Journal of Korea Multimedia Society Vol. 23, No. 12, December 2020(pp. 1454-1463)
https://doi.org/10.9717/kmms.2020.23.12.1454

입 벌림 인식과 팝업 메뉴를 이용한 시선추적 마우스 시스템 성능 개선

변주영†, 정기철††

Improving Eye-gaze Mouse System Using Mouth Open Detection and Pop Up Menu

Ju Yeong Byeon†, Keechul Jung††

ABSTRACT

An important factor in eye-tracking PC interface for general paralyzed patients is the implementation of the mouse interface, for manipulating the GUI. With a successfully implemented mouse interface, users can generate mouse events exactly at the point of their choosing. However, it is difficult to define this interaction in the eye-tracking interface. This problem has been defined as the Midas touch problem and has been a major focus of eye-tracking research. There have been many attempts to solve this problem using blink, voice input, etc. However, it was not suitable for general paralyzed patients because some of them cannot wink or speak. In this paper, we propose a mouth-pop-up, eye-tracking mouse interface that solves the Midas touch problem as well as becoming a suitable interface for general paralyzed patients using a common RGB camera. The interface presented in this paper implements a mouse interface that detects the opening and closing of the mouth to activate a pop-up menu that the user can select the mouse event. After implementation, a performance experiment was conducted. As a result, we found that the number of malfunctions and the time to perform tasks were reduced compared to the existing method.

Key words: HCI, Interface, Eye Gaze, Landmark Detection, Mouse, Midas Touch Problem, Assistive Technology, Spinal Cord Injury.

1. 서 론

마우스, 키보드와 같은 일반적인 입력 수단은 팔, 손 등을 움직일 수 없는 전신 마비 환자에게 적합하지 않다[1]. 이 때문에 시선 추적 인터페이스가 그 대안 중 하나로 활발히 연구되고 있다. 전신 마비 환자를 위한 시선 추적 PC 조작 인터페이스를 구현하는데 중요한 요소는 GUI를 조작하기 위한 마우스 인터페이스를 구현하는 것인데, 이 인터페이스에서 구현해야 기능은 두 가지로 나눌 수 있다. 첫 번째는 사용자가 원하는 화면상의 좌표에 정확히 커서를 이동시키는 커서 이동이다. 두 번째는 사용자가 원하는 시점에 원하는 좌 클릭, 우 클릭, 더블 클릭, 스크롤 등의 이벤트 발생 기능이다. 이벤트 발생 기능을 이용할 땐, 사지 마비 환자가 원하는 시점에, 원하는 이벤트를 정확히 발생시킬 수 있어야 한다. 하지만 시선 추적 인터페이스 연구 분야에서 기존에 제시된 dwell time, blink 등의 방식들은 의도하지 않은 입력

※ Corresponding Author : Keechul Jung, Address: (06978) Sangdo-ro 369, Dongjak-gu, Seoul, Korea, TEL : +82-2-828-7260, E-mail : kcjung@ssu.ac.kr
Receipt date : Aug. 23, 2020, Revision date : Oct. 19, 2020
Approval date : Nov. 10, 2020

† Dept. of The Global School of Media, College of Information Technology, Soongsil University (E-mail : wndudqus123@gmail.com)
†† Dept. of The Global School of Media, College of Information Technology, Soongsil University

이 빈번히 발생하는 문제가 있다. 이를 시선 추적 인터페이스 분야에서는 Midas touch problem이라고 정의하고 계속해서 이를 개선하려는 연구들이 진행되고 있다[2].

2017년 X. Zhang 외 3인은 웹상에서 사용할 수 있는 마우스 입력 방법으로 dwell time 방식과 popup menu 방식을 함께 이용한 이벤트 입력을 제안했다[3]. H. Singh와 J. Singh는 윙크를 이용한 이벤트 발생 방식을 제안했다[4]. 위 방식들은 기존의 dwell time, blink 이벤트 발생 방식에서 사용자가 의도적으로 해야 하는 행동을 추가함으로써 Midas touch problem을 해결하려고 했다. 하지만 여전히 입력 행동이 사용자의 무의식적 반응에 의해 영향을 받을 수 있어 사용자에게 불편함을 줄 수 있다는 한계가 있다. 또한, dwell time에 기반해 popup 메뉴를 함께 사용한 인터페이스는 메뉴가 popup 되는 곳에 사용자가 보고 싶은 콘텐츠가 있어 무의식적으로 응시할 경우 popup에 의해 시야를 가리게 되는 문제점이 있다. 윙크방식의 경우에도 윙크를 할 때 시야가 줄어든다는 단점이 있고, 신경이 아직 완전히 발달하지 못한 미성년자 중 일부는 한쪽 눈만을 깜빡이는 행위를 할 수 없는 경우도 있다[5]. 또한, 윙크에 1:1로 좌 클릭, 우 클릭 등을 매칭시킬 경우 입력할 수 있는 이벤트의 개수가 제한되어있다.

완전히 새로운 입력 수단을 이용해 문제를 해결하려는 연구들도 진행되었다. 2018년 S. Soundarajan, H. Cecotti 등은 클릭 이벤트를 발생시키기 위해 입에 물 수 있는 버튼을 이용하는 mouth switch 이벤트 발생 방식을 제안했다[6]. 2020년 박주현 외 2명은 음성 입력을 이용해 마우스의 좌, 우클릭 이벤트를 발생시킬 수 있는 eye-voice 방식을 제안했다[7]. 위의 논문들에서 제시한 방법들은 사용자가 의도적으로 발생시키는 행동만을 입력 방법으로 삼았기 때문에 오입력이 적다는 장점이 있다. 하지만 mouth switch의 경우는 입에 계속해서 물리적인 스위치를 물고 있어야 하기 때문에 위생상의 문제가 있을 수도 있고, 특수 제작된 구강 스위치를 따로 구매해야 해서 사용자의 경제적인 부담 또한 가중되는 문제점이 있다. 또한, 스위치 방식으로 구현했기 때문에 입력의 개수가 제한적일 수밖에 없다. 음성을 이용한 이벤트 발생 방식의 경우 음성을 사용할 수 없는 사지 마비 장애인이 있다는 문제점이 있다. 한 연구에 따르면 경수 마비 환자의 67%가 호흡기계 합병증으로 관리가 필요하다고 하며 이 중에서는 기관절개관 삽입을 받는 환자들이 다수 있는데, 이 기관절개관 삽입을 받고 커프를 사용할 경우 성대로 공기가 전달되지 않아 말을 할 수 없어 음성 입력 방식을 사용할 수 없다는 한계점이 있다[9].

이러한 선행 연구들의 한계점을 볼 때, 의도하지 않은 오입력인 Midas touch problem을 방지하면서, 전신 마비 환자에게 적합한 새로운 시선추적 마우스 인터페이스가 필요한 바이다. 본 연구에서는 기존 시선추적 인터페이스의 문제들을 해결한 새로운 Mouth-popup 인터페이스를 제시한다. 제시하는 시선추적 마우스 인터페이스는 시선추적기와 함께 일반 웹캠을 이용해 마우스 이벤트를 발생시킨다. 실시간 영상을 통해 입이 열렸음을 감지하고 마우스 이벤트를 선택할 수 있는 팝업 창을 보여준 뒤 시선을 통해 선택된 이벤트를 발생시키는 입력 방식으로, 입의 열림을 이용하기 때문에 무의식적인 마우스 이벤트 실행을 막을 수 있으며 시선과 입의 움직임만을 이용해 본 인터페이스 시스템의 모든 기능을 사용할 수 있어 목과 상, 하지가 모두 마비되었고, 호흡기계의 관리가 필요해 기관절개관을 삽입 받아 말을 할 수 없는 장애인들도 이용할 수 있다. 또한, 팝업 메뉴를 이용하기 때문에 마우스에서 필수적으로 필요한 좌 클릭, 우클릭, 더블 클릭, 스크롤 등의 기능을 모두 수행할 수 있는 방식이다. 본 연구는 새롭게 제시하는 Mouth-popup 시선추적 마우스 인터페이스를 구현하고, 이를 기존 방식들과 속도, 정확도 등을 비교하여 새롭게 제시한 방식이 시선추적 마우스 시스템의 정확성과 기능성을 향상시키는 것을 확인함을 목표로 한다.

2. Mouth-popup 시선추적 마우스 시스템 설계

2.1 시스템 목표

본 논문에서 제시하는 Mouth-popup 시선추적 마우스 시스템의 목표는 세 가지이다. 첫 번째, 기존의 눈 깜빡임, dwell time 등 무의식적인 행동이 아닌, 의식적인 행동인 입의 열림과 닫힘으로 이벤트를 선택하고 발생 시켜 Midas touch problem을 개선하는 것을 목표로 한다. 두 번째, 1번 경수에서 4번 경수(C1~C4) 중 일부가 완전히 손상되어 상, 하지와 목이 마비되었거나, 전신 마비 장애인 중 기관절개관

삽입에 의해 발성에 문제가 있는 장애인들 또한 도움 없이 마우스의 기능을 동일하게 이용할 수 있도록 하는 것을 목표로 한다. 세 번째, 시선추적기와 일반 RGB 카메라 외에는 특수한 장비를 이용하지 않는 것을 목표로 한다. 본 연구에서는 세 가지 목표 기준에 부합하는 시선추적 마우스 시스템을 설계하였고 이를 Mouth-popup 시선추적 마우스 인터페이스라 명명하였다.

2.2 기능 설계

Mouth-popup 시선추적 마우스 인터페이스가 마우스로서 기능하기 위해서는 크게 커서를 이동시키는 기능, 이동된 커서의 좌표에 특정 이벤트를 발생시키는 기능 등이 필요하다.

본 시스템에서 구현하는 기능은 크게 2가지가 있다. Fig. 1에서 볼 수 있는 기능과 같이 커서를 이동시키는 moveCursor 기능이 있고, 이벤트를 발생시키는 makeEvent 기능이 있다. 마우스를 정상적으로 이용하기 위해 makeEvent 기능에서 발생시켜야 할 이벤트는 세부적으로 좌 클릭, 우 클릭 더블클릭, 스크롤 등으로 네 가지가 있다. 본 연구에서 제시하는 마우스 시스템은 시선추적과 입의 동작만을 이용해 moveCursor 기능과 위에서 언급한 4가지 이벤트를 발생시킬 수 있는 makeEvent를 구현하는 것으로 기능을 설계하였다.

2.3 입력 방식 설계

본 마우스 시스템은 Midas touch problem을 해결함과 동시에 사지 마비 및 특수한 상태의 장애인들이 손이나 음성 등을 이용하지 않고 마우스의 기능을 사용할 수 있도록 하는 시스템이다. 따라서 이러한 사용자에게 적합한 인터페이스를 제작하기 위해서는 적절한 입력 방식의 설계가 중요하다. 입은 눈 등의 다른 얼굴 근육보다 의식적으로 제어하기 쉽다는 장점이 있고, 척수가 손상된 전신 마비 환자들도 비교적 다른 부위에 비해 자유롭게 움직일 수 있기 때문에 본 시스템에서는 커서 이동을 제외한 클릭, 스크롤 등의 마우스 이벤트 입력을 위한 행동으로 입의 열림과 닫힘을 이용했다. 또한, 입의 움직임 한 동작으로 여러 이벤트를 발생시킬 수 있게 하기 위해서 이벤트를 선택하기 위한 팝업창을 추가로 이용했다. 이에 기반해 Mouth-popup 시선추적 마우스 시스템의 입력 방식을 Fig. 2와 같이 설계했다.

Mouth-popup 입력 과정은 다음과 같다. 가장 먼저 눈으로 원하는 위치를 바라본다. 그리고 입을 벌려 이벤트를 발생시킬 수 있는 팝업 메뉴를 띄운다. 다시 눈으로 원하는 이벤트에 해당하는 버튼을 바라본 후 입을 닫음으로써 팝업이 뜨기 전에 저장된 커서 위치에 선택된 이벤트를 발생시킨다. 본 입력 방식은 전신 마비 환자들에게 적합한 입의 여닫음이라는 하나의 입력 행동과 팝업만으로 여러 마우스 이벤트를 실행할 수 있도록 하는 것에 초점을 맞추어 설계했다.

3. Mouth-popup 시선추적 마우스 시스템 구현

3.1 개발 환경 및 대상 시스템

본 Mouth-popup 시선추적 마우스 시스템은 Midas touch problem을 해결하고 전신 마비 장애인들에게 적합한 인터페이스를 구현해 전신 마비 장애인들이 도움 없이 PC 상의 마우스 기능을 사용할 수 있도록 하는 것이 목표이다. 때문에 본 논문의 구현에서는 대중적으로 사용하는 Windows 10 OS 상에서 Win-

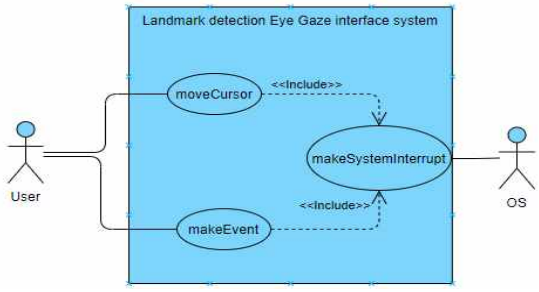

Fig. 1. Use case diagram for required feature that the mouse system must support.

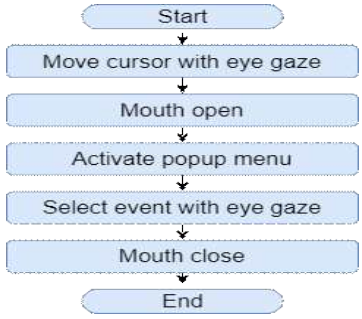

Fig. 2. Flow chart for interaction process diagram.

dows API를 이용해 마우스 시스템 개발을 진행했다. 또한, 시선추적을 위해 시선추적 장치, 이벤트 입력을 위해 일반 웹캠을 각각 사용했다. 개발언어는 본 연구에서 사용한 시선추적 하드웨어 제조사에서 제공하는 라이브러리와의 호환성을 고려하여 C++로 개발했다.

3.2 UI 구현

본 시스템에서 사용하는 UI는 두 가지 창이 있다. 첫 번째는 설정 및 모니터링 창이다. 해당 창은 open CV 영상처리 라이브러리의 GUI API를 사용해 구현하였다.

Mouth-popup 시선추적 마우스 시스템을 사용하는 동안 팝업 메뉴가 뜰 때를 제외하고 지속해서 Fig. 3의 설정 및 모니터링 창이 화면의 오른쪽 아래의 최상위에 부유해 있다. 이를 이용해 마우스 시스템이 입을 잘 인식하고 있는지 확인할 수 있으며 하단의 설정 UI를 이용해 커서 이동의 감도와 입 벌림 감지의 감도를 조절하고 확인 할 수 있도록 하였다. 이 버튼 또한 입 벌림을 통해 선택할 수 있어 사지 마비 장애인 스스로 설정할 수 있도록 구현하였다.

두 번째 UI는 발생시킬 이벤트를 선택할 수 있는 팝업 메뉴 창이다. 해당 창은 입을 벌릴 때 팝업되도록 구현하였다. 입 열림이 감지되면 설정 및 모니터링 UI 창이 일시적으로 꺼지고 팝업 메뉴가 사용자에게 보여진다.

Fig. 4의 팝업창이 뜬 후 시선이동을 통해 원하는 이벤트를 선택하고 입을 닫으면 팝업창이 꺼지고, 설정 및 모니터링 창이 켜짐과 동시에 해당 이벤트가

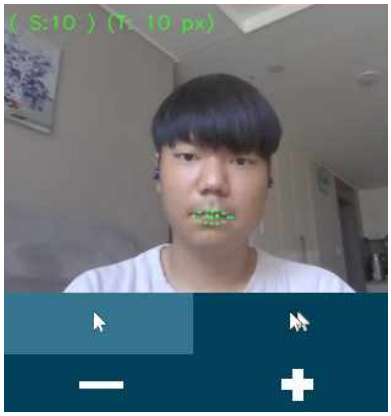

Fig. 3. Setup and monitoring UI window.

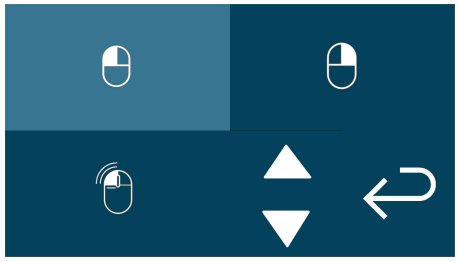

Fig. 4. Pop-up menu window.

미리 저장된 좌표에 발생하도록 구현했다. 각각의 버튼에는 설계 단계에서 정의한 makeEvent의 기능인 좌 클릭, 우 클릭, 더블클릭, 스크롤 등의 기능을 할당했다. 추가로 오른쪽 아래의 return 아이콘에 시선을 위치하고 입을 닫을 경우, 마우스 이벤트를 발생시키지 않고 팝업을 닫을 수 있도록 구현하여 실수로 팝업을 킨 경우에도 오입력이 발생하지 않도록 하였다. 팝업 메뉴는 Fig. 4와 같이 버튼 크기를 최대화하여 전체화면으로 구현했다. 이는 빠르게 직관적으로 이벤트를 선택할 수 있도록 하기 위함이다.

두 창은 모두 마우스 시스템을 이용하는 동안 항상 모든 창의 상위에 있어야 한다. 하지만 매번 다른 창에 있는 메뉴를 마우스를 이용해 조작하다 보면 다른 창의 하위에 있게 되는 문제가 있었다. 때문에 모니터링 창을 생성한 후 창의 이름을 검색하는 findWindow() 함수를 이용해 창의 핸들러를 받아온 후 setWindowPos() 함수로 창의 최상위에 위치시켰다. 또한, 각각의 버튼 등은 사용자의 시선이 위치할 경우 색이 바뀌도록 하여 사용자가 자신이 어느 버튼을 선택하게 될지 알도록 하였다. 각각의 버튼은 이미지를 불러와 사용하기 때문에 기능을 확장하거나, 변경이 쉽도록 구현하였다.

3.3 커서 이동 기능 구현

시선 추적에는 Tobii 4c eye tracker 하드웨어를 이용하였다. 화면상의 시선 좌표는 하드웨어의 제조사에서 제공하는 Interaction library를 이용해 60Hz의 빈도로 시선 좌표를 받아와 이 값을 필터링한 좌표를 생성하였다[10]. Interaction library에서 받아온 시선 추적 좌표를 그대로 마우스 커서에 사용할 경우 안구의 떨림에 따라 커서가 심하게 떨려 사용자에게 불편함을 줄 수 있기 때문에 받아온 좌표를 필터링한 후 사용했다. 그 후 필터링 된 좌표로 Winuser.h윈도

우 API를 이용해 커서를 이동시켰다[11]. 새로 들어온 좌푯값과(nowXpos, nowYpos) 현재 좌푯값(Xpos, Ypos)을 threshold 값 (T)에 따라 처리해주었다. 업데이트되는 좌표 coordinates(x, y)에 대한 식은 다음과 같다.

$$coordinates(Xpos, Ypos) \tag{1}$$

$$Xpos = \begin{cases} nowXpos, & |Xpos - nowXpos| \geq T \\ Xpos, & |Xpos - nowXpos| < T \end{cases}$$

$$Ypos = \begin{cases} nowYpos, & |Ypos - nowYpos| \geq T \\ Ypos, & |Ypos - nowYpos| < T \end{cases}$$

수식(1)에 의해 이전 시선 좌푯값 Xpos와 새로운 시선 좌표 nowXpos의 차의 절댓값이 임계값 (T)이상이 되는 좌표변화가 생길 경우 커서의 위치를 업데이트하도록 하였다. Y축 좌표 Ypos에 대해서도 이와 같은 수식이 적용된다. 임계값 (T)는 계속해서 화면의 왼쪽 아래에 떠 있는 모니터링 및 설정 창을 통해 사용자가 확인하고 조절할 수 있도록 구현했다.

3.4 Mouth-popup 이벤트 발생 시스템 구현

프로그램이 시작되면 웹캠을 통해 입력되는 영상의 프레임마다 Fig. 5와 같은 과정이 수행되도록 구현하였다. 가장 먼저 영상을 입력받은 후 프레임별로 320×240 크기로 변환한다. 입력되는 영상의 크기가 커질수록 얼굴인식, 특징점 추출 등의 처리 시간이 길어지기 때문이다. 그 후 크기가 변환된 영상에서 얼굴의 영역을 찾고 영상에 얼굴이 있으면 입의 특징점을 찾아낸다. 본 논문에서는 얼굴을 인식하는 과정과 특징점 추출 과정을 머신러닝 알고리즘 라이브러리인 Dlib의 frontal_face_detector 클래스를 이용해 얼굴을 인식하고, shape_predictor 클래스와 iBUG 300-W 얼굴 특징점 데이터세트를 통해 미리 학습된 가중치를 이용해 68개의 얼굴 특징점을 추출했다.

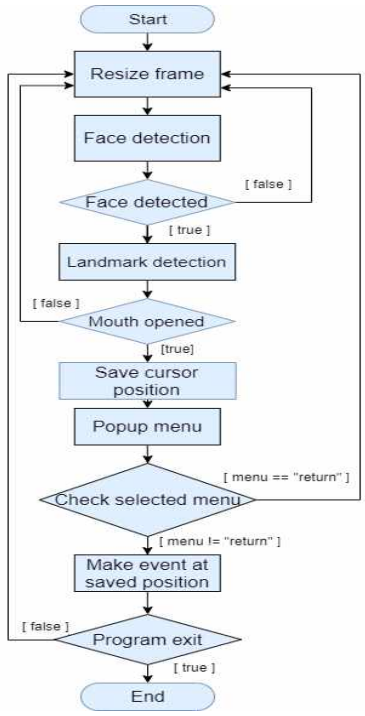

Fig. 5. Flow chart for Mouth-popup system process.

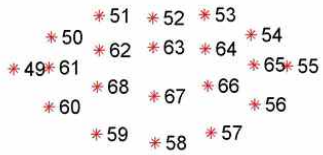

Fig. 6. landmark of lips.

shape_predictor에서 찾아내는 얼굴의 특징점 중 Mouth-popup 시선추적 마우스 시스템에서 사용되는 입술의 특징점은 Fig. 6과 같다. 입의 개폐 여부를 감지하기 위해 63번 특징점과 67번 특징점을 이용했다. 입의 개폐 여부를 확인하는 과정은 윗입술과 아랫입술의 특징점인 63번과 67번 점의 좌표 평면상의 거리 (D)를 계산한 후 이 거리가 임계값을 넘었는지 확인해 입의 열림을 인식했다. 이 임계값 또한 사용자가 설정 창을 통해서 변경할 수 있도록 구현했다.

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{2}$$

입 열림을 감지했을 경우, 입이 열리기 전 마지막으로 시선이 있던 마우스 커서의 위치를 저장하고, 팝업 메뉴를 띄운다. 입을 열고 있는 상태에서 선택할 메뉴에 시선을 향한 후 입을 닫으면 입을 닫기 직전에 보고 있던 메뉴가 선택된다. 그 후 미리 저장해 두었던 좌표에 우클릭, 좌 클릭, 더블 클릭 등의 선택된 이벤트가 실행되도록 구현했다. 이벤트를 발생시키는 과정은 Winuser.h에서 제공하는 mouse_event 함수를 이용해 선택된 이벤트를 입을 열기 전 저장된 커서의 위치에 발생시켰다.

4. 성능 실험 및 사용자 평가

4.1 실험 설계

본 실험은 논문에서 제시하는 Mouth-popup 시선추적 마우스 인터페이스가 기존의 인터페이스에 비해 오입력 횟수와 기능 수행 시간을 줄여 성능을 개선함을 확인함에 목적이 있다. 실험은 크게 2가지 분류로 나누어져 있다. 첫 번째 실험은 객체 선택 실험이다. 본 실험은 사용자가 화면상에 있는 텍스트를 읽고 필요한 객체를 선택해야 하는 상황을 가정해 클릭을 수행하는 실험이다. 두 번째 실험은 기능성 실험이다. 해당 실험은 컴퓨터를 사용하며 수행해야 하는 좌 클릭, 우클릭 더블클릭, 스크롤 등 마우스의 기능을 수행하는 실험이다.

두 실험에서 Mouth-popup 시선추적 마우스 시스템과 기존 방식인 dwell time과 메뉴를 함께 사용한 시선추적 인터페이스를 이용해 각각 태스크를 수행하고 두 가지 항목을 평가한다. 첫 번째는 각 시도에서 발생하는 오입력인 false Positive를 측정하고, 두 번째는 각 실험에서 요구하는 행위를 수행하는 데 걸리는 시간을 측정한다.

실험 환경은 1980×1080px의 해상도를 가진 모니터와 Tobii 4c eye tracker, 1280×720px의 웹캠을 탑재한 window 10 OS의 PC에서 진행되었다. 각 실험의 태스크 제시, 오입력 측정, 시간 측정 등에 사용된 테스트 툴은 브라우저상에서 구동되는 웹 앱의 형태로 자체 제작해서 사용하였다. 해당 테스트에서 입력의 대상이 되는 버튼들은 가로 100×100px의 크기로 통일하였다. 또한, 실험 중 입력 영상에 최대한 같은 광량을 제공하기 위해 조명을 이용해 광량을 충분히 확보한 후 실험을 제공했다.

각 실험을 진행하는 피실험자는 20대 남성 10명으로 Mouth-popup, dwell time 등 각각의 방식으로 태스크를 3회씩 연습한 후 실험 결과를 측정했다. dwell time은 dwell 시간이 실험의 결과에 영향을 줄 수 있으므로 0.6초, 1초로 각각 설정하여 dwell 시간에 따른 정확도와 수행 시간도 측정하였다. 또한, 실험자마다 실험 시작 전 시선추적 좌표 보정 과정과 Mouth-popup 감도 설정 과정을 거쳤다. 피실험자는 손이나 다른 신체 부위를 움직이지 못하고 목 위의 신체 부위만 사용하여 인터페이스를 테스트하도록 했다.

4.2 객체 선택 실험

본 실험은 사용자가 화면상에 있는 텍스트를 읽고 필요한 객체를 선택해야 하는 상황을 가정하는 실험이다. 이러한 작업은 실제 PC 사용 환경에서 웹 콘텐츠를 이용하거나 게임을 하는 등 여러 상황에서 빈번하게 수행되는 작업이다. 본 실험의 오입력 횟수와 수행 시간을 비교해봄으로써 Mouth-popup 시선추적 마우스 시스템이 이러한 상황에서 화면상의 객체 선택 성능을 개선하는지 여부를 확인할 수 있다.

객체 선택 실험은 Fig. 7처럼 구성된 화면에서 진행되었다. 상단에 하나의 유일한 숫자와 여러 무작위의 문자와 특수 문자조합으로 되어있는 문자열 중에서 숫자를 찾아내 해당 숫자가 적혀있는 100×100px 크기의 버튼 20개 중 하나를 선택하도록 하였다. 숫자를 무작위의 문자열 속에 배치함으로써 사용자가 실제 사용 상황에서 특정 텍스트를 오래 읽는 상황을 재연하려고 했다. 또한, 버튼의 배치는 문자열 영역을 제외하고 1980×900 크기의 화면에 고르게 분산시켜 숫자에 따라 화면의 여러 영역을 클릭하도록 해 실제 사용 상황과 유사하게 설계하였다. 제시되는 숫자는 모든 실험자 간에 통일해 미리 정해둔 10개의 숫자가 제시되며 선택 시에, 해당 버튼이 아닌 다른 버튼이나 배경을 누를 때 오입력이 발생한 것으로 처리하고 계수했다. 수행 시간은 한 숫자가 제시된 후 이를 확인 하고 이 숫자를 찾아 해당 숫자의 버튼을 클릭하기까지의 시간을 기록했다.

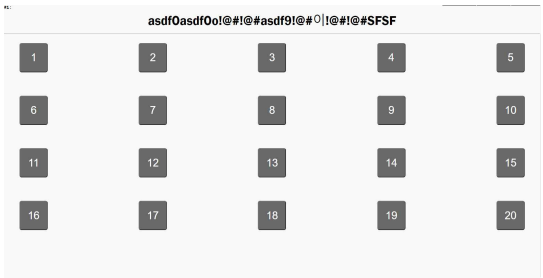

Fig. 7. The object selection experiment.

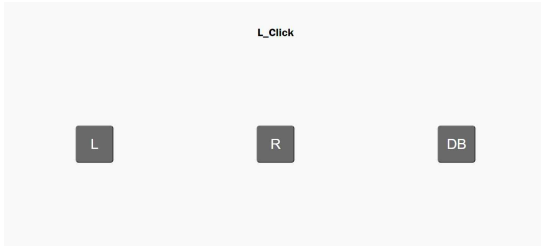

Fig. 8. The click functionality experiment.

4.3 기능성 실험

본 실험은 화면상에 있는 객체에 사용자가 좌 클릭, 우 클릭, 더블클릭 등 특정 이벤트를 발생시키거나, 스크롤 등으로 화면을 이동시키는 상황을 가정하는 실험이다. 해당 실험을 함으로써 Mouth-popup 입력 방식이 각각 다른 이벤트를 발생시킬 때 입력의 정확성과 속도를 개선하는지 확인할 수 있다.

실험 방법은 클릭 기능과 스크롤 실험을 분리하여 실험하였다. 클릭 기능성 실험은 Fig. 8과 같이 각각 좌 클릭, 우 클릭, 더블클릭을 담당하는 버튼을 화면상에 배치하고 화면의 상단에서 미리 정해둔 순서대로 요구하는 기능을 제시하면 버튼에 해당하는 입력을 수행하도록 하였다. 본 실험에서는 각 이벤트를 2번씩 총 6번 무작위로 제시해 수행하도록 하였다. 이 실험 또한 발생시켜야 할 이벤트를 확인 후 이를 사용자가 선택하기까지의 시간을 재고, 제시한 버튼이 아니거나 제시한 동작이 아닐 경우 false positive를 계수했다.

스크롤 실험은 Fig. 9와 같은 화면에서 왼쪽 위 끝의 start 버튼을 누른 후 2000px 하단에 있는 end 버튼을 누르는 방식으로 시행되었다. 해당 실험은 5번 반복했고 start를 누르는 시점과 end 버튼을 누르기까지의 시간을 측정했다. 또한, start와 end 버튼 이외의 화면이나 객체를 누를 때마다 false positive를 계수했다.

Fig. 9. The scroll functionality experiment.

4.4 실험 결과 및 분석

객체 선택 실험, 클릭 기능성 실험, 스크롤 기능성 실험 각각의 결과는 다음과 같다. 총 10명의 피실험자로부터 얻어진 결과이며 각 실험자의 결과를 모두 합산하여 평균을 냈다.

객체 선택 실험 결과는 Table 1과 같다. 세 가지의 비교군 중 수행 시간은 Mouth-popup이 평균 3.13초로 dwell-time (0.6) 초보다는 0.84초로 약 21.15%가 낮아지고 dwell- time(1) 초 방식보다는 1.46초로 약 31.80% 감소하여 유의미한 결과가 있었다. 오입력 수는 Mouth-popup의 오입력이 평균 0.1회로 dwell time (0.6) 초 방식보다는 0.73회 낮아져 약 87.95% 감소하였고 dwell time(1) 초 방식보다 0.45회 낮아져 약 81.81% 감소한 것을 확인할 수 있었다.

본 실험에서는 Mouth-popup 방식이 문서를 읽고 객체를 선택하는 상황에서 특히 정확도를 크게 향상시킴을 알 수 있었다. dwell-time(0.6)와 dwell-time (1)을 비교해보면 비교적 아무 입력을 하지 않고 응시할 수 있는 최대 시간이 긴 dwell-time(1) 방식이 dwell-time(0.6) 초 방식보다 문자열을 응시하는 과정에서 오입력 횟수가 약 33.73% 적음을 확인 할 수 있었다. 이러한 오 입력 횟수에 대한 결과는 본 실험 과정에서 자연스럽게 난독화 된 문자열 속에서 숫자를 찾아내면서 문자열을 오래 응시하게 되는데, dwell time과 같은 경우는 해당 과정에서 응시 시간이 미리 정해둔 dwell-time을 넘겨 오입력을 발생시키는 경우가 많았기 때문으로 분석된다. 하지만 두 방법 모두 문자열을 찾기 위해 일정 시간 응시를 할 경우 오입력이 발생하는 반면 Mouth-popup의 경우 이러한 문제가 없이 원하는 시점에 이벤트를 발생시킬 수 있어 상대적으로 오입력 횟수가 감소해 두 방식에

Table 1. The object selection experiment results

Assessment item	Mouth-popup	Dwell-time(0.6)	Dwell-time(1)
Time(sec)	3.13	3.97	4.59
False positive	0.10	0.83	0.55

Table 2. The click functionality experiment results

Assessment item	Mouth-popup	Dwell-time(0.6)	Dwell-time(1)
Time(sec)	2.42	5.50	5.70
False positive	0.33	0.91	0.93

Table 3. The scroll functionality experiment results

Assessment item	Mouth-popup	Dwell-time(0.6)	Dwell-time(1)
Time(sec)	5.1	5.89	8.48
False positive	0.28	0.66	0.5

비해 정확도를 향상 시키는 것을 확인 할 수 있었다. 다만 Mouth-popup의 경우 화면의 우측 하단에 위치하는 20번 버튼을 선택할 때 설정 및 팝업창이 해당 버튼의 일부를 가려 실험자가 가려지지 않은 일부를 클릭해야 한다는 단점을 발견 할 수 있었다.

클릭 실험 결과는 Table 2와 같다. Mouth-popup이 평균 2.42초로 각각 dwell-time(0.6)보다는 3.08초 빨라져 56.00% 감소했고 dwell-time(1)보다는 3.28초 빨라져 57.54% 감소하였다. 또한, 오입력 횟수는 Mouth-popup이 평균 0.33회로 dwell-time(0.6)보다 0.58회로 63.73% dwell-time(1)보다 0.6회 감소해 64.51% 개선함을 확인할 수 있었다.

클릭 기능 실험 결과 좌클릭 우클릭 더블클릭 등의 이벤트를 발생시키는 과정에서 수행 시간과 정확도 모두 큰 폭으로 유의미하게 증가함을 확인 할 수 있었다. 이는 dwell time의 경우 이벤트를 발생시키기 전, 먼저 플로팅 되어있는 메뉴로 가서 발생시킬 이벤트를 dwell time 방식으로 선택 후 객체로 다시 이동해 dwell time을 대기해 이벤트를 발생시키기 때문에 이벤트를 발생시키는 최소 시간이 dwell-time의 대기시간의 2배로 늘어나며 이러한 과정에서 오입력이 일어날 가능성이 커지기 때문으로 분석된다. 이에 반해 Mouth-popup 방식의 경우 원하는 객체로 이동 후 입을 벌려 바로 원하는 이벤트를 응시 후 닫는 과정만 거치면 대기 시간이나 추가 이동 없이 원하는 시점에 이벤트를 발생시킬 수 있어 수행 시간과 정확도가 향상되는 것으로 분석된다.

스크롤 기능 실험 결과는 Table 3과 같다. Mouth-popup 방식의 평균 수행 시간은 5.1초로 dwell-time(0.6)보다 0.79초로 13.41%가 감소하였으며 dwell-time(1) 방식보다 3.38초가 빨라져 39.85%가 감소했다. 오입력 횟수는 Mouth-popup이 0.28로 각 평균 dwell-time(0.6)보다 0.38회로 57.57% 감소하였고 dwell-time(1)보다 0.22회로 43.99% 감소하는 결과를 보여주었다.

스크롤 기능 실험 결과 수행 시간과 정확도 모두 향상되었다. 이는 Mouth-popup 방식은 직접 스크롤 이벤트를 제공하는 반면 dwell-time 방식은 직접 스크롤바를 이동시켜야 하기 때문에 이를 선택하기 위해 이동하고 응시하는 과정에서 추가적인 시간과 오입력이 발생해 이러한 결과가 나온 것으로 분석된다.

5. 결 론

본 논문에서는 시선추적 인터페이스에서 빈번히 발생하는 문제인 Midas touch problem을 방지하면서, 시선추적기와 일반 RGB 카메라 외에 특수한 장비나 음성 등을 이용하지 않고도 전신이 마비된 환자의 대부분이 PC에서 마우스의 기능을 정상적으로 이용할 수 있는 새로운 마우스 이벤트 입력 방식인 Mouth-popup 시선추적 마우스 인터페이스를 제시하고 각 방법의 오입력 횟수와 입력 속도 비교를 통한 성능 개선 검증을 진행했다.

제안한 Mouth-popup 시선추적 마우스 인터페이스는 일반 웹캠을 이용해 실시간 영상에서 얼굴의 특징점을 추출해 윗입술과 아랫입술의 거리를 통해 일정 넓이 이상의 입 벌림을 감지하고 입을 벌렸을 때 이벤트를 선택할 수 있는 팝업 메뉴를 띄워 주는 마우스 인터페이스로서 목을 움직일 수 없거나 말을 하지 못하는 전신 마비 환자도 도움 없이 PC를 사용할 수 있도록 인터페이스를 설계하고 구현하였다. 구현된 Mouth-popup 방식을 검증하는 과정에서 Dwell time과 메뉴를 함께 사용한 시선추적 인터페이스와 Mouth-popup 방식을 비교해서 실험을 진행했다. 그 결과 Mouth-popup 시선추적 마우스 인터페이스가 기존 방식에 비해 객체를 선택하거나 특정 이벤트를 발생시키고 스크롤을 하는 상황에서 오입력 횟수를 줄이고 실행 시간을 줄여 사용성과 성능을 개선함을 확인할 수 있었다. 신체상의 이유로, 장비에 대한 접근성의 이유로 기존의 인터페이스 방식을 사용하지 못했던 장애인들도 Mouth-popup 인터페이스를 사용하면 PC를 원활히 사용할 수 있을 것으로 예상된다.

본 연구 이후 이루어져야 할 추후 연구는 다음과 같다. 첫 번째, 변화하는 조도와 얼굴의 각도에 대응할 수 있는 영상 처리 시스템에 관한 연구가 필요하다. 현재는 광량이 부족하거나 얼굴의 각도에 따라서 얼굴의 특징점을 잘 읽어내지 못하는 문제가 있다. 실제 사용에서는 광량이 계속해서 변하고 전신 마비 장애인의 경우 얼굴을 정면으로 지지하지 못하는 경우가 많기 때문에 추후의 연구에서 얼굴 인식에 쓰이는 CNN 신경망을 개선해 다양한 상황에서 얼굴을 강건히 인식하도록 하고, 얼굴의 회전 등에 대해 대응할 수 있는 영상처리 분야의 추가적인 연구가 필요

할 것으로 예상된다.

두 번째, UI/UX 개선에 대한 추가적 연구가 필요하다. 실험 중 본 방식이 dwell time에 비해서 사용법이 생소해 사용법에 대한 적응 기간이 비교적 더 필요하고, 설정 및 모니터링 창이 우측 하단의 객체들을 일부 가린다는 문제점을 발견 할 수 있었다. 이를 해결하기 위해 추후에 사용자가 이 모니터링 창을 손쉽게 여닫을 수 있도록 하거나, 모니터링 창의 영상을 상태표시 아이콘으로 대체 하는 등 추가적인 UI/UX에 관한 연구가 필요해 보인다.

세 번째, 다양한 상황에서의 추가적인 실험이 필요하다. 추후 연구에서는 전신 마비 장애인들을 대상으로 실험을 진행해 전신 마비 장애인들에게 적용했을 때에도 성능 개선이 유효한지 확인하고, 장애인들의 특성을 파악해 본 인터페이스를 최적화하는 연구가 필요할 것으로 예상된다. 또한, 현재는 실험 환경과 방법이 다르기 때문에 타 연구의 실험 결과를 1:1 비교 할 수 없어 시선추적 인터페이스로 널리 쓰이는 dwell time 방식 한 가지를 구현해 비교 실험을 진행했다. 때문에 추후에 같은 환경과 실험 내용으로 더 많은 입력 방식의 성능을 비교하는 연구가 필요하다.

REFERENCE

[1] H.N. H, K.M. Um, E.B. Song, C.S. Kim, and J.W. Heo, "Development and Clinical Evaluation of Wireless Gyro-mouse for the Upper Extremity Disabled to Use Computer," *Science of Emotion and Sensibility*, Vol. 9, No. 2, pp. 93-100, 2006.

[2] W. Barfield and T. Furness, *Virtual Environments and Advanced Interface Design,* Oxford University Press, New York, 1995.

[3] X. Zhang, X. Liu, S.M. Yuan, and S.F. Lin, "Eye Tracking Based Control System for Natural Human-computer Interaction," *Computational Intelligence and Neuroscience*, Vol. 2017, No. 1, pp. 1-9, 2017.

[4] H. Singh and J. Singh, "Real-time Eye Blink and Wink Detection for Object Selection in HCI Systems," *Journal on Multimodal User Interfaces,* Vol. 12, No. 1, pp. 55-65, 2018.

[5] L.L. Palmer, "Inability to Wink an Eye and Eye Dominance," *Perceptual and Motor Skills*, Vol. 42, No. 3, pp. 825-826, 1976.

[6] S. Soundarajan and H. Cecotti, "A Gaze-based Virtual Keyboard Using a Mouth Switch for Command Selection," *Proceeding of Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3334-3337, 2018.

[7] J.H. Park, M.H. Park, and S.B Lim, "A Proposal of Eye-voice Method Based on the Comparative Analysis of Malfunctions on Pointer Click in Gaze Interface for the Upper Limb Disabled," *Journal of Korea Multimedia Society*, Vol. 23, No. 4, pp. 566-573, 2020.

[8] W.A. Choi, J.C. Shin, D.Y. Lee, D.H. Kim, S.D. Kim, S.W. Kang, et al., "Noninvasive Respiratory Management for Patients with Cervical Spinal Cord Injury Annals of Rehabilitation Medicine," *Annals of Rehabilitation Medicine*, Vol. 34, No. 5, pp. 518-523, 2010.

[9] WHO. ISCOS, *Iternational Perspectives on Spinal Cord Injury*, Geneva, Switzerland, 2013.

[10] Winuser.h(2019), https://docs.microsoft.com/en-us/windows/win32/api/winuser (accessed July 24, 2020).

[11] Interaction Library(2018), https://developer.tobii.com/consumer-eye-trackers-interaction-library (accessed July 9, 2020).

변 주 영

숭실대학교 IT대학 글로벌미디어학부 학사과정
관심분야 : HCI, 컴퓨터비전, 웹, 미디어아트

정 기 철

숭실대학교 IT대학 글로벌미디어학부 교수
Post-Doc, PRIP Lab.,Michigan State University
Intelligent User Interface Group, (DFKI,독일), Machine Understanding Division, ETL방문연구원
SIAT (School of Interactive Art and Technology, SFU, Canada) 방문 교수, 2009~2010
경북대학교 컴퓨터공학 공학박사
관심분야 : 인공지능, 컴퓨터비전, HCI