방식과 마찬가지로 쌍자음 입력은 A형 방식에 비해 직관적이거나 단

어입력시 앞 글자 종성과 뒷 글자 종성이 같을 경우 추가적인 처리가 필요한 단점도 있다.

복모음은 배치된 모음들을 연달아 눌러 조합한다.

표 2-3 C형 방식

ㄱ ㅋ ㄲ	ㅣ ㅡ ㅗ ㅛ	ㅜ ㅠ ㅝ ㅠ
ㅓ ㅖ ㅗ	ㄴ ㄹ ㅕ ㅋ	
ㅓ ㅖ ㅗ	ㅕ ㅖ ㅗ	ㅓ ㅖ ㅗ
ㅗ ㅛ ㅗ	ㅕ ㅖ ㅗ	ㅓ ㅖ ㅗ

2.1.4. 쿼티(qwerty) 방식

IBM PC 호환 자판의 두벌식 한글 배치와 동일해서 기존의 두벌식 한글 자판에 익숙한 사용자에게는 자판을 익히는 시간이 불필요하다.<그림 2-1>

자음이 왼쪽, 모음이 오른쪽에 배치되어 있고 쌍자음과 겹자음을 제외하곤 한 번의 누름으로 입력할 수 있는 장점이 있다.

하지만 익숙함에도 불구하고 이 익숙함이 시각적 익숙함보단 손가락 감각의 익숙함에 의존하고 있어 휴대전화 특성상 두 손을 모두 올려놓고 입력하는 기존 방식을 사용할 수 없으므로 사람에 따라 각각의 자소를 찾는데 걸리는 시간이 더 드는 경우도 있다. 게다가 버튼 누름 방식은 휴대기기 크기 제약상 버튼이 작아 빠른 속도로 정확하게 입력하기 힘들다는 단점도 있다.

배 ㅂ	ㅈ ㅈ	ㅊ ㅊ	ㅋ ㅋ	ㅌ ㅌ	ㄴ ㄴ	ㄷ ㄷ	ㄹ ㄹ	ㅍ ㅍ	ㅎ ㅎ
ㄱ ㄱ	ㄴ ㄴ	ㅇ ㅇ	ㄷ ㄷ	ㅎ ㅎ	ㄱ ㄱ	ㄴ ㄴ	ㅍ ㅍ	ㅣ ㅣ	
ㅋ ㅋ	ㅌ ㅌ	ㅍ ㅍ	ㅊ ㅊ	ㅌ ㅌ	ㅍ ㅍ	ㅌ ㅌ	ㅍ ㅍ		

그림 2-1 두벌식 한글 자판 배치

2.2. 터치 패드(touch pad) 방식에서의 배열

2.2.1. 쿼티(qwerty) 방식

버튼 방식의 쿼티 방식과 장단점이 거의 같다고 볼 수 있다. 다른 점은 버튼 크기가 작아 손가락으로 빠르고 정확하게 입력해야하는 부분을 접촉 면적이 작은 펜을 사용하여 극복할 수 있다는 점이다. 하지만 "독수리 타법"처럼 입력하기 때문에 사람에 따라 자소를 찾는데 걸리는 시간이 많이 소모될 수 있다.

2.2.2. cube 방식[6]

표 2-4 cube 방식의 자소 구성

화면 위치	포함 자소
위쪽	ㅋ, ㅌ, ㅍ ㄱ 계열 모음 및 ㅣ
왼쪽	ㄱ, ㅌ ㄴ 계열 모음 및 ㅡ
가운데	기본자음 ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ
오른쪽	ㅊ, ㅌ, ㅍ ㅍ 계열 모음 및 ㅡ
아래쪽	ㅊ, ㅌ, ㅎ, ㅌ ㅌ 계열 모음 및 ㅣ

cube 방식은 3×3 배열로 이루어져 있다.

입체 정사각형 모양의 cube 처럼 3×3 배열이 상,하,좌,우,중앙 에 각각 위치하고 있고, 기본 화면으로 사용되는 중앙에 사전순으로 자음 9개를 배치하고 있다. 나머지 자음은 기본 자음과 연관되는 위치에 배치하고 있다.<표 2-4><표 2-5>

스크래치(굵음)되는 방향에 따라 해당하는 면이 표시가 되고, 다시 스크래치하면 기본화면인 중앙에 돌아오게 된다.

어떤 면이든 스크래치가 아닌 단순 누름이면 입력으로 받아들이고 해당 자소를 선택하고 기본 화면인 중앙으로 돌아오게 된다.

표 2-5 cube 방식[6]

			ㄱ			ㅌ		
						ㅍ		
			ㄴ	ㅇ	ㅣ			
ㄷ	ㅅ	ㄹ	ㄷ	ㄴ	ㄷ	ㅌ	ㅍ	ㅌ
	ㅅ	ㅅ	ㄷ	ㅍ	ㅍ	ㅌ	ㅍ	ㅌ
ㅅ		ㅡ	ㅅ	ㅇ	ㅌ	ㅡ		ㅌ
			ㅌ	ㅍ	ㅣ			
						ㅌ		
			ㅅ	ㅇ	ㅌ			

제 3 장 cube 방식 개선 방안

논문[6]에서 제안된 cube 방식은 2장 입력방식에서 언급된 B, C 방식과 같이 앞 글자의 종성과 다음 글자의 초성이 같을 경우 방향키를 눌러 다음 위치로 이동하거나 특정 시간 이상 대기해서 다음 위치로 이동시켜야 하는 문제가 있다. 이런 문제들은 한 버튼을 두 번 이상 눌러서 새로운 자소를 입력할 수 있을 경우 발생하는데 cube 방식에서도 단자음을 두 번 눌러 쌍자음을 입력하는 방식을 사용하므로 이러한 문제가 발생한다. 복자음과 복모음의 조합으로 인한 타수 증가 문제, 중복되는 글쇠들과 배치로 인한 공간 낭비, 그리고 인지도를 높이는 자소 배치를 할 수 없다는 것이 단점이다. 이러한 문제들을 해결하기 위한 배열을 3×3 크기에서 제안을 하고, 3×3 에서의 공간 제약으로 인한 자소 인지도 감소를 극복하기 위한 4×3 배열을 제안한다.

3.1. 평균 타수 문제

표 3-1 한글 40개 자소

ㄱ	ㄴ	ㄷ	ㄹ	ㅁ	ㅂ	ㅅ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	ㅊ	ㅊ	ㅊ	ㅊ	ㅊ	ㅊ
ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅒ	ㅔ	ㅖ	ㅘ	ㅙ	ㅚ	ㅜ	ㅞ	ㅟ	ㅠ	ㅡ	ㅢ

표 3-2 자소당 입력 타수(회) (A, B, C, cube)

타수 방법	1	2	3	4	5	합계	평균
A형	10	15	13	2	0	40	2.17
B형	9	12	13	4	2	40	2.45
C형	12	18	10	0	0	40	1.95
cube	9	24	0	7	0	40	2.13

<표 3-1>과 같은 40개 자소를 입력하는 경우 기존의 제시된 방식은 각 자소당 누르는 횟수(타수)가 평균 1.9회에서 2.45회 까지 필요하다.<표 3-2>

평균적인 자소당 입력 타수를 보면 cube 방식은 C형 방식에 버금가는 성능을 보여준다.

하지만 기존 입력 방식들을 보면 평균 타수와 상관 없이 한 개의 자소를 입력하기 위해 적게는 1타에서 많게는 5타까지 자소별로 편차가 있음을 알 수 있다.

특히 복자음과 복모음의 입력에 많은 조합수가 필요하다.

cube 방식의 경우 굵는 동작(scratch)을 한 개의 타수로 본다면 복자음과 복모음의 입력에 2번의 굵는 동작이 포함되어 4타가 된다.

3.2. 3×3 배열

3.2.1. cube-3A 방식

표 3-3 cube-3A 방식의 자소 구성

화면 위치	포함 자소
위쪽	겹자음 ㄱ, ㄲ, ㅃ, ㅅ, ㅆ, ㅈ, ㅊ, ㅋ
왼쪽	기타 자음 및 기타 모음 ㅌ, ㄷ, ㅌ, ㅍ, ㅎ, ㅡ, ㅣ, ㅑ
가운데	기본자음 ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ
오른쪽	모음 ㅓ, ㅕ 계열
아래쪽	모음 ㅗ, ㅛ 계열

기존의 cube 방식에서 복자음 및 복모음을 입력할 때 굵는 동작(scratch) 포함하여 4타가 필요하다. 이를 줄이기 위해서 <표 3-3>의 자

소배치를 제안한다. 기본 화면에서 조합이 안 되는 복자음을 상단 화면에 배치한다.<표 3-4>. 복모음들은 우측 화면과 하단 화면에 배치하였다.

표 3-4 cube-3A 방식

			ㄱ	ㄴ	ㄷ			
				ㄹ	ㅁ			
			ㅂ	ㅅ	ㅈ			
ㅋ	ㆁ	ㅌ	ㄱ	ㄴ	ㄷ	ㄷ	ㅌ	ㅍ
ㅣ	ㅡ	ㅍ	ㄹ	ㅁ	ㅂ	ㄷ	ㅌ	ㅍ
	ㅎ	ㅊ	ㅅ	ㅇ	ㅈ	ㄷ	ㅌ	ㅍ
			ㅊ	ㅇ	ㅈ			
			ㅊ	ㅌ	ㅈ			
			ㅊ	ㅌ	ㅈ			

<표 3-5>를 보면 각 자소를 입력할 때 굵는 동작(scratch) 포함한 평균 입력횟수가 감소했음을 알 수 있다. 자소의 최대 타수의 경우 cube 방식은 4타인데 cube-3A 방식은 2타이다.

표 3-5 cube 방식과 cube-3A 방식 입력 타수(회)

타수 방법	1	2	3	4	5	합계	평균
cube	9	24	0	7	0	40	2.13
cube-3A	9	31	0	0	0	40	1.77

3.2.2. cube-3B 방식

표 3-6 cube-3B 방식의 자소 구성

화면 위치	포함 자소
위쪽	겹자음 ㄱ, ㄲ, ㅃ, ㅅ, ㅆ, ㅈ, ㅊ, ㅋ
왼쪽	모음 ㅏ, ㅑ 계열
가운데	기본자음 ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅎ
오른쪽	모음 ㅓ, ㅕ 계열
아래쪽	기타 자음 및 기타 모음 ㅂ, ㅅ, ㅋ, ㅌ, ㅍ, ㅡ, ㅗ, ㅣ

cube-3B 방식에서는 굵는 방향의 일관성을 위해 모음은 좌우 화면에, 자음은 상하 화면에 배치하였다.<표 3-6><표 3-7>

또한 중앙 화면에 빈도수가 높은 'ㅎ'을 배치하고 'ㅂ'은 하단 화면으로 옮겼다. 가로로 굵으면 모음, 세로로 굵으면 자음이 나온다는 일관성을 확보했다.

표 3-7 cube-3B 방식

			ㄱ	ㄲ	ㄴ			
				ㄷ	ㄸ			
			ㅏ	ㅑ	ㅓ			
ㅕ	ㅗ	ㅛ	ㄱ	ㄴ	ㄷ	ㅕ	ㅑ	ㅓ
ㅗ	ㅓ	ㅛ	ㄷ	ㄴ	ㅓ	ㅕ	ㅑ	ㅓ
ㅕ	ㅓ	ㅛ	ㅇ	ㅈ	ㅎ	ㅕ	ㅑ	ㅓ
			ㅋ	ㅌ	ㅍ			
			ㅣ	ㅂ	ㅍ			
			ㅡ	ㅗ				

3.2.3. cube-3C 방식

표 3-8 cube-3C 방식의 자소 구성

화면 위치	포함 자소
위쪽	겹자음 ㄱ, ㄷ, ㅁ, ㅂ, ㅅ, ㅈ, ㅊ, ㅋ, ㆁ
왼쪽	모음 ㅏ, ㅑ 계열
가운데	기본자음 ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅎ
오른쪽	모음 ㅓ, ㅕ 계열
아래쪽	기타 자음 및 기타 모음 ㅖ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ, ㅣ

이전까지의 cube 방식에서 모음 배치는 최대한 비슷한 모양끼리 모아서 배치하다. 하지만 ‘ㄱ’, ‘ㄷ’ 계열모음의 자소 수가 월등히 많아서 익숙하지 않은 경우에 찾는 시간이 다소 걸리는 문제가 있다. 그래서 각각의 화면에서 ‘ㄱ’, ‘ㄱ’를 왼쪽 화면의 좌, 우 열에, ‘ㄷ’, ‘ㄷ’를 오른쪽 화면의 좌, 우 열에 두고 사전순으로 배치하여 모음 조화[11]에 기반한 배치가

표 3-9 cube-3C 방식

			ㄱ	ㄴ	ㄷ			
				ㄹ	ㅁ			
			ㅅ	ㅇ	ㅈ			
ㅊ	ㅋ	ㆁ	ㄷ	ㄴ	ㄷ	ㄱ	ㅋ	ㅌ
ㅅ	ㅇ	ㅈ	ㄹ	ㅁ	ㅅ	ㅅ	ㅌ	ㅈ
ㅈ	ㅊ	ㅊ	ㅇ	ㅈ	ㅎ	ㅅ	ㅈ	ㅈ
			ㅋ	ㄱ	ㅌ			
			ㅣ	ㅅ	ㅌ			
			ㅡ	ㅈ				

cube-3C 방식이다.<표 3-8><표 3-9>

3.3. 4×3 배열

3×3 배열에서 모음 자소의 수가 자음 자소의 수보다 많아 전체 모음 자소 중 3개는 자음 자소와 같이 배치될 수밖에 없었다. 이는 모음과 자음이 서로 섞여 있는 화면에서 자소를 인식하는 어려움이 있다. 이 문제는 3×3 배열을 4×3 배열로 바꾸면 해결할 수 있다.

4×3 배열에서는 기본 화면에 배치되는 초성의 수가 증가하여 평균 타수가 줄어든다. 또한, cube-3B 방식에서 겹자음을 조합하지 않아서 조합대상을 찾는 시간과 동선이 줄어든다. 그리고 한 화면에 두 계열의 모음을 모두 집어넣을 수 있어서 모음 일관성이 3×3 배열에 비해 뚜렷해진다.

3.3.1. cube-4A 방식

표 3-10 cube-4A 방식의 자소 구성

화면 위치	포함 자소
위쪽	겹자음 ㄹ 로 시작됨
왼쪽	모음 ㅏ,ㅓ,ㅣ 계열
가운데	기본자음 ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ,ㅅ,ㅇ,ㅈ,ㅊ,ㅌ,ㅎ
오른쪽	모음 ㅑ,ㅕ,ㅡ 계열
아래쪽	겹자음 ㄹ 로 시작되지 않음

‘ㄹ’로 시작하는 모든 겹자음을 위쪽 화면에 몰아넣고, 남은 겹자음을 하단 화면에 집어넣어 조합할 필요없이 종성을 한번에(꺾는 동작을 제외했

을 경우) 입력할 수 있게 했다.<표 3-10>

겹자음을 입력할 때의 동선을 최소화하기 위해 위쪽 겹자음들은 위쪽에 최대한 배치하고 아래쪽 겹자음들은 아래쪽에 최대한 배치하도록 했다.

모음은 좌측 면에 ‘ㄱ’, ‘ㄷ’, ‘ㄴ’ 계열 모음을, 우측 면에 ‘ㅏ’, ‘ㅑ’, ‘ㅓ’ 계열 모음을 각각 세로로 배치해서 인지성을 높였다.<표 3-11>

표 3-11 cube-4A 방식

			ㄹㄱ	ㄹㅅ	ㄹㅊ			
			ㄹㅈ	ㄹㅌ	ㄹㅍ			
				ㄹㅎ				
ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅗ	ㅑ	ㅓ
ㅕ	ㅗ	ㅛ	ㅕ	ㅗ	ㅛ	ㅗ	ㅑ	
ㅗ	ㅛ	ㅓ	ㅕ	ㅗ	ㅛ	ㅗ	ㅑ	
ㅛ	ㅓ	ㅕ	ㅕ	ㅗ	ㅛ	ㅗ	ㅑ	
			ㅕ		ㅛ			
			ㅕ	ㅓ	ㅛ			
			ㅕ	ㅓ	ㅛ			
			ㅕ	ㅓ	ㅛ			

3.3.2. cube-4B 방식

표 3-12 cube-4B 방식의 자소 구성

화면 위치	포함 자소
위쪽	겹자음 ㄹ 로 시작됨
왼쪽	모음 ㅏ, ㅓ, ㅣ 계열
가운데	기본자음 ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅈ, ㅋ, ㅌ, ㅎ
오른쪽	모음 ㅑ, ㅕ, ㅡ 계열
아래쪽	겹자음 ㄹ 로 시작되지 않음

자음 배치는 cube-4B 방식과 같으나 모음 배치를 바꾸었다.

가로 방향으로 유사계열 모음들이 배치되었고, ‘ㅓ’가 ‘ㅏ’와, ‘ㅕ’가 ‘ㅑ’와 함께 배치된 점이 크게 다르다.<표 3-12>

기존 입력 실험 때 ‘ㅓ’, ‘ㅕ’ 계열 모음이 함께 있어 ‘내’, ‘네’ 등등의 비슷한 모음들이 붙어 있는 경우 인지 시간이 길어지는 경우가 많았기 때문에 서로 분리하게 되었다.

그리고 복모음보단 단모음이 출현 빈도가 높은 이유로 굵는 방향에 가깝게 배치하느라 오른쪽 화면의 모음 배치의 단모음과 복모음 위치는 왼쪽과는 반대로 되었다.<표 3-13> cube-3C처럼 모음조화에 기반하여 바뀐기 때문에 모음을 ‘내’, ‘네’와 같이 비슷한 모음과 착각하지 않고, 굵고 난 위치에 출현 빈도가 높은 모음이 위치하여 이동 비용이 줄어드는 장점이 있다.

표 3-13 cube-4B 방식

			ㄹㄱ	ㄹㅁ	ㄹㅂ			
			ㄹㅅ	ㄹㅆ	ㄹㅈ			
				ㄹㅎ				
ㅏ	ㅑ	ㅓ	ㄱ	ㄴ	ㄷ	ㅈ	ㅊ	ㅋ
ㅕ	ㅗ	ㅛ	ㄴ	ㄷ	ㅈ	ㅊ	ㅋ	ㅌ
ㅗ	ㅛ		ㅅ	ㅇ	ㅈ		ㅊ	ㅋ
ㅣ	ㅑ		ㅋ	ㅌ	ㅎ	ㅊ	ㅋ	ㅌ
			ㄱ		ㄷ			
			ㅏ	ㅑ	ㅓ			
			ㅕ	ㅗ	ㅛ			
			ㅗ	ㅛ	ㅕ			

제 4 장 구현

3장에서 제안한 배열 방법에 대한 성능을 평가하기 위해서 안드로이드가 탑재된 HTC사의 Nexus One에 한글 입력기를 구현하였다. 이 입력기는 안드로이드 SDK 소스[12]에 포함되어 있는 영문 쿼티(qwerty) 자판을 입력할 수 있는 예제 입력기 소스를 참고하여 구현하였다.

4.1. 개발 환경

리눅스 상에서 안드로이드 SDK[12]를 포함한 자바 개발환경을 사용하여 구현하였다.

사용한 리눅스 배포판은 우분투(Ubuntu) 10.04이고 자바 환경과 통합 개발 환경(IDE)을 사용하기 위해 sun-java6-jdk, eclipse-jdt 패키지를 설치했다. 사용된 안드로이드 SDK 의 버전은 linux-i386 r06이다. 실행 경로를 지정하는 환경 변수에 SDK 디렉토리의 tools 디렉토리 경로를 추가해줘야 한다.

통합 개발 환경으로 선택한 이클립스(eclipse)에서 안드로이드 프로그래밍을 원활히 하기 위해 JDT(Java Development Tool)와 WST(Web Standard Tool) 플러그인을 설치하고 구글에서 안드로이드 관련 플러그인인 ADT(Android Development Tool)[13]를 설치하였다.

구축환경에 언급된 부분들을 순서대로 갖추면 실제 안드로이드 기기가 없어도 프로그램을 실행시켜 볼 수 있는 달빅 시뮬레이터(Dalvik Virtual Machine)까지 실행시킬 수 있는 환경이 구축된다.

4.2. 프로그램 구성

<그림 4-1>는 프로그램의 구성을 보여준다. 예제 입력기 소스에는 실행 환경을 정의하는 AndroidManifest.xml 파일과 실행될 때 필요한 자원들

중 레이아웃 정의가 포함되어 있는 res/xml 디렉토리와 실제 자바 소스 파일이 들어있는 src 디렉토리로 구성되어 있다.

src 디렉토리에 포함되어 있는 맨 처음 실행되는 SoftKeyboard.java 파일에서 안드로이드 입력기 서비스 생명 주기(Input Method Service Life Cycle)를 따라 실행된다. 큐브 한글 입력기에서 추가한 부분은 한글 배치를 읽어오는 부분, 여러 개의 연속된 한글 자소를 완성된 하나의 글자로 되돌려주는 한글 상태 처리 부분, 특정 방향으로 굽었을 때 정책에 따라 자판을 바꿔주는 세 부분으로 나뉜다. 한글 배치를 읽어오는 부분은 SoftKeyboard.java의 생성자에서, 한글 상태 처리 부분은 추가 메소드에서, 굽는 행위를 인지하는 부분은 KeyboardView.java의 이벤트 메소드에서 처리한다.

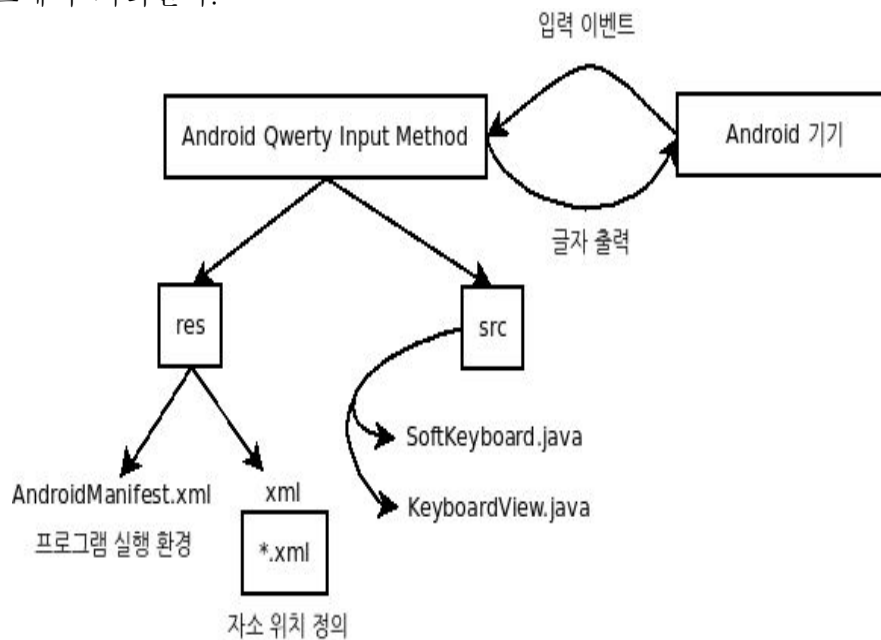


그림 4-1 프로그램 구성

4.3. 한글 자소 조합 알고리즘(Hangul Automata)[14]

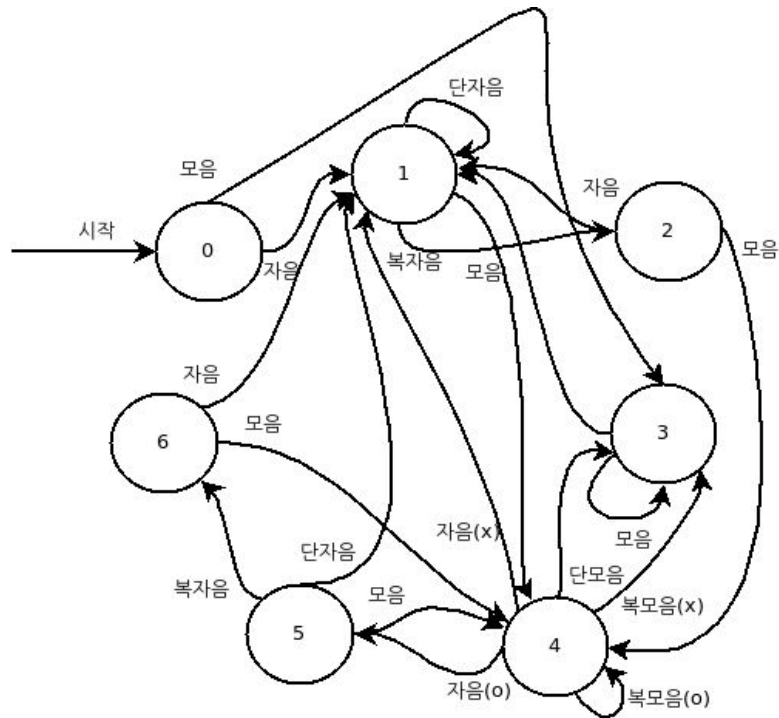


그림 4-2 한글 상태 전이도

구현하면서 실제로 오류가 가장 많이 발생하고 있는 부분으로 연속된 한글 자소를 분석해서 초성 중성 종성의 규칙에 맞으면 각 자소를 하나의 글자로 바꿔주는 부분이다. 이를 한글 오토마타(automata)라고도 한다. 한글 조합 방법은 수도 없이 많다. 제안한 사람이나 한글을 입력하는 환경 등에 따라 조합 방법은 여러 가지가 존재할 수가 있다. 본 논문에서는 두벌식 한글 자판 입력 환경에서 기반한 7가지 상태를 갖는 조합 방법[15]을 사용한다.

각 상태에서 한글 자소가 입력으로 들어온다.

원래는 단자음, 복자음, 단모음, 복모음으로 모든 입력을 표시해야 하는데 단자음 복자음이나 단모음, 복모음이 같은 상태로 이동하는 경우는

간단히 각각 자음, 모음으로 합쳤다.

각 상태에 대한 설명은 <표 4-1>과 같다.

표 4-1 한글 입력 상태 설명

상태	설명	다음 상태
0	한글 입력이 시작됨	1,3
1	초성이 한 개 들어옴	1,2,4
2	복자음이 초성으로 한 개 들어옴	1,4
3	중성이 한 개 입력으로 들어옴	1,3
4	초성과 중성이 들어옴	1,3,4,5
5	초성과 중성, 종성이 들어옴	1,4,6
6	초성과 중성, 복자음 종성이 들어옴	1,4

4.4. 실행 결과

안드로이드 달빅 시뮬레이터(Dalvik Simulator)에서 검색창을 통해 실행시킨 모습과 실제 안드로이드 기기(Nexus One)에서 실행시킨 결과이다.



그림 4-3 시뮬레이터에 실행된 화면

<그림 4-3>은 개발환경에서 시뮬레이터를 통해 구동시킨 화면이다.

이 화면의 오른쪽은 하드웨어 키들을 미리 배치해 놓은 것이고, 왼쪽이 실제 안드로이드의 내용이 출력되는 부분이다.

구현시 하드웨어 키들은 고려하지 않았기 때문에 테스트는 왼쪽 화면에서만 이루어졌다.

시뮬레이터에서는 굵는 동작을 마우스로 입력해야 되어서 실제와는 어느 정도 성능 차이가 날 수 밖에 없었다.

<그림 4-4>는 안드로이드 v2.2 가 설치되어 있는 HTC 사의 Nexus One에서 실행시킨 화면이다.



그림 4-4 안드로이드
기기에서 구동시킨 화면

제 5 장 평가

휴대전화 자판에 대한 성능 평가 방법에는 자소당 키 입력 타수를 비교하는 방법, 물리적인 손가락 이동 시간을 측정하는 방법(Fitt's Law)[16], 선택할 개수에 따른 인지시간을 측정하는 방법(Hick's Law)[16,17], 단위 행위들에 대한 예상 시간을 바탕으로 단위 행위들이 어떻게 이뤄지는지 분석하고 그 시간을 측정하는 방법(GOMS model)[16,18] 등이 있다.

자소당 키 입력 타수 비교 방법은 'ㄱ'같이 한 번의 타수를 소비하는 자소와 '내'같이 자판에 따라 두 번 이상의 타수를 소비하는 자소들이 있는데 각 한 자소당 소비되는 평균 타수를 구해서 평가하는 방법이다.

Fitt's Law는 인간의 손이 시작지점에서 목적지점으로 움직이는데 걸리는 시간을 예측하기 위한 제시된 수학적 모델로, 손이 움직인 거리(A)와 목적지 목표물의 너비(W)의 관계를 밝혀 운동시간을 예측한다. 본 논문에서 제안하는 cube 개선 방식들은 화면이 전환되고 화면을 긁고 난 뒤의 손가락의 위치가 불규칙하여 객관적인 자료 수집이 어려우므로 물리적인 손가락 이동 시간 측정 방법은 사용하지 않았다.

Hick's Law(Hick-Hyman's Law)는 선택하는 대상의 개수에 따른 인지 시간을 평가하는 방법으로 주어진 n 개의 선택 중에 원하는 선택을 찾을 때까지의 반응시간(T)을 구한다. 이 시간을 정보 습득률이라 한다. 전체 항목 개수가 n 이고 평균 반응 시간이 T 라 한다면 다음과 같은 공식이 성립한다.

$$T = b \log_2(n+1)$$

b는 상수로써 측정된 자료값을 선형으로 연결했을 때 나오는 기울기이다.

각 선택의 확률이 서로 비슷하지 않은 경우에는 확률 p_i 를 공식에 반영

한다.

$$T = b \sum_i^n p_i \log_2 \left(\frac{1}{p_i} + 1 \right)$$

GOMS 모델은 HIP(Human Information Processor)의 순차 모형에 기반을 두고 있는 기법으로 미리 정의된 행동양식의 예상시간을 비교하여 분석하는 방법이다.

5.1. 자소당 키 입력 타수 비교

한글은 초성, 중성, 종성으로 이루어져 있어 전자적으로 한 글자를 입력하는 경우 적게는 1번에서 많게는 6번까지 키 입력이 이뤄진다. 이것은 두벌식 자판이나 이와 유사한 배치에서 이뤄지는 입력이고, 종성 자판을 고려한 세벌식 자판에서는 적게는 1번에서 많게는 4번까지 이뤄진다. no-shift 세벌식 자판에서는 많게는 3번까지로 줄어든다.

휴대전화에서는 화면 크기의 제약상 세벌식 자판의 이점을 활용하기가 힘들다. qwerty 자판처럼 키보드 자판 전체를 휴대전화에 구현할 경우엔 눌러야 할 버튼의 크기가 상대적으로 작아져서 기존 사용자의 경우 학습 시간이 줄어드는 이점에도 불구하고 각 키를 인지하여 찾는 시간이 추가로 발생한다. 특히 터치 패드 같은 경우는 펜과 같이 끝이 작은 개체를 사용하지 않으면 입력 정확도가 많이 떨어진다는 단점이 생긴다.

키보드 자판 배열이 사용되는 경우도 종종 있지만 위와 같은 이유로 휴대폰의 T9(3×3 또는 4×3) 배열이 사용된다.

제약된 공간에 배치되는 입력 버튼의 수가 9개 또는 12개로 줄어들면 입력 버튼의 크기가 상대적으로 커져서 다른 입력 버튼을 잘못 입력하는 경우가 줄어들기 때문에 평균 입력 속도가 빨라진다. 하지만 40여개의 자소를 9개 또는 12개의 키에 할당함으로 초성, 중성, 종성의 자소를 한

번의 입력하는 방식이 아닌 자음 자소와 모음 자소로 구별되고 한 버튼에 할당된 첫 번째가 아닌 자소를 입력하기 위해 두 번 세 번 누르게 되어 한 글자를 입력할 때 버튼을 누르는 타수가 증가하게 된다.

<표 5-1>을 보면 cube 방식은 기존 버튼 방식들인 A, B, C 형에 버금가는 성능을 보였다면 cube-3 방식과 cube-4 방식은 평균적으로 뛰어난 성능을 보인다.

표 5-1 자소당 입력 타수(회) cube, cube-3, cube-4

방법 \ 타수	1	2	3	4	5	합계	평균
A형	10	15	13	2	0	40	2.18
B형	9	12	13	4	2	40	2.45
C형	12	18	10	0	0	40	1.95
cube	9	24	0	7	0	40	2.12
cube-3 series	9	31	0	0	0	40	1.78
cube-4 series	12	28	0	0	0	40	1.7

5.2. Hick's Law[16,17]

숙련자 한 명과 초급자 한 명을 상대로 ‘다’, ‘했’, ‘고’, ‘는’, ‘을’, ‘를’, ‘가’, ‘오’, ‘래’, ‘저’, ‘왔’, ‘갔’, ‘밑’, ‘삶’ 이라는 글자를 입력할 때 각 자소들을 생각하기 시작해서 자판에서 찾을 때까지의 시간을 기록하였다. 각 자소에 대한 선택의 확률은 모두 같다고 가정하고 Hick's Law (Hick-Hyman's Law) 공식($T = b \log_2(n+1)$)에 대입하여 나온 상수 b 값들의 평균을 구해서 숙련자와 초급자가 입력할 때 걸리는 평균 시간을 구했다.

<표 5-2>를 보면 전체적으로 cube 방식보다 cube-4B 방식을 사용할 때

자소 탐색시 평균적인 반응 시간이 짧게 걸림을 알 수 있다. 숙련자의 경우 cube 방식보다 cube-4B 방식의 반응 시간이 좀 더 짧긴 했지만 크게 차이 나지 않았다. 한글자를 입력하는 경우를 측정한 결과라 차이가 별로 없지만, 긴 문장을 입력할 경우 차이는 커지리라 예상된다.

표 5-2 cube 방식과 cube-4B 방식
평균 반응 시간(초)

숙련도	cube	cube-4B
초급자	4.98	4.19
숙련자	1.89	1.82

5.3. KLM-GOMS Model[16,18]

본 논문에서는 <표 5-3>과 같은 GOMS를 갖는다.

표 5-3 큐브 입력기 사용자의 GOMS 구성 요소

구성 요소	사례
Goal (목표)	여러 자소 중에서 원하는 자소를 입력한다.
Operation (행위)	자소를 누르거나 화면을 긁는다.
Method (방법)	입력기가 실행된 화면에서 원하는 자소를 찾아 누른다.
Selection Rules (선택 규칙)	맨 처음 보이는 자소 배치에서 자소를 찾아 누르고, 현재 자소 배치에 없으면 상하좌우 방향으로 긁어서 자소 배치를 다른 것으로 교체한다.

본 논문에서는 다섯가지 조작을 정의하였다.

표 5-4 큐브 입력기 사용자의 조작 시간(초)

조작	설명	초보자 숙련자
F	현재 화면에 없을 경우 어떤 방향에 있는 자판으로 이동할지 판단하는 시간	2.5 0.8
K	키를 눌렀다 떼는 시간	0.2 0.1
M	누를 자소를 인지하고 눈으로 위치를 찾기까지의 시간	1.5 0.8
P	화면의 자소에 손가락을 이동하여 가리키는 시간	0.2 0.1
S	다른 화면으로 전환하기 위해 화면을 꺾는 시간	0.2

<표 5-5>를 보면 받침이 없거나 받침이 기본 화면에 있는 경우에는 cube 방식과 cube-4B 방식이 동일한 기댓값(예상 시간)을 보여준다. '했' '자' 같은 구조의 글자들에서 초성이 'ㄷ', 'ㄱ', 'ㅌ', 'ㅍ', 'ㅎ' 인 경우에는 cube-4B 방식이 cube 방식보다 약간 더 나은 기댓값을 보여준다. 초급자의 경우 cube 방식에 익숙해져가면서 cube-4B에서 빨라진 점도 있고 앞 글자보다는 뒷 글자를 입력하면서 좀 더 속도가 빨라지는 경향이 있었다. 숙련자의 경우에는 '쟈'자의 경우를 제외하고는 받침이 있는 글자는 cube-4B 방식이 좀 더 빠른 입력시간을 보였다. 다만 받침이 없는 글자는 서로 비슷한 결과를 보였다. 겹받침을 입력해야하는 글자인 경우, cube 방식과 cube-4B 방식간에 큰 성능 차이가 날 것이란 예상과 달리 별 차이가 없었다. 이는 측정을 한 글자씩 입력하였고, cube 방식에서 기본 화면에서 겹받침을 조합하는 시간이 예상보다 빨랐던 반면에 cube-4B 방식에서는 화면 전환하는 입력이 제대로 안되서 시간이 걸렸다고 분석된다.

표 5-5 cube 방식과 cube-4B 방식 GOMS 모델 비교

	cube 소모 행동	cube-4B 소모 행동	초급자				숙련자			
			예상시간		실측평균		예상시간		실측평균	
			cube	4g	cube	4g	cube	4g	cube	4g
다	M+P+K+F+S+M+P+K	M+P+K+F+S+M+P+K	6.5	6.5	3.9	3.7	2	2	4.1	4.2
했	F+S+M+P+K+F+S+M+P+K+K	M+P+K+F+S+M+P+K+F+S+M+P+K	11.3	11.1	12.1	10.7	5.1	5	7.3	6.1
고	M+P+K+F+S+M+P+K	M+P+K+F+S+M+P+K	6.5	6.5	7.3	4.2	3	3	3.1	2.9
는	M+P+K+F+S+M+P+K+M+P+K	M+P+K+F+S+M+P+K+M+P+K	8.4	8.4	10.1	5.9	4	4	5.2	7.6
을	M+P+K+F+S+M+P+K+M+P+K	M+P+K+F+S+M+P+K+M+P+K	8.4	8.4	9.2	7.9	4	4	5.8	5.1
를	M+P+K+F+S+M+P+K+M+P+K	M+P+K+F+S+M+P+K+M+P+K	8.4	8.4	8.7	6.1	4	4	5.3	4.0
가	M+P+K+F+S+M+P+K	M+P+K+F+S+M+P+K	6.5	6.5	4.0	3.3	3	3	2.5	2.7
오	M+P+K+F+S+M+P+K	M+P+K+F+S+M+P+K	6.5	6.5	6.7	3.8	3	3	3.7	2.6
래	M+P+K+F+S+M+P+K	M+P+K+F+S+M+P+K	6.5	6.5	5.4	4.9	3	3	2.7	2.5
저	M+P+K+F+S+M+P+K	M+P+K+F+S+M+P+K	6.5	6.5	8.7	7.7	3	3	4.2	5.6
왔	M+P+K+F+S+M+P+K+M+P+K+K	M+P+K+F+S+M+P+K+F+S+M+P+K	8.6	11.1	9.2	6.9	4.1	5	8.9	5.9
갔	M+P+K+F+S+M+P+K+M+P+K+K	M+P+K+F+S+M+P+K+F+S+M+P+K	8.6	11.1	7.3	7.5	4.1	5	5.2	5.6
밑	M+P+K+F+S+M+P+K+F+S+M+P+K	M+P+K+F+S+M+P+K+M+P+K	11.1	8.4	12.4	13.1	5	4	6.4	4.8
삶	M+P+K+F+S+M+P+K+M+P+K+M+P+K	M+P+K+F+S+M+P+K+F+S+M+P+K	10.3	11.1	8.2	18.2	5	5	5.3	4.9

5.4. 분석 결과

자판이 익숙한 정도가 결과에 가장 큰 영향을 주었다. 성능 평가를 위한 일련의 테스트들을 수행 도중에 익숙한 정도가 점점 증가해서 결과도

점점 빨라졌다. 이런 이유로 다른 요인을 정확하게 파악하기가 어려웠다. 특히 테스트 참가자 모두 스마트 폰에 익숙하지 않아서 굽는 동작을 하는데 생각보다 시간이 들었고, 자판 배치가 익숙하지 않아서 기존 쿼터 자판에 없는 자소들을 찾을 때 가장 많은 시간이 걸렸다. 가장 차이가 날 거라 예상했던 받침 부분에서는 쌍자음에서는 cube 방식이 약간 더 빠른 경우도 있고, 그 외의 경우에는 익숙해진 경우 1~3초의 성능 향상이 있는 정도였다. GOMS 모델에 따른 시간 예측에서는 cube 방식과 cube-4B 방식이 비슷한 성능을 보여주었다. 예측 시간에서는 어느 한 방식이 화면을 한 번 이상 더 전환할 때 다른 방식보다 성능이 떨어졌다. 실제 시간에서는 예측 시간과는 다르게 전체적으로 cube-4B 방식이 나은 성능을 보여줬다. 다만 일부 cube 방식이 앞선 부분이 있는데, 특정 자소들이 cube 방식보다 cube-4B 방식이 잘 인지되지 않아서였다. 이 부분은 해당 자소들이 좀 더 인지되기 쉽도록 연구를 해보아야 할 부분이다.

측정에 참가한 사용자들이 지적한 부분은 복모음 입력시 완성된 입력만 지원하지 말고 모음끼리의 조합입력도 지원하는게 낫다는 점이 있었고 ' | ', 'ㅡ' 모음이 양 쪽 모음에 다 등장하는게 낫다라는 의견도 있었다. 이런 부분들은 향후 입력기를 개선할 때 참고할 예정이다.

제 6 장 결론

본 논문에서는 터치 스크린 상에서 사용가능한 5가지 배열을 제시하였다.

cube-3A 방식의 배열은 기존 cube 방식에서 겹모음들과 일부 겹자음들을 배치하여 cube 방식에 비해 평균 입력 타수를 줄어들었다.

cube-3B 방식의 배열은 굵는 상하 또는 좌우 굵는 방향에 따라 자음과 모음을 재배치하였고 cube-3A 방식에 비해 굵는 방향에 대한 일관성이 확보되었다.

cube-3C 방식 배열은 모음조화에 따라 양성 모음과 음성 모음을 나눠 배치하였고 cube-3B 방식에 비해 모음 인지도가 향상되었다.

cube-4A 방식 배열은 cube-3B 정책에 기반하여 배치가능한 버튼 수를 늘려서 배치하였고 cube-3B 방식에 비해 자음과 모음 분리라는 일관성이 확보되었다.

cube-4B 방식 배열은 cube-3C 정책에 기반하여 cube-4A 방식에 비해 모음 인지도가 향상되었다.

성능 분석을 위해 안드로이드 기기를 기반으로 하여 한글 입력기를 구현하여 기존의 cube 방식과 cube-4B 방식을 시험하였다. 시험 결과 평균 입력 타수, Hick's Law에 따른 인지 시간 측정, GOMS 모델에 기반한 각 입력 동작에 대한 수준별 입력 시간 예상 및 측정 부분에서 cube 방식에 비해서 cube-4B 방식이 입력 성능이 앞섰다.

앞으로 좀 더 연구하여야 할 부분은 다음이다. 첫째, 한글 입력에 관한 것만 집중되어서 실제 생활에 필요한 숫자, 영문자, 특수 문자 등은 전혀 고려되지 못했다는 점이다. 둘째, 보다 다양한 사용자들을 대상으로 하는 실험을 통해 사용 유형별로 분석이 필요하다는 점이다. 마지막으로 사용자가 지적한 모음 조합을 허용하는 점이나 'ㅣ', 'ㅡ' 모음들의 중복 배

치하는 부분들에 대한 분석도 필요하다. 이러한 부분들을 연구 개선해 나간다면 실제 스마트폰에서 널리 쓰일 수 있는 입력기를 완성할 수 있을 것이다.

참고 문헌

- [1] 위키 백과사전, 스마트폰,
<http://ko.wikipedia.org/wiki/%EC%8A%A4%EB%A7%88%ED%8A%B8%ED%8F%B0>, 2010.01.24
- [2] 지디넷, 구글이 넥서스 원을 출시한 이유,
<http://zdnet.co.kr/Contents/2010/01/26/zdnet20100126084451.htm>,
2010.01.26
- [3] 모비즌, 기존 전화 및 스마트폰 traffic 비용 통계,
<http://mobizen.pe.kr/868>, 2009.11.12
- [4] 로아그룹코리아, 2010 통신 시장 전망 보고서, 2010.01.08
- [5] 스마트폰 최고의 한글 입력기는 <http://jwmx.tistory.com/1709>,
2010.01.04
- [6] 서종화, Touch UI에 적합한 한글 입력기 설계 및 구현 큐브입력기
(Design and Implementation of Applicable Hangul Input Method for
Touch-UI cube Input Method), 석사학위논문, 숭실대학교, 2009.08
- [7] 오세청, 사용자 편의성을 고려한 이동 단말기에 대한 고찰, 석사학위
논문, 한양대학교, 2008.02
- [8] 박재화, 전화기 숫자 자판을 이용한 대화형 한글 문자 입력 방법, 정
보처리학회논문지 B 제14-B권 제5호 통권 제115호, pp.391-400, 2007.05
- [9] 위키 백과사전, 휴대 전화 문자 입력 방식,
http://ko.wikipedia.org/wiki/%ED%9C%B4%EB%8C%80_%EC%A0%84%ED%99%94_%EB%AC%B8%EC%9E%90_%EC%9E%85%EB%A0%A5_%EB%B0%A9%EC%8B%9D, 2010.03.06
- [10] 이진영, 모바일 기반의 다국어 입력 시스템, 정보처리학회지 Vol.15,

No.4, pp.65-72, 2008.07

[11] 국립국어 연구원, 민원 - 모음조화 관련,

http://www.korean.go.kr/09_new/minwon/qna_view.jsp?idx=54408,

2009.08.26

[12] 안드로이드 sdk, <http://developer.android.com/sdk/>, 2010.12.20

[13] 이클립스용 안드로이드 플러그인,

<https://dl-ssl.google.com/android/eclipse/>, 2010.09.01

[14] 위키 백과사전, 오토마타 이론,

http://ko.wikipedia.org/wiki/%EC%98%A4%ED%86%A0%EB%A7%88%ED%83%80_%EC%9D%B4%EB%A1%A0, 2009.06.29

[15] 칸드로이드, 칸드로이드 두벌식 한글 입력기

<http://www.kandroid.org/board/board.php?board=sourcecode&search=%ED%95%9C%EA%B8%80&shwhere=subject&command=body&no=4>,

2008.05.19

[16] 기도형, 실물 이동 전화를 이용한 한글 입력 방식의 수행도 및 선호도 평가, 대한인간공학회 저널 Vol.25, No.3 pp.33-41 2006.8

[17] 위키 백과사전, Hick's law,

http://en.wikipedia.org/wiki/Hick%27s_law, 2010.08.09

[18] 위키 백과사전, KLM-GOMS,

<http://en.wikipedia.org/wiki/KLM-GOMS>, 2010.06.17

부록

한글 자소 출현 빈도표

표 부-1 한글 자소 사용 빈도

no.	자소	구분	백분율	no.	자소	구분	백분율	no.	자소	구분	백분율
1	ㅇ	초성	74.02	1	ㅏ	중성	67.7	1	ㄴ	종성	46.68
2	ㄱ	초성	41.16	2	ㅣ	중성	47.65	2	ㄹ	종성	27.31
3	ㄷ	초성	28.15	3	ㅡ	중성	40.49	3	ㅇ	종성	21.69
4	ㅅ	초성	26.59	4	ㅑ	중성	33.14	4	ㄱ	종성	14.46
5	ㅈ	초성	26.31	5	ㅓ	중성	30.3	5	ㅓ	종성	9.08
6	ㅎ	초성	22.33	6	ㅕ	중성	20.82	6	ㅕ	종성	7.7
7	ㄴ	초성	21.45	7	ㅗ	중성	14.85	7	ㅗ	종성	4.7
8	ㄴ	초성	20.98	8	ㅜ	중성	14.4	8	ㅜ	종성	4.02
9	ㅓ	초성	15.1	9	ㅟ	중성	13.83	9	ㅟ	종성	1.06
10	ㅗ	초성	12.62	10	ㅑ	중성	6.32	10	ㅎ	종성	0.94
11	ㅕ	초성	7.04	11	ㅓ	중성	5.71	11	ㅓ	종성	0.9
12	ㅓ	초성	3.75	12	ㅑ	중성	3.5	12	ㅓ	종성	0.89
13	ㅓ	초성	3.47	13	ㅓ	중성	3.32	13	ㅓ	종성	0.65
14	ㅓ	초성	2.55	14	ㅓ	중성	2.24	14	ㅓ	종성	0.6
15	ㅓ	초성	2.28	15	ㅓ	중성	2.01	15	ㅓ	종성	0.59
16	ㅓ	초성	1.69	16	ㅓ	중성	1.67	16	ㅓ	종성	0.34
17	ㅓ	초성	1.02	17	ㅓ	중성	1.65	17	ㅓ	종성	0.25
18	ㅓ	초성	0.71	18	ㅓ	중성	1.56	18	ㅓ	종성	0.21
19	ㅓ	초성	0.56	19	ㅓ	중성	0.39	19	ㅓ	종성	0.15
				20	ㅓ	중성	0.13	20	ㅓ	종성	0.1
				21	ㅓ	중성	0.07	21	ㅓ	종성	0.09
								22	ㅓ	종성	0.07
								23	ㅓ	종성	0.01
								24	ㅓ	종성	0.01
								25	ㅓ	종성	0.0046
								26	ㅓ	종성	0.0027
								27	ㅓ	종성	0.0001