사물인터넷융복합논문지 Vol. 9, No. 6, pp. 125-134, 2023 https://doi.org/10.20465/KIOTS.2023.9.6.125

MediaPipe Face Mesh를 이용한 얼굴 제스처 기반의 사용자 인터페이스의 성능 개선

목진왕[1], 곽노윤[2*]
[1]광주과학기술원 AI대학원 석박사통합과정, [2]백석대학교 컴퓨터공학부 교수

Performance Improvement of Facial Gesture-based User Interface Using MediaPipe Face Mesh

Jinwang Mok[1], Noyoon Kwak[2*]
[1]MS/Ph.D. Integrated Candidate of Graduate School of Artificial Intelligence, GIST
[2]Professor, Division of Computer Engineering, Baekseok University

요 약 본 논문은 MediaPipe Face Mesh 모델을 이용해 일련의 프레임 시퀀스에서 얼굴 제스처를 인식해 해당 사용자 이벤트를 처리하는 얼굴 제스처 기반의 사용자 인터페이스 선행 연구의 성능 개선 방안을 제안함에 그 목적이 있다. 선행 연구는 MediaPipe Face Mesh 모델에서 선택한 7개의 랜드마크의 3차원 좌표들로부터 얼굴 제스처를 인식해 해당 사용자 이벤트를 발생시키고 이에 대응하는 명령을 수행하는 것이 특징이다. 제안된 방법은 그 과정에서 커서 위치들에 적응형 이동 평균 처리를 적용해 미세 떨림을 완화함으로써 커서 안정화를 도모하고, 양안 동시 개폐 시에 양안의 일시적 개폐 불일치를 차단해 그 성능을 개선하였다. 제안된 얼굴 제스처 인터페이스의 사용성 평가 결과, 얼굴 제스처의 평균 인식률이 선행 연구에서 95.8%였던 것에 비해 98.7%로 상향되는 것이 확인되었다.

주제어 : 얼굴 제스처 인식, 양안 개폐, MediaPipe, Face Mesh Model, NUI

Abstract The purpose of this paper is to propose a method to improve the performance of the previous research is characterized by recognizing facial gestures from the 3D coordinates of seven landmarks selected from the MediaPipe Face Mesh model, generating corresponding user events, and executing corresponding commands. The proposed method applied adaptive moving average processing to the cursor positions in the process to stabilize the cursor by alleviating microtremor, and improved performance by blocking temporary opening/closing discrepancies between both eyes when opening and closing both eyes simultaneously. As a result of the usability evaluation of the proposed facial gesture interface, it was confirmed that the average recognition rate of facial gestures was increased to 98.7% compared to 95.8% in the previous research.

Key Words : Facial Gesture Recognition, Binocular Opening and Closing, MediaPipe, Face Mesh Model, NUI

본 논문은 2023년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 연구과제(2021RIS-004)로 수행되었음.
*교신저자 : 곽노윤(nykwak@bu.ac.kr)
접수일 2023년 9월 29일 수정일 2023년 11월 21일 심사완료일 2023년 11월 23일

1. 서론

현재 컴퓨터나 디지털 기기를 제어하는 가장 보편적인 HCI(Human Computer Interaction) 기술은 키보드와 마우스 기반의 GUI(Graphic User Interface)이다. 마우스는 GUI의 핵심적 포인팅 입력 장치이다. 사용자가 마우스 커서 위치에 클릭, 드래그, 스크롤 등과 같은 사용자 이벤트가 발생하도록 마우스를 조작하면, 운영체제는 그 사용자 이벤트에 대응하는 명령을 처리함으로써 컴퓨터에서 사용자 조작을 실현한다. 센서 기술과 인공지능, 그리고 CPU, GPU, 메모리 등의 비약적 발전에 힘입어 키보드와 마우스만을 이용한 전통적 방식이 아니라 사용자의 음성, 시선, 표정, 제스처, 터치, 근전도, 뇌파 등을 통해 디지털 기기를 조작하는 신개념의 NUI (Natural User Interface) 방식들이 속속 연구되고 있다. 실용적이고 완성도 높은 NUI의 구현은 HCI 분야의 숙원이자 당면 과제이다. NUI 방식은 사용자 인터페이스는 인간과 기계 사이의 접점을 없애고 인간의 의도를 자율적으로 파악하는 방향으로 발전하고 있다. NUI는 직관적이고 자연스러운 사용자 경험을 제공할 뿐만 아니라 컴퓨터 조작 시, 키보드와 마우스가 필히 장착돼야 하는 제약을 없애준다[1-4].

이러한 전환의 근간을 제공하는 HCI 기술들 중에서 손, 얼굴, 몸짓 등의 제스처 인식 기술은 1990년대 이래로 인간-로봇 상호 작용[5], 3D 게임 인터페이스[6], 가상현실[7,8], 가전기기나 모바일 장치와의 상호 작용[9], 의학적 자세 교정이나 운동량 측정[10], 수화 인식[11], 드론 제어[12] 등 다양한 분야에서 걸쳐 연구되어 왔다. 그리고 실감 미디어 분야에서도 MANO[13,14], Fast Hand[15], DIGIT[16], 플로팅 홀로그램 캐릭터 제어[17], MVHM[18] 등이 연구되고 있다.

최근 들어 구글의 MediaPipe[19]가 제스처 인식 분야에서 크로스 플랫폼 프레임워크의 중심으로 부상하면서 MediaPipe Hands 모델 기반의 손 제스처 인터페이스 기술들[20-23]이 제안되고 있다. 그러나 손 제스처 인터페이스의 경우, 장시간 사용 시 피로감이 누적되거나 경련이 날 수 있으며, 양손이 자유롭지 못한 경우, 제스처 인식이 불가하다는 단점이 있다. 반면, 얼굴 제스처는 장시간 사용 시 손 제스처에 비해 피로감이 덜하고, 무엇보다도 얼굴과 시선이 함께 움직여 더욱 직관적이라는 장점이 있다[4]. 이러한 장점에 기인해 많은 얼굴 제스처 인터페이스 기술들[24,25]이 발표되어 왔고 특히, 최근엔 MediaPipe Face Mesh 모델 기반의 얼굴 인식[26] 혹은 얼굴 제스처 인식 기술[27]이 속속 제안되고 있다.

한편, 본 논문의 연구진도 직관성과 편의성을 높인 MediaPipe Face Mesh 모델에 기반의 얼굴 제스처 인터페이스 기술[28]을 발표한 바 있는데, 본 논문은 이를 확장 재구성함과 동시에 이 선행 연구[28]의 조작성을 제고하기 위해 커서 안정화 및 양안 개폐 인식 성능의 개선 방안을 제안함에 그 목적이 있다.

통상, MediaPipe Face Mesh 모델의 추정 결과인 3D 랜드마크들(3D landmarks)은 단안 카메라에 기초해 입력 프레임 전체에 대한 깊이가 아니라 입력 프레임 중 얼굴 영역 내 상대적 깊이로 추정된 것이기에 깊이 값의 신뢰도가 다소 낮다는 단점이 있다. 이런 이유로 인해 본 연구진의 선행 연구[28]에서 얼굴의 Pan 각도, Tilt 각도를 통해 조작되는 커서의 움직임에 잦은 미세 떨림이 발생해 정교한 인터페이스 조작에 방해가 되고 있었다. 또한 양안 동시 개폐 시, 좌우 안구가 근소한 차이로 불일치되게 개폐됨으로 인해 특정 제스처들의 인식률이 다소 저하되는 문제도 있었다. 이러한 문제들을 해결하기 위해 본 논문에서는 모든 커서 위치에 적응형 이동 평균 처리를 적용해 미세 떨림을 완화함으로써 커서 조작의 안정화와 정교화를 도모하고자 한다. 또한 얼굴 제스처 기반의 사용자 인터페이스에서 양안 동시 개폐 시, 양안 동시 개폐의 불연속성을 반복적으로 검사해 양안의 일시적 개폐 불일치를 차단함으로써 해당 사용자 이벤트들의 오작동을 개선하고자 한다.

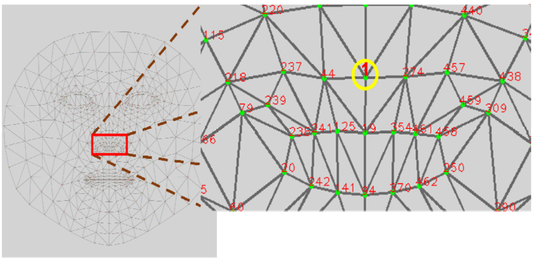
본 논문에서는 선행 연구인 MediaPipe Face Mesh 모델을 이용한 얼굴 제스처 인터페이스[28]에 대해 선술하고 그 문제점들의 개선 방안을 후술할 것이다.

2. 제안된 인터페이스의 설계 및 구현

2.1 MediaPipe Face Mesh 모델

최근 딥러닝 기술의 발달과 해당 기술의 오픈 소스들(open sources)과 소프트웨어 라이브러리들(software libraries)의 공개로 인해 사전 학습된 모델을 활용해 소프트웨어를 연구·개발하는 사례가 날로 증가하고 있다. MediaPipe[19]는 실시간 미디어 처리에 특화된 머신러닝 프레임워크로, 구글이 공개했으며 Face Mesh, Hands, Pose, Iris, Holistic, Object Detection, Face Detection, Box Tracking 모델 등의 다양한 사전 학습된 딥러닝 솔루션을 제공하고 있다. 그 중 Face Mesh

모델은 입력 영상에서 사용자의 얼굴을 검출한 뒤 검출된 영역 내에서 [Fig. 1]과 같이 468개의 얼굴 3D 랜드마크들을 추론한다. Face Mesh 모델은 입력 프레임에서 얼굴의 위치를 탐색하는 Detector 모델과 해당 위치에서 대략적인 3차원 표면을 예측하는 3D Face Landmark 모델로 구성되어 있다. 입력 프레임 내 전체 공간상에서 얼굴의 3차원 좌표를 추정하면 전역적 깊이 정보를 얻을 수 있다는 장점이 있다. 하지만 과도한 연산으로 실시간 처리에 부담이 된다. 이에 따라 Face Mesh 모델의 경우, 먼저 입력 프레임에서 얼굴 영역을 탐지한 후, 얼굴 영역 내에서 3차원 좌표를 추정한다. 이에 따라 불필요한 연산을 줄이고 3D 랜드마크 추정에 집중함으로써 얼굴 영역 내 3차원 좌표값의 추정 정확도를 높인다. Face Mesh 모델은 오른손 정규직교 3D 좌표계(right-handed orthonormal metric 3D coordinate system)를 3D 얼굴 랜드마크 좌표계로 삼는데, 추정된 각 3D 랜드마크의 좌표값은 X, Y, Z축의 값으로 구성되고 각각 $[0, 1]$, $[0, 1]$, $[-1, 1]$의 범위를 갖는다. X, Y축의 값은 각각 이미지 너비와 높이를 기준 삼아 각 얼굴 랜드마크의 위치를 $[0, 1]$ 사이의 값으로 정규화한 것이다. Z축의 값은 얼굴 랜드마크 깊이를 $[-1, 1]$ 사이의 값으로 나타낸 것이며 머리의 중심에 원점이 있다. 이 깊이의 크기는 X축과 동일한 스케일을 사용하고 값이 작을수록 카메라에 가까운 얼굴 랜드마크다[28].



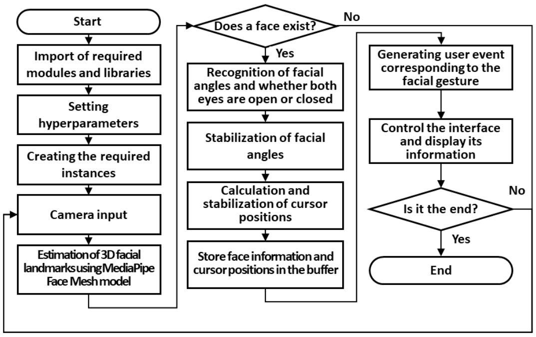
[Fig. 1] 3D facial landmarks of the MediaPipe Face Mesh model and the nose tip landmark marked with a yellow circle

2.2 사용자 인터페이스의 실행 과정

본 논문에서는 Face Mesh 모델을 이용해 사용자의 얼굴 제스처를 인식한다. [Fig. 2]는 본 논문의 연구진이 제안한 얼굴 제스처 기반의 사용자 인터페이스의 순서도이다.

제안된 사용자 인터페이스의 실행 과정은 [Fig. 2]와 같이 우선, 필요한 모듈들과 라이브러리들을 임포트(import)하고 하이퍼파라미터들(hyperparameters)를 기설정된 값으로 세팅한 후, 필요한 인스턴스들을 생성하는 것으로부터 시작된다.

본 논문의 얼굴 제스처 인터페이스 시스템은 일련의 카메라 입력 시퀀스가 들어오면, Face Mech 모델을 활용해 입력 프레임에서 얼굴 영역을 탐지한 후, 얼굴 영역 내에서 [Fig. 1]의 3D 얼굴 랜드마크 좌표들을 추정하는 과정을 반복한다. 이 과정에서 얼굴 영역이 탐지되지 않으면 카메라로부터 그 다음 프레임을 입력받는다. 만약 일련의 카메라 입력 시퀀스의 3D 얼굴 랜드마크 좌표들을 검사해 얼굴 제스처 인터페이스 활성화 이벤트가 입력된 것으로 판단되면, 추정된 3D 얼굴 랜드마크 좌표들 중에서 선택한 7개의 랜드마크 좌표들로부터 양안 개폐 여부와 그 유지시간, 그리고 얼굴의 Pan 각도, Tilt 각도, Roll 각도 등의 얼굴 제스처를 인식해 해당 사용자 이벤트를 발생시키고 이 사용자 이벤트에 대응하는 명령을 수행하도록 인터페이스를 제어한다.



[Fig. 2] Flowchart of the proposed facial gesture-based user interface

여기서 얼굴의 Pan 각도, Tilt 각도는 [Fig. 1]에서 노란색 원으로 표기된 3D 코끝 랜드마크 좌표를 이용해 산출하는데, 이렇게 구한 2개의 얼굴 각도들을 이용해 모니터 화면상 커서 위치의 수평 및 수직 좌표를 계산한다. 이를 위해 제안된 방법에서는 우선, 얼굴 제스처 인터페이스 모드가 활성화되자마자 최초 소정 시간 동안 사용자가 얼굴의 움직임을 억제하고 모니터 화면의 정중앙을 바라보도록 한 채 입력받은 프레임들의 코끝 랜드마크 좌표들을 합산해 평균한 값으로 3D 얼굴 랜드마크 좌표계의 각도 산출 기준점으로 삼는다. 각도 산출 기준점을 이렇게 평균값으로 초기화하는 이유는 얼굴 각도 산출 시, 그 기준이 되는 각도 산출 기준점을 안정화시키기 위한 것이다. 이후 3D 얼굴 랜드마크 좌표계의 원점과 각

도 산출 기준점을 연결하는 3차원 벡터를 구한다. 그리고 3D 얼굴 랜드마크 좌표계의 원점과 이 초기화 이후에 입력되는 코끝 랜드마크 좌표를 연결하는 3차원 벡터를 구한다. 이후, 이 두 3차원 벡터 사이의 수평 방향과 수직 방향의 각도를 계산해 얼굴의 Pan 각도와 Tilt 각도로 삼는다. 이렇게 구한 얼굴의 Pan 각도와 Tilt 각도는 각각 화면상 커서 위치의 수평 좌표와 수직 좌표에 대응되도록 환산된다. 이때, 3D 얼굴 랜드마크 좌표계에서 코끝 랜드마크 좌표가 각도 산출 기준점과 일치해 그 수평 및 수직 각도가 모두 0°가 될 시, 모니터 화면에 표시할 커서의 수평 좌표와 수직 좌표가 화면 정중앙에 정해지도록 환산한다. 그리고 3D 얼굴 랜드마크 좌표계의 원점과 코끝 랜드마크 좌표 간 벡터의 방향 변화에 대응해 얼굴의 Pan 각도와 Tilt 각도가 변경된다. 얼굴의 Pan 각도와 Tilt 각도가 기설정된 전 범위 내에서 변할 경우, 해당 커서 위치는 각각 모니터 화면상 전 영역의 수평 좌표와 수직 좌표에 걸쳐 대응되도록 환산된다. 이러한 과정을 통해 결정된 커서 위치가 화면상에서 이동될 시, Face Mech 모델의 3D 랜드마크 추정의 불완전성에 기인해 커서에 불안정한 잔진동이 야기된다. 따라서 커서 위치들에 적응형 이동 평균 처리를 적용해 미세 떨림을 완화함으로써 커서 조작의 정교화와 안정화를 도모할 수 있다. 한편, 얼굴의 Roll 각도는 양쪽 눈꼬리 랜드마크 좌표들을 연결한 벡터를 이용해 계산한다. 다음으로 이렇게 구한 3개의 얼굴 각도들과 커서 위치, 그리고 양안 개폐 여부와 그 유지시간 등의 얼굴 정보를 프레임별로 버퍼에 저장한다. 이 버퍼의 얼굴 제스처들을 검사해 카메라 입력으로부터 기설정된 인터페이스 활성화 이벤트가 발생되는지를 판별한다. 이 인터페이스 활성화 이벤트가 발생되는 경우에 국한해 그 이후, 버퍼에 저장된 일련의 얼굴 정보에서 얼굴 제스처를 인식해 해당 사용자 이벤트를 발생시킨다. 예컨대, Roll 각도의 음과 양의 방향와 크기에 따라 대응된 상하 스크롤 이벤트를 발생시킨다. 그리고 얼굴 제스처 인식 과정에서 양안 동시 개폐 여부를 판별할 시, 양안 동시 개폐의 불연속성을 반복적으로 검사해 양안의 일시적 개폐 불일치를 차단한다. 이상과 같은 과정을 통해 사용자 이벤트가 발생되면, 관련 정보를 화면에 표시함과 동시에 운영체제는 이 사용자 이벤트에 대응하는 명령을 처리하도록 사용자 인터페이스를 제어한다. 이후 카메라 입력으로부터 인터페이스 비활성화 이벤트가 입력되면 얼굴 제스처 인터페이스 모드를 종료하고, 그렇지 않으면 카메라로부터 다음 프레임을 입력받는 과정을 반복적으로 수행한다.

2.3 프로그램 파일 구성 및 그 역할

제안된 얼굴 제스처 기반의 사용자 인터페이스는 객체 지향 프로그래밍 패러다임에 근거해 설계·구현되었다. ⟨Table 1⟩은 본 논문의 연구진이 구현한 얼굴 제스처 기반의 사용자 인터페이스 프로그램의 파일명들과 의존성 관계를 나타낸 것이다.

⟨Table 1⟩ File names and dependency relationships of the proposed facial gesture interface program

bynames	filenames	Sub-dependency modules
A	main.py	B, C, D
B	utilized_face_mesh.py	MediaPipe Face Mesh
C	face_gesture_processor.py	E, F, G, H, I, J
D	painter.py	-
E	face_info_per_frame.py	-
F	face_angle_stabilizer.py	E
G	cursor_position_calculator.py	-
H	buffer_for_face_info.py	E
I	buffer_for_cursor_position.py	-
J	gesture_producer.py	H, I

파일 A는 프로그램의 엔트리 포인트로, 최상위 계층에 해당하는 작업을 수행하는 코드를 포함하고 있다. 파일 B는 MediaPipe Face Mesh 모델을 상속받아 요구에 맞게 수정하는 Utilized_face_mesh 클래스가 구현된 파일이다. 파일 C는 E부터 J까지 6개의 하위 의존성 모듈을 활용해 얼굴 제스처를 처리하는 Face_gesture_processor 클래스가 구현된 것이다. 파일 D는 얼굴 제스처 정보에 따라 커서 위치를 화면에 표시하고 해당 사용자 이벤트에 대응하는 명령을 실행하는 Painter 클래스가 구현된 것이다. 파일 E는 프레임별 얼굴의 Pan 각도, Tilt 각도, Roll 각도, 양안 개폐 여부 및 그 유지시간 등의 얼굴 정보들을 계산하고 관리하는 Face_info_per_frame 클래스를 구현한 것이다. 파일 F는 얼굴의 안정적 각도 계산을 위해 얼굴 인터페이스 활성화 직후, 일정 시간(예컨대, 3초) 동안 프레임들의 코끝 랜드마크 좌표들을 합산해 평균한 값으로 3D 얼굴 랜드마크 좌표계의 각도 산출 기준점으로 삼음으로써 얼굴의 Pan 각도와 Tilt 각도를 안정화시키는 Face_angle_stabilizer 클래스가 구현된 것이다. 파일 G는 일련의 프레임 시퀀스 간에 걸쳐 얼굴의 Pan 각도, Tilt 각도가 변하면 그에 따라 가변된 커서 위치들을 계산한 후, 미세 떨림을 완화하기 위해 이 커서 위치들에 적응형 이동 평균을 적용해 안정

화시키는 Cursor_position_calculator 클래스를 구현한 것이다. 파일 H는 제한된 크기 내에서 프레임별 얼굴 정보(Face_info_per_frame 인스턴스)를 저장하기 위해 Buffer_for_face_info 클래스를 구현한 것이다. 파일 I는 제한된 크기 내에서 프레임별 커서 위치를 저장하는 Buffer_for_cursor_position 클래스를 구현한 것이고, 파일 J는 H와 I에 이 버퍼에 저장된 얼굴 정보에서 얼굴 제스처를 인식해 해당 사용자 이벤트를 발생시키는 Gesture_producer 클래스를 구현한 것이다[28].

3. 얼굴 제스처 인터페이스의 성능 개선

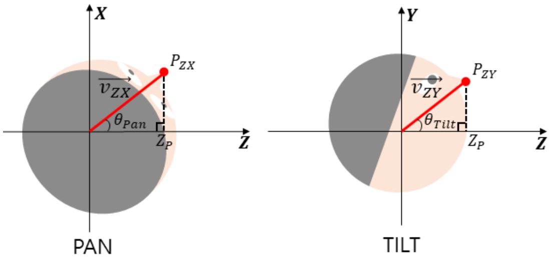
3.1 커서 위치 이동

커서 위치의 좌우 및 상하 이동은 각각 얼굴의 Pan 각도 및 Tilt 각도의 가변에 의해 일어난다. 얼굴의 Pan 각도는 커서를 X축 방향으로, 얼굴의 Tilt 각도는 커서를 Y축 방향으로 이동시킨다. 이때, Pan 방향 회전은 얼굴의 수평 방향 회전이고, Tilt 방향 회전은 얼굴의 수직 방향 회전을 의미한다. 본 논문의 연구진은 [Fig. 1]의 MediaPipe Face Mesh 모델의 랜드마크 번호 1인 3D 꼬끝 랜드마크 좌표를 이용해 얼굴의 Pan 각도와 Tilt 각도를 계산한다. 우선, 얼굴 제스처 인터페이스 모드가 활성화되자마자 최초 소정 시간 동안 사용자가 얼굴의 움직임을 억제하고 모니터 화면의 정중앙을 바라보도록 한 채 입력받은 프레임들의 코끝 랜드마크 좌표들을 누적 평균해 3D 얼굴 랜드마크 좌표계의 각도 산출 기준점을 구한다. 이후, 3D 얼굴 랜드마크 좌표계의 원점과 각도 산출 기준점을 연결하는 3차원 벡터를 구하고, 3D 얼굴 랜드마크 좌표계의 원점과 코끝 랜드마크 좌표를 연결하는 3차원 벡터를 구한 후, 이 두 3차원 벡터 간의 수평 및 수직 방향의 각도를 계산해 얼굴의 Pan 각도와 Tilt 각도를 구한다.

앞서 설명했듯이 원래 Face Mesh 모델의 3D 얼굴 랜드마크 좌표계의 원점은 별도로 존재하지만, 설명의 편의상 그 원점을 가상적으로 3D 두상의 중심으로 이동시킨 상태에서 원래 3D 얼굴 랜드마크 좌표계의 원점과 각도 산출 기준점을 연결하는 3차원 벡터에 그 좌표계의 Z축을 일치시킨 3D 가상 좌표계가 있다고 가정할 때, 3D 얼굴 랜드마크 좌표계의 원점과 코끝 랜드마크 좌표를 연결하는 3차원 벡터를 그 3D 가상 좌표계에 표시한 벡터를 벡터 $\vec{v}$라고 하자. 그리고 그 3D 가상 좌표계에 3D 코끝 랜드마크 좌표를 표시한 것을 좌표 P라고 하자.

[Fig. 3]은 3차원 벡터 $\vec{v}$와 3차원 좌표 P를 3D 가상 좌표계의 ZX 평면에 투영시켜 각각 2차원 벡터 $\overrightarrow{v_{ZX}}$와 2차원 좌표 P_{ZX}를 표시한 것이다. 3D 얼굴 랜드마크 좌표계의 원점과 각도 산출 기준점을 연결하는 3차원 벡터를 이 3D 가상 좌표계의 Z축과 일치시켰기 때문에 얼굴의 Pan 각도 θ_{Pan}은 식 (1)의 첫 번째 수식을 이용해 $\overrightarrow{v_{ZX}}$와 Z축 사이의 각도로 계산함으로써 쉽게 구할 수 있다. 식 (1)에서 Z_P는 2차원 좌표 P_{ZX}를 그 Z축에 투영시킨 값이다. 마찬가지로, 3차원 벡터 $\vec{v}$와 3차원 좌표 P를 3D 가상 좌표계의 ZY 평면에 투영시켜 각각 2차원 벡터 $\overrightarrow{v_{ZY}}$와 2차원 좌표 P_{ZY}를 표시한 것이다. 얼굴의 Tilt 각도 θ_{Tilt}는 식 (1)의 두 번째 수식을 이용해 $\overrightarrow{v_{ZX}}$와 Z축 사이의 각도를 계산함으로써 구할 수 있다. 여기서 Z_P는 2차원 좌표 P_{ZY}를 Z축에 투영시킨 값이다.

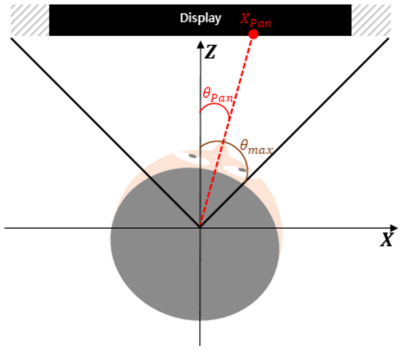
$$\theta_{Pan} = \cos^{-1}\left(\frac{Z_P}{\|\overrightarrow{v_{ZX}}\|}\right) \quad \because \cos(\theta_{Pan}) = \frac{Z_P}{\|\overrightarrow{v_{ZX}}\|} \qquad (1)$$
$$\theta_{Tilt} = \cos^{-1}\left(\frac{Z_P}{\|\overrightarrow{v_{ZY}}\|}\right) \quad \because \cos(\theta_{Tilt}) = \frac{Z_P}{\|\overrightarrow{v_{ZY}}\|}$$



[Fig. 3] An angle in the pan direction θ_{Pan} and an angle in the tilt direction θ_{Tilt}

사전에 화면의 가로 크기 $width$와 세로 크기 $height$, 얼굴의 Pan 각도와 Tilt 각도의 최대 범위 각도인 $\theta_{\max}$가 정해져 있으면, 식 (2)와 식 (3)과 같이 X_{Pan}과 dx 값을 계산할 수 있다. [Fig. 4]는 얼굴의 Pan 각도 θ_{Pan}에 대한 화면상의 X축 화소 위치값 X_{Pan}을 나타낸 것이다. 화면 영역의 좌우측에 빗금 영역은 사용성을 고려한 여유 영역(marginal zone)이다. X_{Pan}은 식 (2)와 같이 계산된다. 이때 dx는 식 (3)과 같이 사전에 정한 $\theta_{\max}$에 따라 결정된다. Y_{Pan}의 경우도 X_{Pan}과 유사한 계산 과

정으로 구할 수 있다. $\theta_{\max}$는 필요 시, Pan 방향과 Tilt 방향에 대해 각각 다른 각도로 설정할 수도 있다.



[Fig. 4] The angle of the pan direction θ_{Pan} and its correspondence on the screen

$$X_{Pan} = \left(\frac{\text{width}}{2} + \text{margin}\right) + \tan(\theta_{Pan}) \times dx \qquad (2)$$

$$dx = \frac{\text{width}}{2} \times \frac{1}{\tan(\theta_{\max})} \qquad (3)$$

3.2 커서 안정화

본 논문의 연구진이 선행 연구에서 제안한 얼굴 제스처 인터페이스[28]의 사용자 시나리오 시뮬레이션을 통해 커서 이동 시, 불안정한 미세 떨림이 발생함을 알 수 있었다. 이러한 현상을 사용자가 미숙련자일수록 심해지고 정밀한 커서 조작을 위해 긴장도를 높일수록 더 가속화됨을 알 수 있었다. 이러한 문제를 해결하기 위해 본 논문에서는 커서의 미세 떨림을 억제함으로써 커서 조작의 정교화와 안정화를 도모한다.

본 논문에서는 커서 위치를 산출하는 전 과정에 현재 커서 위치에 대한 적응형 이동 평균 처리를 적용해 미세 떨림을 완화한다. 이에 따라 사용자 명령으로 사용되는 모든 커서 위치는 최초 3개 혹은 4개 프레임을 제외하고는 누적된 이동 평균 처리가 적용된 값이다. 이를 '평균 커서 위치'라고 부르기로 한다. 제안된 방법에서는 커서 이동속도의 완급에 따라 현재 프레임의 커서 위치를 포함한 가장 최근 3개 프레임 혹은 4개 프레임의 평균 커서 위치들을 합산해 다시 평균한 값을 현재 프레임의 실제 커서 위치로 사용한다.

가령, 아직 평균 처리를 하지 않는 현재 커서 위치와 직전 평균 커서 위치 간의 화소 거리를 d라고 할 때, d가 10 미만($d < 10$)이면 현재 프레임의 커서 위치를 포함한 가장 최근 네 프레임의 평균 커서 위치들의 평균값을 현재 프레임의 실제 커서 위치로 사용하고, 그렇지 않으면 현재 프레임의 커서 위치를 포함한 가장 최근 세 프레임의 평균 커서 위치들의 평균값을 실제 커서 위치로 사용한다. 평균 계산 시, 사용하는 프레임 수를 이동 거리 d에 따라 가변하는 이유는 사용 프레임 수를 높일수록 커서의 미세 떨림은 개선되지만, 커서의 이동속도가 지연돼 반응성이 저하되는 문제가 발생하기 때문이다. 즉, 10화소 미만의 미세한 이동이 필요한 경우에는 이동속도보다 떨림 개선에 좀 더 치중하기 위해 최근 네 프레임을 활용하고, 10화소 이상의 이동이 필요한 경우에는 떨림 개선보다는 이동속도를 더 감안하기 위해 최근 세 프레임을 활용한다. 여기서, d의 임계값인 10화소는 실험적으로 구한 값이기에 개인 선호에 따라 변경가능한 값이다.

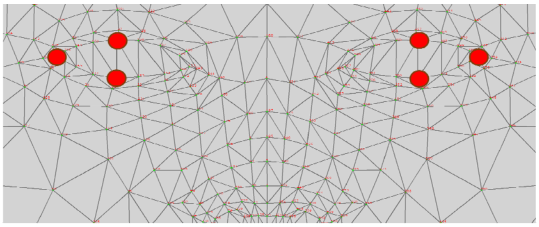
3.3 얼굴 제스처 인식

제안된 얼굴 제스처 인터페이스의 얼굴 제스처와 그에 대응하는 사용자 이벤트의 종류는 〈Table 2〉와 같다. 얼굴 제스처 판별에 사용되는 얼굴 정보는 현재 프레임과 과거 프레임들의 양안 개폐 여부 및 그 유지 시간과 얼굴 Roll 각도이다. 이를 위해 본 논문은 [Fig. 5]의 적색 원으로 나타낸 양안의 눈꼬리와 맞닿은 상하 안검(upper and lower eyelid) 두 지점으로 구성된 총 6개의 랜드마크들을 사용해 양안 개폐 여부 및 얼굴의 Roll 각도를

〈Table 2〉 Types of facial gestures and corresponding user events in the proposed facial gesture interface

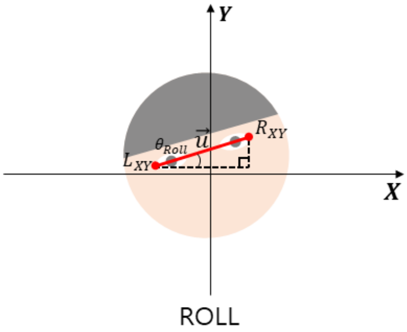
User's facial gestures	Operating conditions and holding times	User Events
After keeping both eyes closed for a certain period of time, they open.	More than 0.2 seconds but less than 3 seconds	Left click
Perform left-click gesture twice within a certain period of time.	Successive trial gap of less than 2 seconds	Double click (left)
After keeping one eye closed for a certain period of time, it opens.	More than 0.2 seconds but less than 3 seconds	Right click
Move the cursor after maintaining the right-click gesture for a certain period of time.	Exceed 1 second	Drag
Both eyes open during drag event.	-	Drop
Roll head to the right	Critical roll angle: -35°	Scroll up
Roll head to the left	Critical roll angle: +35°	Scroll down
After keeping both eyes closed for a certain period of time, they open.	Exceed 3 seconds	Enable/Disable interface

계산한다. [Fig. 5]는 제안된 얼굴 제스처 인식에서 사용된 MediaPipe Face Mesh 모델의 6개 얼굴 랜드마크들을 나타낸 것이다.



[Fig. 5] Six facial landmarks from MediaPipe Face Mesh model used in the proposed method

먼저, 양쪽 안구의 상하 안검 두 지점, 총 4개의 랜드마크가 양안 개폐 여부 판단에 사용되며 양안의 상 하 안검 Y축 위치값의 차이가 지정된 임계값 이하일 때, 해당 안구가 폐쇄된 것으로 판단한다. 이처럼 구한 양안의 개폐 여부로, 좌클릭 이벤트, 더블 좌클릭 이벤트, 우클릭 이벤트, 드래그 이벤트, 드롭 이벤트, 인터페이스 활성화 이벤트 및 인터페이스 비활성화 이벤트, 제스처 모드 전환 이벤트를 발생시킨다.



[Fig. 6] The angle of the roll direction θ_{Roll}

다음으로 3D 가상 좌표계에서 양안의 눈꼬리 랜드마크를 각각 L_{XY}, R_{XY}이라고 했을 때, 얼굴을 양 어깨 쪽으로 기울임에 따라 변하는 [Fig. 6]의 벡터 $\vec{u}$와 X축이 이루는 각도가 Roll 각도 계산에 사용된다. θ_{Roll}은 [Fig. 6]에서 식 (4)와 같이 구할 수 있다. 이후, θ_{Roll}의 절대값이 지정된 임계값 이상인지 판단해 상하 스크롤 이벤트를 발생시킨다.

$$\theta_{Roll} = \tan^{-1}\left(\frac{\overrightarrow{u_y}}{\overrightarrow{u_x}}\right) \quad \because \tan(\theta_{Roll}) = \frac{\overrightarrow{u_y}}{\overrightarrow{u_x}} \qquad (4)$$

3.4 양안 동시 개폐 오동작 개선

본 논문의 연구진은 실험 과정에서 양안 동시 개폐시, 개인차에 따라 좌우 안구가 근소한 차이로 불일치되게 개폐되는 경우가 종종 발생함을 발견하였다. 다시 말해서, 양안 동시 개폐 시, 전환 과정에서 좌우 안구가 근소한 차이로 불일치되게 개폐됨으로 인해 양안 개폐 제스처들의 인식률이 저감되는 문제가 있었다.

이러한 문제를 개선하기 위해 양안 개폐 인식 시, 양안이 모두 개방된 인식 시점을 기준으로, 최근 두 프레임을 검사해 양안이 모두 개방된 현재 프레임과 양안이 모두 폐쇄된 전전 프레임 사이에서 직전 프레임이 한쪽 안구만 개방된 것으로 인식되면, 양안 모두가 폐쇄된 것으로, 강제로 수정하는 과정을 반복적으로 수행한다. 즉, 양안 동시 개폐의 불연속성을 조사해 양안의 일시적 개폐 불일치 문제를 해결함으로써 해당 명령의 오작동을 개선한 것이다. 그 결과, 연산 부담을 상쇄할 만큼 양안 개폐 제스처의 인식률이 크게 상승하는 것을 확인할 수 있었다. 이를 통해 양안 개폐 모드를 사용하는 좌클릭, 더블 좌클릭, 인터페이스 활성화 및 비활성화, 제스처 모드 전환과 같은 이벤트 발생의 오작동을 현저하게 줄일 수 있었다.

4. 시뮬레이션 결과 및 고찰

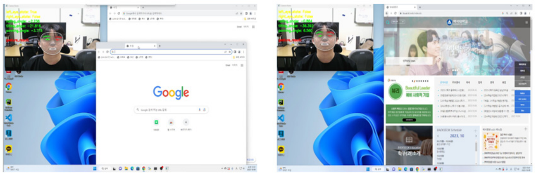
제안된 사용자 인터페이스의 사용성과 인식 성능을 평가하기 위해, Intel Core i7-12700(2.1GHz) CPU, DDR5 32GB RAM 데스크탑의 Windows 11 Pro 환경에서 NVIDIA RTX 3070 D6(8GB) GPU와 LogiTech C920 HD Pro 웹캠과 Python 3.8/MediaPipe 0.9.1.0/OpenCV4.6.0/CUDA 11.4/cuDNN 8.2.2를 이용해 제안된 방식에 대한 컴퓨터 시뮬레이션을 수행하였다.

4.1 사용자 시나리오 시뮬레이션

본 논문의 연구진은 사용자의 컴퓨터 환경에서 사용성을 검증하고 활용 사례를 보이기 위해 다양한 사용자 시나리오에 대해 시뮬레이션을 수행하였다. 웹 서핑, 게임, 비디오 재생, 문서 열람 등의 사용자 시나리오 시뮬레이션을 통해 본 연구진의 선행 연구[28]에 비해 그 사용성이 개선됐음을 확인할 수 있었다.

[Fig. 7(a)]는 사용자가 웹 브라우저의 탭을 옮기는 상

황으로 사용자의 우측 안구가 닫혀있는 상태에서 드래그 중인 장면을 예시한 것이다. [Fig. 7(b)]는 사용자가 뒤로 가기 버튼을 누르는 상황으로 좌클릭을 위해 좌우측 안구가 닫혀있는 화면을 예시한 것이다. 제안된 방법의 사용자 시나리오 시뮬레이션을 통해 본 연구진의 선행 연구[28]보다 세밀하고 안정적인 조작이 가능해졌음을 확인할 수 있었다.



(a) (b)

[Fig. 7] User scenario simulation using the proposed facial gesture interface

4.2 얼굴 제스처 인식률

앞서 소개한 사용자 시나리오 시뮬레이션 외에도 얼굴 제스처의 인식률을 측정하기 위해 10명의 실험자들을 대상으로 얼굴 제스처를 취하였다.

〈Table 3〉 Usability evaluation of the proposed facial gesture recognition

User	Left Click	Double Click	Right Click	Drag	Drop	Scroll Up	Scroll Down	Enable/ Disable	Total
1	30	30	29	29	30	30	30	30	238
2	30	30	26	28	30	30	30	28	232
3	30	30	30	30	30	30	30	28	238
4	30	30	30	30	30	30	30	26	236
5	30	30	27	30	30	30	30	28	237
6	30	30	29	30	30	30	30	28	237
7	30	30	30	30	30	30	30	30	240
8	30	29	30	30	30	30	30	26	235
8	30	30	30	29	30	30	30	29	238
10	30	30	30	30	30	30	30	28	238
Avg (%)	100	99.6	97.0	98.6	100	100	100	93.6	98.7

〈Table 3〉은 제안된 얼굴 제스처 인식의 사용성 평가를 나타낸 것이다. 본 연구진의 선행 연구[28]와 비교했을 때, 거의 모든 얼굴 제스처 인식률이 향상된 것을 보였고 그중에서도 좌클릭 제스처의 인식률이 90.6%에서 100%로 가장 많이 향상된 것을 확인할 수 있었다. 더블 클릭의 경우 91.6%에서 99.6%로, 우클릭은 95.3%에서 97.0%로, 인터페이스 활성화 및 비활성화은 89.3%에서 93.6%로 향상된 것을 확인할 수 있었다. 얼굴 제스처의 평균 인식률이 선행 연구[28]에서 95.8%였던 것에 비해 98.7%로 상향되었다. 그리고 초당 프레임 수(fps)는 30 프레임으로, 실시간 처리에 무리가 없음을 확인할 수 있었다.

5. 결론

본 논문에서는 MediaPipe Face Mesh를 이용한 얼굴 제스처 인터페이스의 우수성을 입증하기 위해 사용자 시나리오 시뮬레이션과 얼굴 제스처 인식률 실험 측면에서 제안된 방법과 본 연구진의 선행 연구[28] 간의 평가 결과를 비교·분석하였다.

제안된 얼굴 제스처 인터페이스는 얼굴 제스처 인식 과정에서 발생한 커서의 미세 떨림에 적응형 이동 평균 처리를 적용해 오동작을 개선할 수 있었다. 또한, 좌우 안구가 근소한 차이를 두고 불일치되게 개폐됨으로 인해 발생하는 얼굴 제스처 인식률 저하가 발생하는데, 양안 동시 개폐의 불연속성을 반복적으로 검사해 양안의 일시적 개폐 불일치를 차단함으로써 오작동을 개선할 수 있었다. 사용성 평가 실험에서 얼굴 제스처의 평균 인식률이 선행 연구[28]의 95.8%에 비해 98.7%로 상향되는 것을 확인하였다. 더불어 제안된 방법의 초당 프레임 수는 30프레임으로, 통상의 데스크탑에서 실시간 처리가 가능함을 확인할 수 있었다.

제안된 얼굴 제스처 인터페이스에서는 사용자의 얼굴 제스처를 이용해 자연스럽고 직관적인 방법으로 인터페이스를 제어할 수 있었다. 또한 기기와의 직접적인 접촉 없이 인터페이스를 제어할 수 있음에 따라 소부장 장비 조작 환경 등에서 그 활용 가능성이 높을 것으로 기대된다. 그러나 MediaPipe Face Mesh 모델의 3D 랜드마크 좌표는 입력 프레임 전체에 대한 깊이가 아니라 입력 프레임 중 얼굴에 국한된 영역 내 상대적 깊이를 제공하는 한계가 있다. 또한, 단안 카메라를 이용한 방식이기에 모델이 제공하는 깊이 값의 신뢰도가 낮은 태생적 한계가 있다. 따라서 양안 카메라를 이용하는 스테레오 정합 등을 도입해 입력 프레임에서 얼굴의 특정 지점에 대한 좀 더 정확한 깊이 정보를 추출해 활용할 수 있는 추가적인 연구를 진행할 필요가 있다.

REFERENCES

[1] F. Karray, et al, "Human-Computer Interaction: Overview on State of the Art," International Journal on Smart Sensing and Intelligent Systems, Vol.1, No.1,

pp.137-159, 2008.

[2] T. H. Tsai, C.C. Huang, and K.L. Zhang, "Design of Hand Gesture Recognition System for Human-computer Interaction," Multimedia Tools and Applications, Vol.79, No.9-10, pp.5989-6007, 2020.

[3] G. Kim and J. Baek, "Real-Time Hand Gesture Recognition Based on Deep Learning," Journal of Korea Multimedia Society, Vol.22, No.4, pp.424-431, 2019.

[4] B. Kumar, R. K. Bedi, and S. K. Gupta, "Facial Gesture Recognition for Emotion Detection: A Review of Methods and Advancements," Handbook of Research on AI-Based Technologies and Applications in the Era of the Metaverse, pp.542-358, 2023.

[5] Q. Gao, Y. Chen, Z. Ju and Y. Liang, "Dynamic Hand Gesture Recognition Based on 3D Hand Pose Estimation for Human-robot Interaction," IEEE Sensors Journal, pp.17421-17430, 2021.

[6] H. Kaur and J. Rani, "A Review: Study of Various Techniques of Hand Gesture Recognition," Proceedings of 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems, pp.1-5, 2016.

[7] Y. Li, J. Huang, F. Tian, H. Wang, and G. Dai, "Gesture Interaction in Virtual Reality," Virtual Reality and Intelligent Hardware, pp.84-112, 2019.

[8] C. A. Cruz, N. Tatsuya, M. Ichihara, F. Shibata, and A. Kimura, "Sequential Eyelid Gestures for User Interfaces in VR," Proceedings of 2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, Mar. 2023.

[9] A. Shimada, T. Yamashita and R. Taniguchi, "Hand Gesture Based TV Control System— Towards Both User-Machine-friendly Gesture Applications," Proceedings of The 19th Korea-Japan Joint Workshop on Frontiers of Computer Vision, pp.121-126, 2013.

[10] H. Stern, Y. Edan, M. Gillam, C. Feied, M. Smith, J. Handler, et al., "A Real-Time Hand Gesture Interface for Medical Visualization Applications," Applications of Soft Computing, Vol.36, pp.153-162, Springer, 2006.

[11] G. Pala, J.B. Jethwani, S.S. Kumbhar, and S. D. Patil, "Machine Learning-based Hand Sign Recognition," Proceedings of 2021 International Conference on Artificial Intelligence and Smart Systems, pp.356-363, 2021.

[12] M. Iskandar, K. Bingi, B. R. Prusty, M. Omar, and R. Ibrahim, "Artificial Intelligence-based Human Gesture Tracking Control Techniques of Tello EDU Quadrotor Drone," Proceedings of International Conference on Green Energy, Computing and Intelligent Technology, Jul. 2023.

[13] MANO, https://mano.is.tue.mpg.de (accessed Dec. 10, 2023).

[14] N. Qian, J. Wang, F. Mueller, F. Bernard, V. Golyanik, C. Theobalt, et al., "HTML: A Parametric Hand Texture Model for 3D Hand Reconstruction and Personalization," Proceedings of the European Conference on Computer Vision, pp.54-71, 2020.

[15] S. An, X. Zhang, D. Wei, H. Zhu, J. Yang, K. A. Tsintotas, et al., "Fast Hand: Fast Monocular Hand Pose Estimation on Embedded Systems," Journal of Systems Architecture, Vol.122, 2022.

[16] Z. Fan, A. Spurr, M. Kocabas, S. Tang, M.J. Black, O. Hilliges, et al., "Learning to Disambiguate Strongly Interacting Hands via Probabilistic Per-pixel Part Segmentation," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.1-10, 2021.

[17] M. Jang and W. Lee, "Implementation of User Gesture Recognition System for Manipulating a Floating Hologram Character," The Journal of the Institute of Internet, Broadcasting and Communication, Vol.19, No.2, pp.143-149, Feb. 2019.

[18] L. Chen, S.Y. Lin, Y. Xie, Y.Y. Lin, and X. Xie, "MVHM: A Large-Scale multi-View Hand Mesh Benchmark for Accurate 3D Hand Pose Estimation," Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, pp.836-845, 2021.

[19] Google MediaPipe, https://developers.google.com/mediapipe (accessed Dec. 10, 2023).

[20] K. Heo, B. Song, and J. Hong, "Hierarchical Hand Pose Model for Hand Expression Recognition," Journal of the Korea Institute of Information and Communication Engineering, Vol.25, No.10, pp.1323-1329, 2021.

[21] K. Heo, M. Kim, B. Song, and B. Shin, "Hand Expression Recognition for Virtual Blackboard," Journal of the Korea Institute of Information and Communication Engineering, Vol.25, No.12, pp.1770-1776, 2021.

[22] B. Song, S. Lee, H. Choi and S. Kim, "Design and Implementation of a Stereoscopic Image Control System Based on User Hand Gesture Recognition," Journal of the Korea Institute of Information and Communication Engineering, Vol.26, No.3, pp.396-402, 2022.

[23] R. Song, Y. Hong, and N. Kwak, "User Interface Using Hand Gesture Recognition Based on MediaPipe Hands Model," Journal of Korea Multimedia Society, Vol.26, No.2, pp.101-113, Feb. 2023.

[24] J. Prameela, K. V. Lakshmi, K. Manju, and M. S. Devi, "Mouse Handling Using Facial Gesture," International Research Journal of Modernization in Engineering Technology and Science, Vol.04, No.5, pp.468-475, May 2022.

[25] S. Sreeni, M. Sabeel, E. S. Kumar, V. H. Vardhan, and K. Chandrakala, "Mouse Cursor Control Using Facial Movements-An HCI Application," International Journal of Techno-Engineering, pp.270-274, Vol.15, No.2, Apr. 2023.

[26] Z. Sharifisoraki, M. Amini, and S. Rajan, "A Novel Face Recognition Using Specific Values from Deep Neural Network-based Landmarks," Proceedings of 2023 IEEE International Conference on Consumer Electronics, Jan. 2023.

[27] S. Thino, "Developing a Program to Detect Face Direction and the State of Partially Closed Eyes," Thesis of Master's Degree, Naresuan University, Oct. 2023.

[28] J. Mok and N. Kwak, "Facial Gesture-based User Interface Using MediaPipe Face Mesh," Proceedings of 2023 Summer Annual Conference of The Institute of Electronics and Information Engineers, pp.1407-1411, Jun. 2023.

목 진 왕(Jinwang Mok) [준회원]

- 2023년 8월 : 백석대학교 컴퓨터공학부 (공학사)
- 2023년 9월 ~ 현재 : 광주과학기술원 AI대학원 석박사통합과정

〈관심분야〉
제스처 기반 UI/UX, Human Computer Interface(HCI), 클라우드 컴퓨팅, 자연어 처리

곽 노 윤(Noyoon Kwak) [종신회원]

- 1994년 2월 : 한국항공대학교 항공전자공학과 (공학사)
- 1996년 2월 : 한국항공대학교 대학원 항공전자공학과 (공학석사)
- 2000년 2월 : 한국항공대학교 대학원 항공전자공학과 (공학박사)
- 2000년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 교수

〈관심분야〉
딥러닝 기반 영상처리 및 컴퓨터비전, 얼굴 및 시선 인식, 객체 트래킹, 3D 재구성, 제스처 기반 UI/UX, 인공지능