

<http://dx.doi.org/10.7236/IIBC.2016.16.5.157>

IIBC 2016-5-24

비디오 카메라를 이용한 핑거 키보드의 설계 및 구현

Design and Implementation of Finger Keyboard with Video Camera

황기태*

Kitae Hwang*

요약 본 논문은 종이 위에 그려진 키보드 상에 카메라를 이용하여 사용자의 키 입력을 탐지하는 핑거 키보드를 제안한다. 본 논문에서는 핑거 키보드를 윈도우 운영체제와 모바일 단말기에서 사용할 수 있도록 구현하였다. 핑거 키보드의 소프트웨어는 표준 C/C++로 작성되어 포팅이 쉽다. 윈도우를 탑재한 PC에서는 보급형 USB 웹 카메라를 장착하고 핑거 키보드 소프트웨어를 애플리케이션 형태로 구현하여 탐지한 키를 운영체제의 메시지 큐에 삽입하였다. 그리고 전용 카메라와 리눅스를 장착한 Raspberry Pi 2 임베디드 컴퓨터에 핑거 키보드 소프트웨어를 구동시키고 블루투스를 이용하여 모바일 단말기의 키보드로 구축하였다. 성능 평가 결과 인식성공률은 분당 120타 수준에서 약 80% 정도로 측정되었다.

Abstract This paper presents Finger Keyboard which detects the user's key types on a keyboard drawn on the paper using a video camera. The Finger Keyboard software was written in standard C/C++ language and thus easy to port to other computing environments. We installed a popular USB-type web camera on a Windows PC and implemented the Finger Keyboard as a Windows application which detects key typing and then injects the key code into the message queue of the Windows operating system. Also we implemented the Finger Keyboard on the Raspberry Pi 2 embedded computer with a dedicated camera and connected it to the Android device as an external keyboard through the Bluetooth. The result of experiments showed that the average ratio of recognition success is around 80% at the typing speed of 120 characters per minute.

Key Words : Keyboard, Paper keyboard, User Interface

1. 서론

컴퓨터가 생긴 이래 하드웨어 키보드는 오랫동안 컴퓨터의 주 입력기로서 사용되어 왔다. 하지만, 최근 들어 하드웨어 키보드를 사용하기 어려운 모바일 단말기의 출현으로 LCD 터치 스크린 위에 키보드를 구현한 소프트웨어 키보드가 등장하였고, 키 입력 대신 음성 입력이나 제스처 입력 등 다양한 입력 방식이 개발되어 사용되고 있다[1,2,3].

한편 하드웨어 키보드의 불편함을 해소하는 다양한 형태의 키보드가 꾸준히 연구 개발되어 왔다. 1992년 IBM은 키보드 이미지를 바닥에 투사하여 사용자의 손가락 움직임을 탐지하는 프로젝션 키보드(projection keyboard)를 처음으로 개발하였고, Canesta, Celluon 등에서 여러 종류의 프로젝션 키보드를 개발하였다[4,5,6,7]. 하지만 이들 키보드는 프로젝션 장치로 인해 가격이 비싸고 사용자에게 따르면 실제 분당 100타 수준으로 인식률이 낮고, 프로젝트의 좁은 투사, 장애물이 없고 표

*정회원, 한성대학교 컴퓨터공학부(교신저자)
접수일자 : 2016년 8월 2일, 수정완료 : 2016년 9월 5일
게재확정일자 : 2016년 10월 7일

Received: 2 August, 2016 / Revised: 5 September, 2016 /
Accepted: 7 October, 2016

*Corresponding Author: calafk@hansung.ac.kr
Dept. of Computer Engineering, Hansung University, Korea

면이 평평하고 불투명해야 하는 투사 장소 등 범용으로 사용되기 어려운 여러 단점을 가지고 있다.

한편 애플은 최근에 종이 키보드(Paper Keyboard)를 개발하였다[8]. 종이 키보드는 키보드가 그려진 PDF 파일을 종이에 출력하고, 출력된 이 종이 위의 지정한 위치에 아이폰 단말기를 세워두고 애플의 앱 스토어에서 다운로드 받은 애플리케이션과 아이폰의 카메라를 이용하여 사용자의 키 입력을 인식하는 시스템이다. 하지만 애플의 종이 키보드는 많은 키가 지원되지 않고, 아이폰을 세워두는 등 구조적인 한계를 가지고 있기 때문에 실용적으로 사용되기에는 여전히 극복해야 하는 문제들이 있다.

본 논문은 애플의 종이 키보드와 유사하게 종이 위에 그려진 핑거 키보드(Finger Keyboard)를 제안한다. 핑거 키보드는 그림 1과 같이 보급형 웹캠 수준의 비디오 카메라를 이용하여 종이에 그려진 키보드 위에 손가락으로 타이핑하는 장면을 인식하여 사용자의 키 입력을 받아들이는 방식이다. 핑거 키보드는 여러 형식의 키보드(QWERTY, 천지인)를 지원하며, 종이와 카메라 위치가 자유롭고, 운영체제와 무관하여 윈도우 PC나 어떤 모바일 단말기에서도 사용할 수 있는 순수 소프트웨어 형태로 구현한다.



그림 1. 핑거 키보드
Fig. 1. Finger Keyboard

본 논문은 2장에서 핑거 키보드를 설계하고 3장에서 구현한 내용을 소개하며, 4장에서는 성능 평가의 결과를 보이고, 5장에서 결론을 맺는다.

II. 핑거 키보드의 설계

1. 전체 시스템 설계

핑거 키보드(Finger Keyboard)는 그림 2와 같이 키보드가 그려진 종이와 웹캠과 같은 보급형 소형 카메라와

종이 키보드 위에서 손가락의 움직임을 판단하여 어떤 키가 눌리었는지 판단하는 소프트웨어로 구성된다.



그림 2. 키보드가 그려진 종이
Fig. 2. Keyboard drawn on the paper

이 소프트웨어는 3 계층으로 구성되며, 핑거 키보드의 구조는 그림 3과 같다.

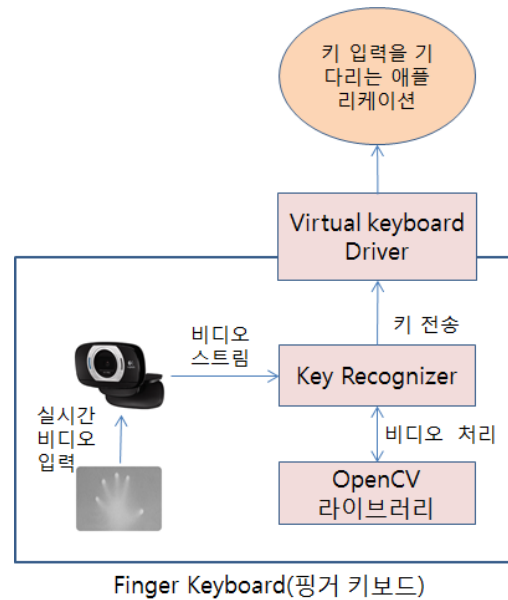


그림 3. 핑거 키보드 설계
Fig. 3. Design of Finger Keyboard

키 인식기(Key Recognizer) 모듈은 종이 키보드 위에서 손가락의 움직임을 탐지하여 어떤 키가 눌리었는지를 인식하는 모듈이다. 이 모듈은 카메라의 실시간 영상을 처리하는 OpenCV 라이브러리를 활용한다[9]. OpenCV는 컴퓨터 비전이나 이미지 프로세싱 애플리케이션의 개

발에 사용되는 오픈 소스 프로젝트로서 현재 많은 비디오 처리 연구나 상용 애플리케이션 제작에 이용되고 있다. 본 연구에서는 다양한 운영체제에서 구동되는 환경을 고려하여 표준 C버전의 OpenCV 라이브러리를 이용하였다.

컴퓨터 시스템에서 입력된 키를 애플리케이션에게 공급한 것은 운영체제의 기능이다. 그러므로 핑거 키보드는 입력된 키를 운영체제에 공급하고 운영체제는 다시 키 입력을 기다리는 애플리케이션에 공급한다. 가상 키보드 드라이버(Virtual Keyboard Driver) 모듈은 맨 위의 소프트웨어 계층으로서 키 인식기 모듈이 탐지한 키를 운영체제로 전달한다. 운영체제에 따라 키보드로부터 입력된 키를 전달받는 방법은 약간씩 다르므로 가상 키보드 드라이버는 핑거 키보드가 구동되는 컴퓨터 시스템에 따라 조금씩 다르게 작성되어야 한다.

2. 키 인식기(Key Recognizer) 모듈 설계

가. 키 인식 과정

키 인식기에 의해 처리되는 키 인식 과정은 그림 4와 같다. 키 인식 과정은 선 처리(Pre-processing)와 본 처리(Main-processing)의 두 과정으로 나뉜다. 선 처리는 사용자의 키 입력을 받기 전에 필요한 준비의 과정이며, 본 처리는 키를 입력하는 손가락의 움직임을 인식하여 어떤 키인지, 키를 눌렀는지, 떴는지 등을 판단하여 가상 키보드 드라이버에 전달하는 과정이다.

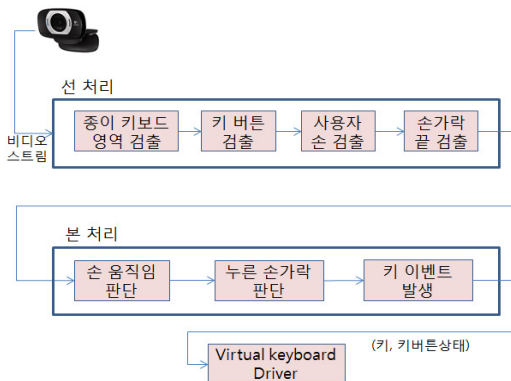


그림 4. 키 인식 과정
 Fig. 4. Process to recognize key typing

나. 선처리 과정

1) 종이 키보드 영역 검출

사용자의 키 타이핑을 인식하기 위해서는 먼저 종이 키보드의 영역을 인식한다. 사용자가 그림 5와 같이 마우스로 종이 키보드 영역을 지정하면, 프로그램은 그림 5에서 빨간 점으로 보이는 종이 키보드의 네 모퉁이를 탐지한다.



그림 5. 종이 키보드 영역 검출
 Fig. 5. Detection of the Area of Paper Keyboard

종이 키보드는 흰색과 검은색으로 이루어져 있어서 HSV 컬러모델로 변경하고 채도를 추출한 다음 꼭지 점을 탐색하면 된다.

2) 키 버튼 검출

이 과정은 종이 키보드 위에 그려진 키 버튼의 위치를 검출하는 과정이다. 그림 5에서 종이 키보드가 사다리꼴 형태로 보이는 것은 비디오 카메라의 원근에 의한 현상이다. 우선 사다리꼴 형태로 찌그러진 영상에 원근 변환을 실행하여 그림 6과 같이 종이 키보드를 직사각형 형태로 변환한다.



그림 6. 직사각형으로 변환된 종이 키보드 이미지
 Fig. 6. Paper Keyboard Image converted to Rectangle

이 이미지에서 80개의 키 버튼 영역을 모두 인식하고 각 키 버튼마다 고유한 ID 번호를 부여한다. 키 버튼 영역을 정확히 검출하기 위해 키 버튼 내부에 있는 키 레이블 문자들을 지워야 한다. 이를 위해 키보드의 배경 이미지를 검은색으로 채우고, 종이키보드의 이진 이미지와 XOR 연산을 수행하여 그림 7과 같이 레이블 문자를 담고 있는 키 버튼을 모두 검은색으로 만든다.

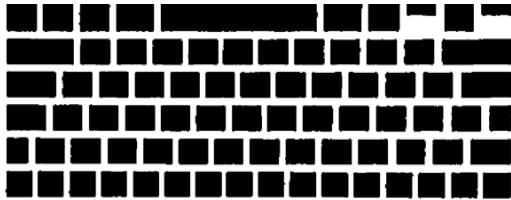


그림 7. 키 버튼의 배경색을 검은색으로 채우기
Fig. 7. Filling black color in all key buttons

그리고 꼭지 점을 찾는 알고리즘으로 320개의 점을 추출하여 그림 8과 같이 키 버튼 영역을 찾아낸다.



그림 8. 검출된 키 버튼의 꼭지점
Fig. 8. Corners of Key Buttons detected

80개의 각 키에 대해 그림 9와 같이 영역을 표시하는 값 (x, y, width, height)를 기억해둔다.

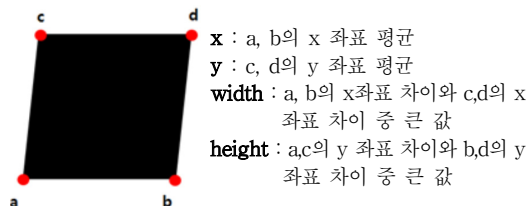


그림 9. 검출된 키 버튼의 꼭지 점
Fig. 9. Corners of Key Buttons detected

3) 사용자의 손 검출

종이 키보드와 사용자의 손을 구분하기 위해 사용자의 손을 검출한다. 우선 마우스로 사용자의 손 부분을 클릭하여 사용자 손의 피부색을 인식하고 피부색에 대한 히스토그램을 생성한다. 피부색에 대한 히스토그램을 기반으로 키보드를 포함하는 사용자 손의 역투영한 이미지를 만든다. 이 역투영 이미지와 손이 올라가기 전 종이 키보드 이미지의 차이를 구해 손 이미지를 검출한다. 손을 검출하는 것은 손가락 부분을 찾기 위함이다.

4) 손가락 끝 검출

손에 대한 외곽선을 따 내고, 손가락 끝을 검출한다. 그림 10은 핑거 키보드가 손가락 끝을 검출하여 색으로 표시한 결과이다.

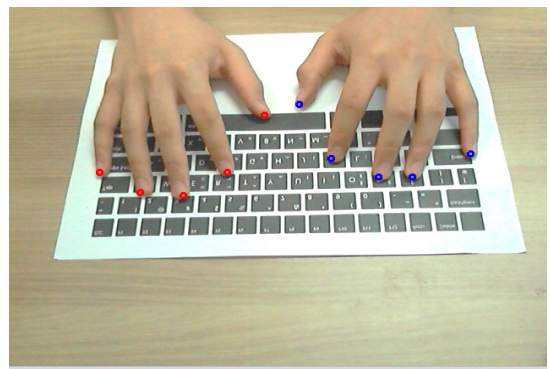


그림 10. 손가락 끝 검출
Fig. 10. Detection of the Tips of All Fingers

다. 후처리 과정

후처리 과정은 사용자가 타이핑하는 동안 실시간으로 손가락 움직임을 탐지하여 어떤 키가 눌러 졌는지 판단하는 과정이다.

1) 손 움직임 판단

손의 움직임에 따라 손가락이 같이 움직인다. 그러므로 타이핑이 일어나지 않았음에도 손의 움직임만으로 키가 눌러졌다고 판단할 수 있기 때문에, 손의 움직임을 검출하여 이를 제외시켜 주어야 한다.

2) 누른 손가락 판단

손가락의 움직임을 모니터링한다. 하지만 하나의 손가락으로 키를 누를 때 다른 손가락도 함께 움직이기 때문

에 한 번에 반드시 하나의 키 타이핑만 가정한다. 그러므로 가장 많이 움직인 손가락이 키를 눌렀다고 판단하며 손가락 끝점에 있는 키가 눌려졌다고 판단한다.

3) 키 이벤트 발생

가장 많이 움직인 손가락을 결정하였으면, 그 손가락의 끝점에 있는 키가 눌려졌다고 판단하고, 본 연구에서 정한 키 이벤트를 발생시킨다. 키 이벤트는 표 1과 같이 4 개 중 하나의 상태 정보와 키 버튼 ID 값을 전달한다. 키 이벤트 발생을 기다리는 가상 키보드 드라이버에게 전달되며 가상 키보드 드라이버는 키 코드를 운영체제로 전달한다.

표 1. 키 이벤트
 Table 1. Key Events

상태	설명
KEY_STATE_NONE	키 버튼의 초기 상태
KEY_STATE_PRESS	키 버튼이 눌린 상태
KEY_STATE_PRESS_HOLD	키 버튼이 계속 눌러지고 있는 상태
KEY_STATE_RELEASE	키 버튼이 떴진 상태

3. 가상 키보드 드라이버 (Virtual Keyboard Driver)

가상 키보드 드라이버는 운영체제와 핑거 키보드를 연결하는 소프트웨어 모듈로서 운영체제와 컴퓨터의 하드 플랫폼에 따라 모양을 달리한다. 하지만 가상 키보드 드라이버와 키 인식기 모듈 사이에는 그림 11과 같은 키 이벤트 전달 구조로 설계된다.

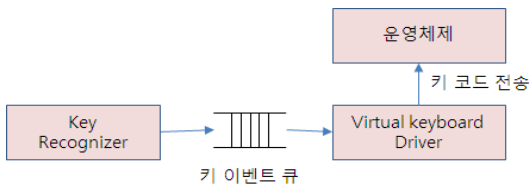


그림 11. 키 이벤트 큐
 Fig. 11. Key Event Queue

III. 핑거 키보드의 구현

본 연구에서는 윈도우 운영체제를 설치한 PC와 블루

투스 기능을 갖춘 모바일 단말기에서 키 입력이 가능하도록 두 시스템에 대해 각각 핑거 키보드를 구현하였다.

1. 윈도우 PC

핑거 키보드에서 사용하는 카메라는 Logitech C270, 1280x720에 300만화소를 지원하는 보급형 USB 웹 캠을 사용하고 핑거 키보드 소프트웨어는 윈도우 응용프로그램으로 구현하였다. 그림 12는 카메라를 구동시키고 사용자의 손가락을 탐지하고 그 움직임을 보여주며 ‘안녕하세요 잘부탁드립니다’라고 메모장에 입력하는 핑거 키보드의 작동 사례를 보여준다.

핑거 키보드의 가상 키보드 드라이버 모듈은 윈도우 C 라이브러리의 keybd_event() 함수를 호출하여 탐지한 키를 운영체제에게 전달하면, keybd_event() 함수는 윈도우 운영체제의 시스템 메시지 큐에 삽입하고, 메시지를 처리하는 운영체제의 루틴이 현재 포커스를 가진 애플리케이션에게 키를 공급한다.

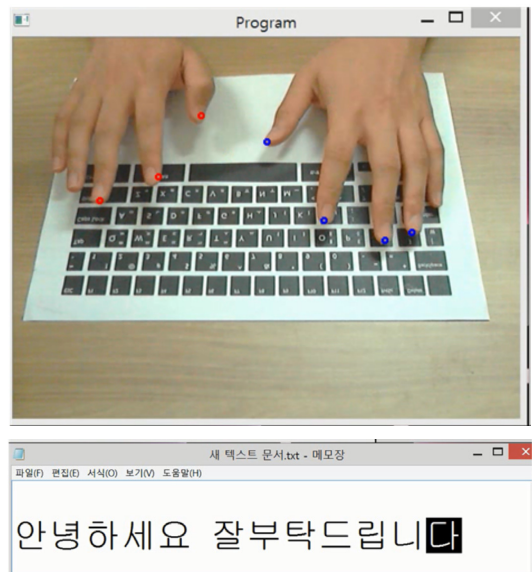


그림 12. 윈도우에서 작동하는 핑거 키보드
 Fig. 12. Finger Keyboard Operating in Windows

2. 모바일 단말기

본 연구 팀은 핑거 키보드를 스마트폰과 같은 모바일 단말기의 키보드로 사용할 수 있도록, 리눅스 운영체제를 탑재한 Raspberry Pi 2 임베디드 컴퓨터에 핑거 키보드의 소프트웨어를 구현하였다[10]. 500만 화소에

640x480의 해상도를 가진 Raspberry Pi 2 임베디드 컴퓨터 전용 카메라 모듈을 사용하였으며 구성은 그림 13과 같다. 핑거 키보드 소프트웨어는 표준 C/C++로 작성되었기 때문에 윈도우나 리눅스에서 모두 문제없이 컴파일된다.

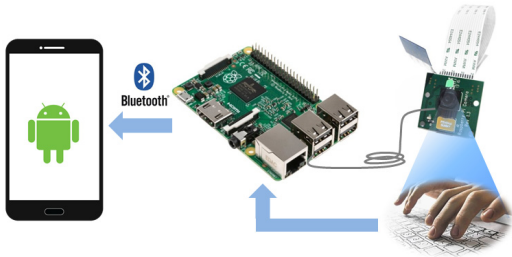


그림 13. Raspberry Pi 2의 핑거 키보드
Fig. 13. Architecture of Finger Keyboard in Raspberry Pi 2

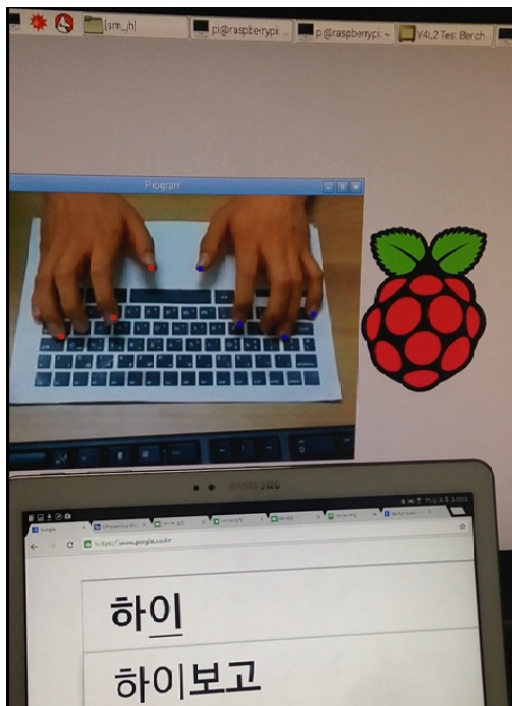


그림 14. Raspberry Pi 2의 핑거 키보드 사용
Fig. 14. Key Typing with Finger Keyboard in Raspberry Pi 2

그림 14는 Raspberry Pi 2에 연결된 카메라를 통해 키 타이핑을 인식하고 안드로이드를 탑재한 삼성 갤럭시 탭에 키 타이핑이 이루어지고 있는 모습을 보여준다.

IV. 성능 평가

핑거 키보드의 성능을 평가하기 위해 키 인식성공률을 측정하였다. 키 인식성공률은 핑거 키보드에서 타이핑 횟수에 대해 키 인식이 성공한 비율을 뜻하며 다음 식 (1)과 같이 정의된다.

$$\text{인식성공률} = (\text{인식성공 횟수} / \text{타이핑 횟수}) \times 100 \quad (1)$$

본 논문에서는 분당 타자수를 높여 가면서 식 (1)을 측정한 결과 그림 15와 같은 결과를 얻었다. 현재 초등학교의 평균타자 속도가 분당 120타 정도라고 알려져 있으며, 실험 결과 핑거 키보드의 성능이 초등학교 수준에 머무르고 있다는 것을 보여준다.

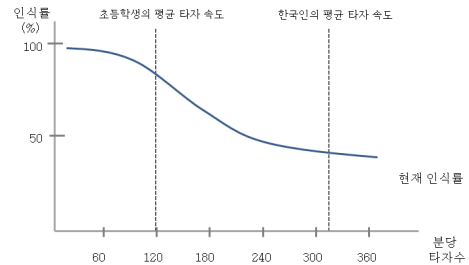


그림 15. 인식 성공률
Fig. 15. Recognition Success Ratio

본 연구팀은 실험 데이터를 분석하여 인식 성공률이 낮은 이유를 다음과 같이 정리하였다. 첫째, 손가락의 움직임은 3차원 수준에서 파악되어야 하는데 카메라 1대를 사용하면 2차원 영상에서 인식하므로 손가락의 움직임에 대한 정확한 탐지가 어렵다. 또한 같은 이유로 타이핑이 이루어지는 동안 손 전체가 함께 움직이므로 엉뚱한 손가락의 움직임을 키 타이핑으로 잘못 인식하는 경우가 많다.

둘째, 손가락의 움직임은 여러 프레임을 분석하여야 정확히 파악되지만, Raspberry Pi의 CPU 처리 속도가 낮아 손가락의 움직임을 탐지하는데 필요한 비디오 프레임 비율(FPS, Frames Per Second)이 낮기 때문이다. 추후 비디오 프레임의 비율을 높이기 위해 비디오의 크기를 줄이는 등 다양한 시도가 필요하다.

본 연구팀은 본 연구를 통해 핑거 키보드를 윈도우 PC나 모바일 단말기 등 다양한 환경에서 사용할 수 있는

구현 가능성을 확인하였다. 추후 연구를 통해 핑거 키보드의 낮은 성능 원인을 보다 면밀히 평가하고, 3차원으로 손가락의 움직임을 탐지하기 위해 2 대의 카메라를 사용하는 등 인식 성공률을 높이는 다양한 연구를 계속할 예정이다.

V. 결 론

하드웨어 기술과 멀티미디어 기술의 발전으로 하드웨어 키보드에서 벗어나 음성 인식, 모션 인식 등을 활용하는 입력 인터페이스에 대한 연구가 활발히 이루어지고 있다. 하지만 여전히 키보드는 텍스트를 입력하는 쉽고도 확실한 사용자 인터페이스이다. 본 논문에서는 키보드가 그려진 종이 위에 사용자의 손가락 움직임을 탐지하여 카메라로 키 입력을 인식하는 핑거 키보드를 설계 구현하였다.

구체적으로 핑거 키보드를 보급형 USB 웹 카메라를 갖춘 윈도우 PC에서 애플리케이션 형태로 구현하였으며, 이를 간단히 수정하여 전용 카메라와 리눅스를 탑재한 Raspberry Pi 2 임베디드 컴퓨터에 설치 구동하고 블루투스를 이용하여 안드로이드를 탑재한 삼성 갤럭시 탭의 키보드로 작동시켰다.

실험을 통해 성능을 평가한 결과 초등학생의 평균 타자 수준인 분당 120타로 입력하는 경우 80% 수준의 인식 성공률을 보였다. 인식률이 높은 편은 아니지만 핑거 키보드의 활용 가능성이 검증되었다. 추후 인식 실패의 원인을 보다 면밀히 분석하고 2 개의 카메라를 사용하는 등 개선을 위한 다양한 연구를 수행할 계획이다.

References

[1] Kitae Hwang, Jae Moon Lee, "An Implementation Experience of Accelerometer-based Gesture Recognition with Android Smartphone", International Journal of Advancements in Computing Technology(IJACT), Vol. 5, No. 12, August 2013

[2] Kitae Hwang, Jae Moon Lee, "Preliminary Study on Soft Keyboard with Recommendation for Mobile Device", The Journal of The Institute of Internet, Broadcasting and Communication, vol. 13, no. 6, pp.137-145, 2013.

[3] Kitae Hwang, Tae Wan Kim, Hye Kyung Cho, "Performance Evaluation of Missless Soft Keyboard with Recommendation", The Journal of The Institute of Internet, Broadcasting and Communication, vol. 14, no. 4, pp.187-195, 2014.

[4] https://en.wikipedia.org/wiki/Projection_keyboard#cite_note-4

[5] <http://www.laserkeyboard.com/>

[6] http://news.bbc.co.uk/2/hi/in_depth/sci_tech/2000/dot_life/2326077.stm

[7] http://www.celluon.com/products_epic_overview.php

[8] <https://itunes.apple.com/kr/app/paper-keyboard-fast-typing/id715319520?mt=8>

[9] <http://opencv.org/>

[10] <https://www.raspberrypi.org>

저자 소개

황 기 태(정회원)



- 서울대학교 컴퓨터공학과 학사
- 서울대학교 컴퓨터공학과 석사
- 서울대학교 컴퓨터공학과 박사
- 경력 : University of Callifornia, Irvine 방문교수
- University of Florida 방문 교수
- <주관심분야 : 모바일 시스템>

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음