MediaPipe 모델을 이용한 손-얼굴 혼성 제스처 인터페이스에 관한 연구

곽노윤*
백석대학교 컴퓨터공학부 교수

A Study on Hand-Face Hybrid Gesture Interface Using MediaPipe Models

Noyoon Kwak*
Professor, Division of Computer Engineering, Baekseok University

요 약 본 논문은 MediaPipe Hands 모델과 MediaPipe의 Face Mesh 모델을 이용해 얼굴 제스처와 손 제스처를 인식한 후, 이들을 결합한 MediaPipe 기반의 혼성 제스처 인터페이스에 관한 것이다. 먼저, 카메라 입력 프레임들에서 개별적으로 MediaPipe Hands 모델의 3D 손 랜드마크들과 MediaPipe Face Mesh 모델의 3D 얼굴 랜드마크들을 검출해 손과 얼굴의 유무를 판별한 후, 얼굴 커서 위치와 얼굴 제스처, 손 커서 위치와 손 제스처를 인식한다. 이후 이들을 사용자 친화적으로 혼용해 손-얼굴 혼성 제스처 기반의 사용자 인터페이스를 구현한다. 제안된 MediaPipe 기반의 손-얼굴 혼성 제스처 인터페이스는 손과 얼굴 중 어느 하나로 제스처 모드가 설정되지만, 모드 전환을 위한 추가 제스처 없이 얼굴과 손을 자유롭게 사용해 인터페이스를 제어할 수 있는 것이 장점이다. 또한 윈도우즈 환경에서 소프트웨어 조작 실험을 통해 제안된 손-얼굴 혼성 제스처 인터페이스의 실용성과 유용성을 확인할 수 있었다.

주제어 : 혼성 제스처 인터페이스, 얼굴 제스처 인식, 손 제스처 인식, MediaPipe, NUI

Abstract This paper describes a hybrid gesture interface based on MediaPipe that recognizes facial gestures and hand gestures using the MediaPipe Hands model and MediaPipe Face Mesh model, and then combines them. First, the presence of hands and faces is determined by individually detecting 3D hand landmarks of MediaPipe Hands model and 3D face landmarks of MediaPipe Face Mesh model from camera input frames, and then the face cursor position and face gestures, as well as the hand cursor position and hand gestures are recognized. Then, these are mixed in a user-friendly way to implement a user interface based on hand-face hybrid gestures. The proposed hand-face hybrid gesture interface based on MediaPipe has the advantage that the gesture mode is set to either the hand or the face, but the interface can be controlled freely using the face and hands without additional gestures for mode switching. In addition, the practicality and usefulness of the proposed hand-face hybrid gesture interface were confirmed through software operation experiments in Windows environment.

Key Words : Hybrid Gesture Interface, Facial Gesture Recognition, Hand Gesture Recognition, MediaPipe, NUI

본 논문은 2024년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 연구과제(2021RIS-004)로 수행되었음.
*교신저자 : 곽노윤(nykwak@bu.ac.kr)
접수일 2024년 09월 20일 수정일 2024년 10월 01일 심사완료일 2024년 10월 20일

1. 서론

HCI(Human-Computer Interaction) 기술은 센서와 인공지능, 그리고 CPU, GPU, 메모리 등의 급격한 발전 덕분에 인간과 컴퓨터 간의 자연스러운 소통을 강화하기 위해 꾸준히 혁신을 이어오고 있다. 특히, 음성, 시선, 표정, 제스처, 터치 외에도 근전도, 심전도, 뇌파, 맥파, 눈전위도와 같은 생체 신호를 활용해 디지털 기기를 제어하는 새로운 NUI(Natural User Interface) 기술이 활발히 연구되고 있다[1]. NUI 기술은 사람과 기계 사이의 물리적 장벽을 허물고, 직관적이고 자연스러운 사용자 경험을 제공하는 것을 목표로 하며, 사용자의 인지적 또는 신체적 상태와 상황을 반영해 동작하는 것이 특징이다[1-3].

사람의 제스처 인식 기술은 NUI의 핵심적인 연구 분야로, 1990년대부터 인간-로봇 상호작용[4], 3D 게임 인터페이스[5], 가상현실[6], 가전기기 상호작용[7], 의료적 자세 교정과 운동량 측정[8], 수화 인식[9], 드론 제어[10], 실감 미디어 분야에서 MANO[11], Fast Hand[12], DIGIT[13], 플로팅 홀로그램 캐릭터 제어[14], MVHM[15] 등과 같이 그 분야가 넓고 날로 확장되고 있다. 구글의 MediaPipe[16]가 제스처 인식 분야에서 크로스 플랫폼 프레임워크의 중심으로 부상하면서 제스처 인식 기술을 더 쉽게 구현할 수 있도록 돕고 있으며, MediaPipe Hands 모델을 기반으로 한 손 제스처 인터페이스 기술이 다양한 응용 분야에서 활용되고 있다[17-19]. 그러나 손 제스처는 장시간 사용 시 피로가 쌓이거나 양손이 자유롭지 않은 상황에서 제약이 발생하는 한계가 있다. 이에 비해 얼굴 제스처는 손을 사용할 필요가 없고, 시선과 얼굴의 움직임을 통해 더 직관적으로 상호작용할 수 있다는 장점이 있어, 최근 MediaPipe Face Mesh 모델을 활용한 얼굴 제스처 인식 기술[20-24]이 주목받고 있다.

본 논문의 저자도 최근에 MediaPipe Hands 모델 기반의 손 제스처 인터페이스 기술[19]과 MediaPipe Face Mesh 기반의 얼굴 제스처 인터페이스 기술[24]을 연이어 제안하였다. 또한 두 방식 모두 각각의 장단점이 존재함에 따라 이를 상호 결합한 혼성 제스처 인터페이스[25,26]를 후속으로 발표한 바 있는데, 본 논문은 이를 확대 재구성한, 손-얼굴 혼성 제스처 기반의 사용자 인터페이스를 제안함에 그 목적이 있다.

제안된 손-얼굴 혼성 제스처 인터페이스의 가장 큰 특징은 얼굴과 손 제스처 간의 자연스러운 모드 전환이 가능하다는 점이다. 사용자는 별도의 제스처 전환 동작 없이도 손과 얼굴을 자유롭게 활용할 수 있어 사용자 피로를 줄이고 편의성을 높일 수 있다. 얼굴 제스처로 인터페이스를 제어하다가 손 제스처로 전환하려면 단순히 손을 카메라 프레임 내에 위치시켜 원하는 제스처를 취하면 되고, 반대로 손 제스처에서 얼굴 제스처로 전환할 때는 손을 카메라 프레임 밖으로 이동시키기만 하면 된다. 이를 통해 좌클릭, 우클릭, 스크롤, 음량 조절, 드래그 앤 드롭 등의 다양한 기능을 직관적이고 자연스럽게 수행할 수 있다[25].

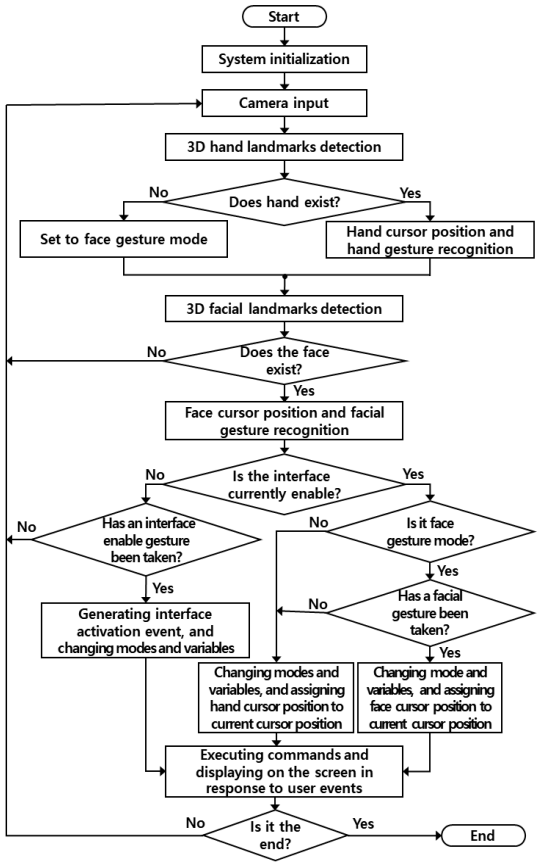
본 논문의 2장에서는 제안된 손-얼굴 혼성 제스처 인터페이스의 동작을 개괄적으로 설명하고, 3장에서는 MediaPipe Face Mesh 기반의 얼굴 제스처 인식 기술[24]을 소개하고, 4장에서는 MediaPipe Hands 기반의 손 제스처 인식 기술[19,25]을 다룬다. 5장에서는 시뮬레이션 결과 및 고찰을 다루고, 마지막으로 6장에서는 연구의 결론과 향후 연구 방향에 대해 논한다.

2. 제안된 손-얼굴 혼성 제스처 인터페이스

[Fig. 1]은 제안된 손-얼굴 혼성 제스처 인터페이스의 전체적인 순서도를 나타낸 것이다.

본 논문에서는 MediaPipe Face Mesh 모델과 MediaPipe Hands 모델을 이용해 카메라 입력 프레임에서 커서 위치와 얼굴 제스처 및 손 제스처를 인식한다. 이후 이를 사용자 친화적으로 혼용해 손-얼굴 혼성 제스처 기반의 사용자 인터페이스를 구현한다. 모든 카메라 입력 프레임 내에는 얼굴이 있어야만 혼성 제스처 인터페이스 제어가 가능하고, 얼굴과 손이 동시에 입력 프레임 내에 위치한 상태에서는 제스처 모드와 상관 없이 얼굴 커서 위치와 얼굴 제스처, 그리고 손 커서 위치와 손 제스처 모두를 인식한다. 이후, 해당 제스처 모드에 대응하는 커서 위치와 제스처를 사용해 혼성 사용자 인터페이스를 제어한다. 하지만 제스처 모드가 얼굴 제스처 모드인 상황에서 기등록된 어떤 얼굴 제스처도 검출되지 않으면, 별도의 제스처 모드 전환 명령이 없어도 손 제스처를 이용해 인터페이스를 즉시 제어할 수 있는 것이 특징이다[25].

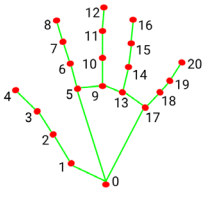
[Fig. 1]을 참고해 2.1절, 2.2절 및 2.3절에서 제안된 손-얼굴 혼성 제스처 인터페이스의 전체 동작 과정을 세부적으로 설명하기로 한다.



[Fig. 1] Overall flowchart of the proposed hand-face hybrid gesture interface

2.1 시스템 초기화 및 손-얼굴 제스처 인식 과정

우선, 본 논문의 손-얼굴 혼성 제스처 인터페이스 시스템은 [Fig. 1]과 같이, 현재 커서 위치를 화면 수평 중심값과 수직 중심값으로 초기화하고 제스처 모드를 얼굴 제스처 모드로 초기화하며 인터페이스 활성화 플래그를 비활성화값(disable)으로 초기화하고 손 유무 플래그를 부정값(false)으로 초기화한다.

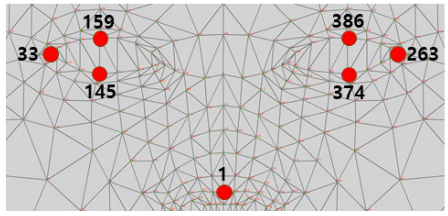


0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

[Fig. 2] 21 hand landmarks used in the proposed hand gesture recognition[2]

이후, 일련의 카메라 입력 시퀀스가 들어올 경우, MediaPipe Hands 모델을 이용해 카메라 입력 프레임에서 [Fig. 2]의 3D 손 랜드마크 검출을 시도한다. 3D 손 랜드마크가 검출되면, 손이 있다고 판단해 손 유무 플래그를 긍정값(true)으로 설정하고, 제스처 모드를 손 제스처 모드로 설정한 상태에서 MediaPipe Hands 모델 기반의 손 제스처 인식 기술[19,25]을 활용해 화면상 손 커서 위치와 손 제스처를 인식한다. 반면에 3D 손 랜드마크가 검출되지 않으면, 제스처 모드를 얼굴 제스처 모드로 전환한다. 이렇게 하면 이전 프레임까지 손으로 사용자 인터페이스를 제어하던 상황에서 갑자기 현재 프레임에서 손이 사라지더라도 즉시 인터페이스 제어권을 얼굴 제스처 모드로 전환시킬 수 있다.

연이어, MediaPipe Face Mesh 모델을 이용해 상기 동일한 카메라 입력 프레임에서 [Fig. 3]의 3D 얼굴 랜드마크 검출을 시도해 3D 얼굴 랜드마크가 검출되지 않으면, 카메라로부터 다음 프레임을 입력받는 단계로 되돌아간다. 이것은 각각의 입력 프레임에서 얼굴이 있어야만 혼성 제스처 인터페이스 제어가 가능함을 의미한다. 반면에 3D 얼굴 랜드마크가 검출되면, 얼굴이 있다고 판단해 얼굴 유무 플래그를 긍정값으로 설정하고 MediaPipe Face Mesh 모델 기반의 얼굴 제스처 인식 기술[24]을 활용해 화면상 얼굴 커서 위치와 얼굴 제스처를 인식한다.



[Fig. 3] 7 facial landmarks used in the proposed facial gesture recognition[3]

2.2 인터페이스 활성화와 손-얼굴 제스처 처리

[Fig. 1]의 인터페이스 활성화 판단 단계에서는 인터페이스 활성화 플래그가 이미 활성화 상태이고, 제스처 모드가 얼굴 제스처 모드인 경우, 상기 현재 커서 위치에 상기 얼굴 커서 위치값을 대입하고, 이 조건하에서 손 유무 플래그와 손 제스처 변수를 검사해 손이 존재하고 손 제스처의 유형이 '제스처 모드 전환' 제스처이면, 제스처 모드에 손 제스처 모드를 대입하고 얼굴 제스처 변수에 '제스처 모드 전환' 제스처를 대입한다. 그리고, 얼굴 제스처 변수를 검사해 얼굴 제스처가 '인터페이스 비활성

화' 제스처이면, 인터페이스 활성화 플래그를 비활성화 값으로 설정한다.

인터페이스 활성화 플래그가 인터페이스 활성화 상태이고, 제스처 모드가 얼굴 제스처 모드인 조건하에서, 기 등록된 어떤 얼굴 제스처도 검출되지 않은 상태인데, 손이 존재하면 손 제스처 변수값을 얼굴 제스처 변수로 복사하고, 손 제스처를 검사해 손 모양이 열린 가위 모양, 좌클릭 모양, 우클릭 모양, 닫힌 가위 모양 중 어느 하나이면 현재 커서 위치에 직전 손 커서 위치를 대입한 다음에 이상의 과정에서 정해진 현재 커서 위치에서 얼굴 제스처의 유형에 대응하는 사용자 이벤트를 발생시켜 해당 명령을 수행하도록 사용자 인터페이스를 제어한 후, 카메라로부터 다음 프레임을 입력받는 단계로 되돌아간다.

만일 인터페이스 활성화 플래그가 인터페이스 활성화 상태이고, 제스처 모드가 얼굴 제스처 모드가 아니라 손 제스처 모드인 조건인 경우, 현재 커서 위치에 상기 손 커서 위치값을 대입하고, 이 조건하에서 얼굴 제스처 변수를 검사해 얼굴 제스처의 유형이 제스처 모드 전환이면, 제스처 모드를 얼굴 제스처 모드로 변경하고 손 제스처 변수에 '제스처 모드 전환' 제스처를 대입한다. 그리고, 손 제스처 변수를 검사해 손 제스처가 '인터페이스 비활성화' 제스처이면, 인터페이스 활성화 플래그를 비활성화값으로 설정한다. 이상의 과정에서 정해진 현재 커서 위치에서 손 제스처의 유형에 대응하는 사용자 이벤트를 발생시켜 해당 명령을 수행하도록 사용자 인터페이스를 제어한 후, 카메라로부터 다음 프레임을 입력받는 단계로 되돌아간다.

그리고, 혼성 제스처 인터페이스가 활성화 상태에서 사용자 이벤트를 처리하던 중 인터페이스 비활성화 이벤트가 입력돼 비활성화 상태로 전환되면 혼성 제스터 인터페이스의 전체 동작이 종료된다.

2.3 인터페이스 비활성화와 손-얼굴 제스처 처리

[Fig. 1]의 인터페이스 활성화 판단 단계에서는 인터페이스 활성화 플래그를 검사해 비활성화 상태인 경우, 얼굴 제스처의 유형이 '인터페이스 활성화' 제스처이면 제스처 모드에 얼굴 제스처 모드를 대입하고 인터페이스 활성화 플래그를 활성화값(enable)으로 설정한 다음에 현재 커서 위치에서 얼굴 제스처의 유형('인터페이스 활성화' 제스처)에 대응하는 사용자 이벤트를 발생시켜 해당 명령을 수행하도록 사용자 인터페이스를 제어한 후, 카메라로부터 다음 프레임을 입력받는 단계로 되돌아간다.

만약 얼굴 제스처의 유형이 인터페이스 활성화 제스처가 아닌 조건하에서, 손 유무 플래그와 손 제스처 변수를 검사해 손이 존재하고 손 제스처의 유형이 인터페이스 활성화 제스처이면, 제스처 모드에 손 제스처 모드를 대입하고 인터페이스 활성화 플래그를 활성화값(enable)으로 설정한 다음에 현재 커서 위치에서 손 제스처의 유형('인터페이스 활성화' 제스처)에 대응하는 사용자 이벤트를 발생시켜 해당 명령을 수행하도록 사용자 인터페이스를 제어한 후, 카메라로부터 다음 프레임을 입력받는 단계로 되돌아간다.

손과 얼굴 양쪽으로부터 동시에 인터페이스 활성화 제스처가 검출되더라도 이렇게 얼굴의 인터페이스 활성화 제스처를 먼저 처리하면 둘 중 얼굴이 제어의 우선권을 갖게 된다.

한편, 인터페이스 활성화 플래그가 비활성화인 조건하에서, 손과 얼굴 모두로부터 인터페이스 활성화 제스처가 검출되지 않으면, 현재 커서 위치에서는 어떤 사용자 명령도 수행하지 않은 상태에서 카메라로부터 다음 프레임을 입력받는 단계로 되돌아간다.

3. Face Mesh 기반의 얼굴 제스처 인식[24]

본 논문에서는 Face Mesh 모델을 이용해 사용자의 얼굴 제스처를 인식한다. 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. 3]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[24].

3.1 커서 위치 산출

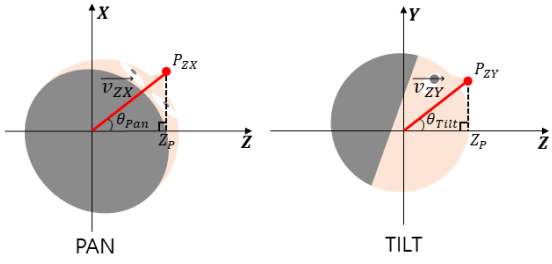
얼굴 제스처를 이용한 커서 위치의 좌우 및 상하 이동은 각각 얼굴의 Pan 각도 및 Tilt 각도의 변경을 감지해 산출한다. 얼굴의 Pan 각도는 커서를 X축 방향으로, 얼

굴의 Tilt 각도는 커서를 Y축 방향으로 이동시킨다. 이때, Pan 방향 회전은 얼굴의 수평 방향 회전이고, Tilt 방향 회전은 얼굴의 수직 방향 회전을 의미한다. [Fig. 3]의 MediaPipe Face Mesh 모델의 랜드마크 번호 1인 3D 꼬끝 랜드마크 좌표를 이용해 얼굴의 Pan 각도와 Tilt 각도를 계산한다.

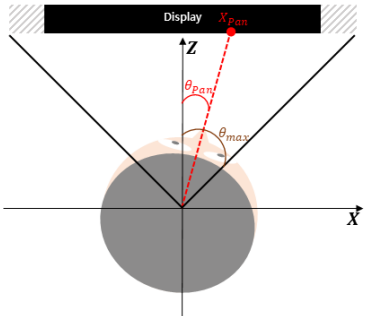
우선, 얼굴 제스처 인터페이스 모드가 활성화되자마자 최초 소정 시간 동안 사용자가 얼굴의 움직임을 억제하고 모니터 화면의 정중앙을 바라보도록 한 채 입력받은 프레임들의 코끝 랜드마크 좌표들을 누적 평균해 3D 얼굴 랜드마크 좌표계의 각도 산출 기준점을 구한다. 이후, 3D 얼굴 랜드마크 좌표계의 원점과 각도 산출 기준점을 연결하는 3차원 벡터를 구하고, 3D 얼굴 랜드마크 좌표계의 원점과 코끝 랜드마크 좌표를 연결하는 3차원 벡터를 구한 후, 이 두 3차원 벡터 간의 수평 및 수직 방향의 각도를 계산해 얼굴의 Pan 각도와 Tilt 각도를 구한다[24].

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. 4]는 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[24].

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



[Fig. 4] An angle in the pan direction θ_{Pan} and an angle in the tilt direction θ_{Tilt}



[Fig. 5] The angle of the pan direction θ_{Pan} and its correspondence on the screen

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

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

화면의 가로 크기 $width$와 세로 크기 $height$, 얼굴의 Pan 각도와 Tilt 각도의 최대 범위 각도인 $\theta_{\max}$가 미리 정해져 있으면, 식 (2)와 식 (3)과 같이 X_{Pan}과 dx 값을 계산할 수 있다. [Fig. 5]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 과정으로 구할 수 있다. $\theta_{\max}$는 필요 시, Pan 방향과 Tilt 방향에 대해 각각 다른 각도로 설정할 수도 있다[24].

3.2 얼굴 제스처 인식

제안된 얼굴 제스처 인터페이스의 얼굴 제스처와 사용자 이벤트의 종류는 〈Table 1〉과 같다.

〈Table 1〉 Types of facial gestures and corresponding user events in the proposed facial gesture interface

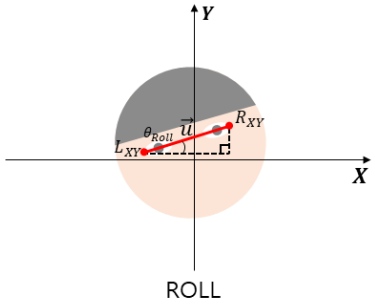
User's facial gestures	Operating conditions and holding times	User Events
After keeping both eyes closed for a certain period of time, they open.	More than 0.2 seconds but less than 3 seconds	Left click
Perform left-click gesture twice within a certain period of time.	Successive trial gap of less than 2 seconds	Double click
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. (Gesture mode switching: When used in hand gesture mode, it switches to face gesture mode, and when used in hand gesture mode, it switches to face gesture mode.)	Exceed 3 seconds	(1) Interface Enable (2) Interface disable (3) Gesture mode switching

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. 3]의 적색 원으로 나타낸 맞닿은 상하 안검(upper and lower eyelid) 두 지점과 양안의 눈꼬리로 구성된 총 6개의 랜드마크들을 사용해 양안 개폐 여부 및 얼굴의 Roll 각도를 계산한다. [Fig. 3]은 제안된 얼굴 제스처 인식에서 사용된 MediaPipe Face Mesh 모델의 7개 얼굴 랜드마크들을 나타낸 것이다[24].

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, 〈Table 1〉과 같이 좌클릭(left click) 이벤트, 더블 클릭(double click) 이벤트, 우클릭(right click) 이벤트, 드래그(drag) 이벤트, 드롭(drop) 이벤트, 인터페이스 활성화(interface enable) 이벤트 및 인터페이스 비활성화(interface disable) 이벤트, 제스처 모드 전환(gesture mode switching) 이벤트를 발생시킨다[24]. 제스처 모드 전환 이벤트는 손 제스처 모드에서 사용 시, 얼굴 제스처 모드로 전환되고 손 제스처 모드에서 제스처 모드 전환 이벤트가 발생되면 얼굴 제스처 모드로 전환된다.

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[24].

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



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

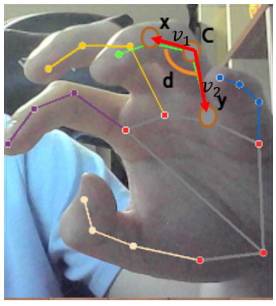
4. Hands 모델 기반의 얼굴 제스처 인식[19]

본 논문에서는 MediaPipe Hands 모델에서 제공하는 [Fig. 2]의 21개 랜드마크들을 사용해 손 자세 모델을 구축한다. MediaPipe Hands 모델은 랜드마크들을 3차원 좌표로 추출하며, 추출된 랜드마크 좌표들을 2차원 영상 좌표로 변환하고 손 영상 위에 겹쳐 표시하는 기능을 제공한다. 추출된 랜드마크들은 손 자세 인식에 바로 사용될 수는 없으므로 랜드마크들의 상대적 위치를 통해 손 자세를 구성하는 손가락 자세를 정의하고, 이 손가락 자세를 통해 손 자세를 정의한다. 이러한 계층적인 손 자

세 표현 방법을 통해 다양한 손 자세를 추상적으로 정의할 수 있으며, 추정된 손 자세를 동적 모델과 함께 사용하면 완전한 손 제스처 인식 모델을 구성할 수 있다[19].

4.1 손 모양 인식

손가락의 펼침(stretch)과 굽힘(bend) 여부를 판단하기 위한 가장 단순한 방법은 [Fig. 2]에서 특정 손가락 F_i일 때, F_i의 끝부분인 TIP과 첫 번째 관절인 DIP(distal interphalangeal joint)의 좌표값을 비교하는 방법이다. 하지만 단순히 TIP과 DIP을 비교해 손가락이 펼침 여부를 탐지하는 알고리즘은 손을 아래 방향으로 돌려 손 끝 부분 TIP이 바닥을 향하면 인식 오류가 발생하는 단점이 있다[19].



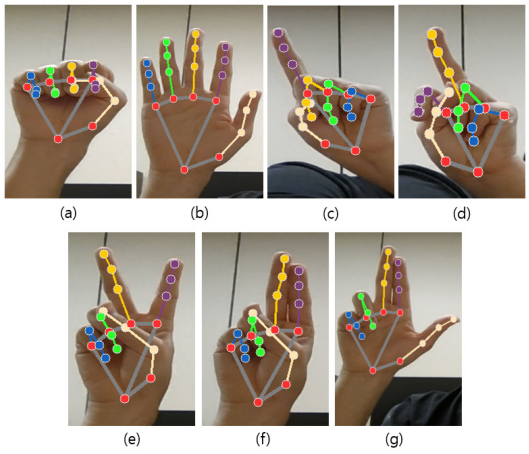
[Fig. 7] Determining the bending of the fingers using two vectors

본 논문에서는 손가락의 관절 각도를 이용해 손가락의 펼침 여부를 탐지하는 방법[17]을 실시간 처리가 용이하도록 변형해 사용하고 있다. [Fig. 7]와 같이 손가락의 특정 랜드마크를 C로 지정하고, C를 기준으로 연결된 위, 아래 랜드마크를 각각 x, y로 지정한다. 이후 C에서 x로 향하는 3차원 벡터 $\vec{v_1}$과 C에서 y로 향하는 3차원 벡터 $\vec{v_2}$를 구한다. 이후, 식 (5)와 같이 벡터의 내적을 이용해 두 벡터의 각도 θ_d를 구한다.

$$\theta_d = \cos^{-1}\left(\frac{x_1x_2+y_1y_2+z_1z_2}{\|\vec{v_1}\| \times \|\vec{v_2}\|}\right) \quad (5)$$

$$\because \vec{v_1} = (x_1,y_1,z_1),\ \vec{v_2} = (x_2,y_2,z_2)$$

이후, 각 손가락 관절의 각도 변화를 참고해 검지(index finger), 중지(middle finger), 약지(ring finger) 및 소지(pinky)는 150°보다 작으면 손가락의 굽힘으로 판단하고 엄지(thumb)의 경우에는 165°보다 작으면 굽힘으로 판정한다[25].



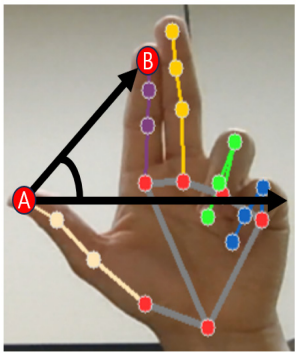
[Fig. 8] Hand shapes used in the proposed hand gesture recognition

손 모양을 인식하기 위해서는 각 손가락들의 맞닿음과 벌림, 그리고 각 손가락의 펼침과 굽힘, 중지와 검지가 이루는 각도의 조합으로 통해 [Fig. 8]의 손 모양 중 하나를 탐지할 수 있다. [Fig. 8]에서 (a)는 주먹 모양, (b)는 가위바위보의 보 모양, (c)는 우클릭 모양, (d)는 좌클릭 모양, (e)는 열린 가위 모양, (f)는 닫힌 가위 모양, (g)는 다이얼 모양이다. (e)와 (f)의 경우, 검지와 중지만 펼친 상태인데 두 상태를 구분하기 위해 검지와 중지가 이루는 각이 7.5° 이하이면 (f)의 닫힌 가위 모양이고, 이를 초과하면 (e)의 열린 가위 모양으로 판단한다[25]. 이 손 모양들은 상호 중복되지 않도록 정의해야 한다. 수화 같은 정교한 손 자세와 손 동작 제스처에 익숙하지 않은 사람들이 많고, 난해하고 복잡한 손가락 관절 표현은 손의 피로도를 증가시키며 직관성을 저하시킨다는 사실을 필히 감안해야 한다[19].

4.2 손 제스처 인식

제안된 손 제스처 인터페이스의 손 제스처와 그에 대응하는 사용자 이벤트의 종류는 〈Table 2〉와 같다. 손 제스처 판단에 사용되는 정보는 현재 프레임과 과거 프레임들에서 손 모양, 위치, 각도, 모양 유지 시간이다. 이를 위해 본 논문은 손 제스처 판단 전에 우선 [Fig. 8]의 손 모양들 중 어느 하나인지 아닌지를 판단한다. 먼저, 현재 프레임과 과거 프레임들에서의 손 모양을 통해 좌클릭 이벤트, 우클릭 이벤트, 더블 클릭 이벤트, 드래그

이벤트, 드롭 이벤트, 인터페이스 활성화 이벤트 및 인터페이스 비활성화 이벤트, 제스처 모드 전환 이벤트를 발생시킨다. 다음으로 [Fig. 2]의 손 랜드마크들 중 5번과 9번 랜드마크 위치의 평균점을 기준 삼아 수평 및 수직 방향으로의 손 움직임을 감지한다. 수평 방향의 손 움직임이 감지되면 수평 스크롤 이벤트를 발생시키고, 수직 방향의 손 움직임이 감지되면 수직 스크롤 이벤트를 발생시킨다[25].



[Fig. 9] Dial gesture that triggers a volume control event

그리고 [Fig. 9]의 점 A와 점 B를 지나는 직선의 기울기를 $grad$라고 했을 때, 지면에 수평인 직선과 점 A와 점 B를 지나는 직선이 이루는 각 θ를 식 (6)과 같이 구한다. 직전 프레임에서의 θ에서 현재 프레임에서의 θ를 뺀 값을 θ_{diff}라고 할 때, $\theta_{diff} > 3$인 경우엔 음량 감소 이벤트를 발생시키고 $\theta_{diff} < -3$인 경우엔 음량 증가 이벤트를 발생시킨다[25].

$$grad = \frac{y_b - y_a}{x_b - x_a}, \theta = \tan^{-1}(grad) \quad (6)$$

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

User's hand gestures	Operating conditions and holding time	User Events
Current frame hand shape (e), Previous frame hand shape (d)	-	Left click
Perform left-click gesture twice within a certain period of time.	Successive trial gap of less than 2 seconds	Double click
Current frame hand shape (e), Previous frame hand shape (c)	-	Right click
Move the cursor after maintaining the right-click gesture for a certain period of time.	Exceeds 0.5 seconds	Drag
During a darg event Current frame hand shape (e), Previous frame hand shape (d)	-	Drop
Move in one of the directions (up, down, left, right) with the hand shape in the form of (f)	-	(1) Scroll up (2) Scroll down (3) Scroll left (4) Scroll right
Hand shape in the form of (g) Decrease when rotating clockwise/ Increase when rotating counterclockwise	-	Volume up/down
After keeping hand shape (a) for a certain period of time, hand shape (b) (Gesture mode switching: When used in hand gesture mode, it switches to face gesture mode, and when used in hand gesture mode, it switches to face gesture mode.)	Exceeds 0.1 seconds	(1) Interface Enable (2) Interface disable (3) Gesture mode switching

5. 시뮬레이션 결과 및 고찰

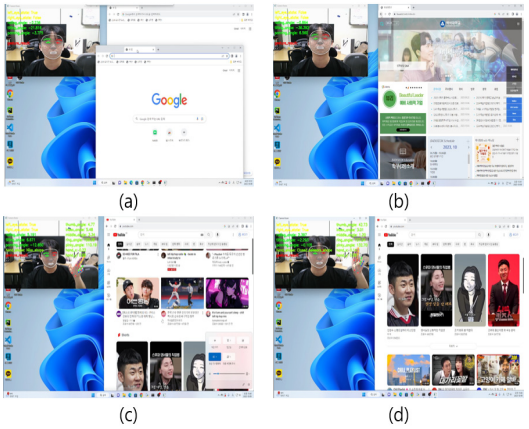
제안된 사용자 인터페이스의 사용성과 인식 성능을 평가하기 위해, Intel Core i7-12700(2.1GHz) CPU, DDR5 32GB RAM 데스크탑의 Windows 11 Pro 환경에서 NVIDIA RTX 3070 D6(8GB) GPU와 LogiTech C920 HD Pro 웹캠과 Python 3.8/MediaPipe 0.9.1.0/OpenCV 4.6.0/CUDA 11.4/cuDNN 8.2.2를 이용해 제안된 방식에 대한 컴퓨터 시뮬레이션을 수행하였다.

5.1 사용자 시나리오 시뮬레이션

본 논문에서는 사용자 컴퓨터 환경에서의 사용성을 검증하고 다양한 활용 사례를 제시하기 위해 다양한 사용자 시나리오에 대한 시뮬레이션을 진행하였다. 가상의 사용자는 제안한 손과 얼굴을 결합한 제스처 기반 사용자 인터페이스를 이용해 웹 브라우저 탭 전환, 웹서핑 중 뒤로 가기 버튼 누르기, 음량 조절, 수직 스크롤 이동 등의 작업을 수행하였다.

[Fig. 10]은 제안된 손-얼굴 혼성 제스처 인터페이스를 이용한 사용자 시나리오 시뮬레이션으로, 윈도우즈 환경에서의 소프트웨어 동작 제어 장면을 예시한 것이다. [Fig. 10]에서 (a)는 사용자가 웹 브라우저의 탭을 이동하는 장면으로, 사용자의 오른쪽 눈이 감긴 상태에서 드래그하는 장면을 보여준다. (b)는 뒤로 가기 버튼을 누르

는 장면으로, 좌클릭을 위해 양쪽 눈이 감긴 상태를 나타내며, (c)는 사용자가 음량을 조절하는 상황으로 손이 다이얼 모양을 표현하고 있다. 그리고 (d)는 수직 스크롤을 조작하는 장면으로, 손이 닫힌 가위 모양을 하고 있음을 보여준다. 이 시뮬레이션 과정을 통해 제안된 방법이 세밀하고 안정적인 제어를 가능하게 함을 확인할 수 있었다.



[Fig. 10] User scenario simulation using the proposed hand-face hybrid gesture interface

5.2 제스처 인식률

앞서 설명한 사용자 시나리오 시뮬레이션 외에도 얼굴과 손 제스처의 인식률을 평가하기 위해 각각 5명과 10명의 실험자에게 해당 제스처 모드를 수행하도록 하였다. 8가지 얼굴 제스처와 8가지 손 제스처에 대해 각각 30회씩 테스트를 진행하였으며, 〈Table 3〉은 각 사용자별 8가지 얼굴 제스처의 인식 성공 횟수와 평균 인식률을 나타낸다. 〈Table 4〉는 각 사용자별 8가지 손 제스처의 인식 성공 횟수와 평균 인식률을 보여준다. 평가 결과에 따르면, 얼굴 제스처의 평균 인식률은 95.8%, 손 제스처의 평균 인식률은 96.4%로 나타났으며, 실시간 처리에서도 실용적으로 무리가 없음을 확인할 수 있었다.

〈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	26	26	27	30	30	30	30	27	226
2	29	29	29	30	30	30	30	28	235
3	26	27	29	30	30	30	30	26	228
4	27	28	28	30	30	30	30	26	229
5	28	27	30	30	30	30	30	27	232
Avg (%)	**90.6**	**91.3**	**95.3**	**100**	**100**	**100**	**100**	**89.3**	**95.8**

〈Table 4〉 Usability evaluation of the proposed hand gesture recognition

User	Left Click	Double Click	Right Click	Drag	Drop	Scroll Up	Scroll Down	Enable/ Disable	Total
A	30	30	27	29	30	29	29	30	234
B	30	30	29	29	29	30	30	30	237
C	28	30	30	29	30	30	30	29	236
D	29	29	25	30	24	24	30	28	219
E	30	28	26	27	28	30	30	30	229
F	30	29	24	30	30	29	30	28	230
G	28	26	25	28	30	30	30	30	227
H	30	30	30	26	30	29	30	30	235
I	30	30	30	26	30	28	30	30	234
J	30	30	29	29	28	28	30	30	234
Avg (%)	**98.3**	**97.3**	**91.6**	**94.3**	**96.3**	**95.6**	**99.6**	**98.3**	**96.4**

6. 결론

본 논문에서는 MediaPipe Face Mesh와 Hands 모델을 기반으로 얼굴 및 손 제스처를 인식하는 손-얼굴 혼성 제스처 인터페이스를 구현해 웹서핑, 문서 작업, 사진 편집 등 다양한 소프트웨어 제어 시나리오를 위한 컴퓨터 시뮬레이션을 수행하였다.

제안된 인터페이스는 MediaPipe Face Mesh와 Hands 모델을 사용해 입력 영상에서 얼굴과 손의 3D 랜드마크 좌표를 추출하고, 얼굴의 Pan, Tilt, Roll 각도 및 양안의 개폐 상태를 바탕으로 얼굴 제스처를 인식하고, 손가락의 접힘과 손 모양 변화를 통해 손 제스처를 인식한다. 이를 통해 마우스 커서 이동과 같은 인터페이스 이벤트를 제어하였다.

사용성 평가 실험에서 얼굴 제스처는 평균 95.8% 인식률을, 손 제스처는 평균 96.4%의 인식률을 보였으며, 제안된 방식은 초당 30프레임의 속도로 일반적인 데스크탑에서 실시간 처리가 가능함을 확인하였다. 이 제스처 인터페이스는 얼굴과 손 제스처만으로도 자연스럽고 직관적인 사용자 경험을 제공하였으며, 기기와의 물리적 접촉 없이 제어가 가능해 산업 장비 조작 환경 등에서 활용 가능성이 높을 것으로 기대된다.

다만, MediaPipe Face Mesh와 Hands가 제공하는 3D 랜드마크 좌표는 입력 영상 전체의 깊이 정보가 아닌, 얼굴과 손에 국한된 상대적 깊이 정보를 제공하는 한계가 있다. 또한 단안 카메라 방식으로 인해 깊이 값의 신뢰도가 낮은 편이다. 이를 보완하기 위해 양안 카메라를 사용한 스테레오 정합 기술을 도입하여 얼굴 특정 지점에 대한 보다 정확한 깊이 정보를 추출하는 추가 연구가 필요하다.

REFERENCES

[1] 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.

[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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] N.Qian, J.Wang, F.Mueller, F.Bernard, V.Golyanik, and 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Google MediaPipe, https://ai.google.dev/edge/mediapipe/ (accessed Oct. 14, 2024).

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] J.Mok and N.Kwak, "Performance Improvement of Facial Gesture-based User Interface Using MediaPipe Face Mesh," Journal of Internet of Things and

Convergence, Vol.9, No.6, pp.118-125, Dec. 2024.

[25] J.Park, J.Mok, and N.Kwak, "Hybrid Gesture based User Interface Using MediaPipe," Proceedings of 2023 Fall Annual Conference of Korea Digital Contents Society, pp.34-40, Nov. 2023.

[26] T.Kim and N.Kwak, "User Interface Using Face Verification and Hand-Face Hybrid Gesture Recognition," Proceedings of 2024 Winter Annual Conference of Korean Institute of Communication Sciences, 19B-8, Feb. 2024.

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