MediaPipe Hands Model을 이용한 손 다이얼 및 마우스 제스처 인식 방법

송락빈, 홍윤아, 곽노윤
백석대학교 컴퓨터공학부
e-mail : songrb777@gmail.com

Hand Dial and Mouse Gesture Recognition Method Using MediaPipe Hands Model

Song Rakbin, Yuna Hong, Noyoon Kwak
Division of Computer Engineering, Baekseok University

Abstract

This paper is related to a gesture-based user interface, and the purpose of this paper is to propose a hand dial and mouse gesture recognition method using MediaPipe Hands Model. In this paper, various static poses are detected using landmarks, which are output values of MediaPipe Hands Model, and dynamic gestures are detected through motion and change of static poses. In this paper, computer simulation was performed focusing on the scenario of controlling a virtual smart home device using the proposed hand gesture recognition method.

I. 서론

사용자 인터페이스는 인간과 기계 사이의 접점을 없애고 인간의 의도를 자율적으로 파악하는 융합화·지능화 방향으로 발전하고 있고, 그 중에서도 직관적이고 자연스러운 몸짓(gesture)을 통해 사용자의 명령을 온전히 입력하는 것은 오랜 숙원이다. 손은 자연스러운 의사 표현 및 정보 전달 방법의 하나로 키보드나 마우스와 같은 전통적인 입력장치를 사용할 수 없는 디지털 기기에서 입력장치로 널리 사용되고 있다[1]. 손을 사용하는 정보 전달을 자동화하기 위해서는 손으로 표현하는 정보를 인식하는 방법이 필요한데 이를 손 표현 인식(hand expression recognition)이라고 한다. 손 표현 인식을 위해서는 손의 정적인 형태를 기반으로 하는 손 자세 인식(hand pose recognition)과 손의 동적인 움직임을 기반으로 하는 손 제스처 인식(hand gesture recognition)의 두 가지로 구분된다[2][3]. 최근 들어 구글의 미디어파이프(MediaPipe)[4]가 자세 추정 분야의 크로스 플랫폼 프레임워크의 중심으로 부상하면서 미디어파이프 핸즈 모델(MediaPipe Hands Model) 기반의 다양한 손 표현 인식 기술들[2][3]이 제안되고 있다. 예컨대 단안 카메라를 이용한 손 마우스 기능 구현, 손가락 숫자 인식, 손 필기 인식, 손 공간 터치, 수화 인식 등 다양한 손 표현 인식 방법들이 제안되고 있다. 하지만 그러한 기능들 중에서도 사용자 친화적이고 직관적인 사용자 인터페이스인 다이얼 기능이나 스크롤 기능의 탁월한 구현 사례는 찾아보기 어려운 편이다.

본 논문은 직관성과 편의성이 양호한 미디어파이프 핸즈 모델 기반의 손 다이얼 및 마우스 제스처 인식 방법을 제안함에 그 목적이 있다. 특히, 손 다이얼 및 마우스 기능이 실용적인 수준으로 구현되면 다양한 일상 생활 환경이나 소부장 산업 분야 등에서 유용한 사용자 인터페이스로 기능할 것이다. 이를 위해 본 논문에는 손 다이얼 및 마우스 제스처 인식을 위해 손가락의 위치와 형태를 기반으로 하는 계층적 손 자세 모델과 이 모델의 인식 결과를 동적으로 응용한 손 다이얼 제스처 모델을 제안하고자 한다. 손 자세 인식을 위해서는 오픈소스인 미디어파이프 핸즈 모델 기반으로 하고, 손가락 상태를 나타내는 모델과 이를 통해 손 다이얼 동작을 나타내는 모델을 계층적으로 구성하고자 한다.

II. 본론

본 논문에서는 손 다이얼 및 마우스 제스처 인식을 위한 정적 모델을 구글의 미디어파이프를 사용해 구성한다. 미디어파이프는 파이프라인 처리를 통해 다양한 데이터를 처리할 수 있도록 만들어진 크로스 플랫폼 프레임워크다. 미디어파이프에서는 자세 추정에 사용할 수 있도록 학습된 얼굴, 신체, 손 등의 모델을 제공하고 있으며, 본 논문에서는 미디어파이프에서 손 관절과 관련해 제공하는 21개 특징점들(landmarks)을 사용해 손 자세 모델을 구현한다. 추출된 특징점은 손 자세 인식에 바로 사용할 수는 없으므로 특징점의 상대적인 위치를 통해 손 자세를 나타낼 시 사용하는 손가락 자세를 정의하고, 손가락 자세를 통해 손 자세를 정의한다. 이러한 계층적인 손 자세 표현 방법을 통해 다양한 손 자세를 추상적으로 정의할 수 있으며 추정된 자세를 동적 모델과 함께 사용하면 완전한 손 제스처 인식 모델을 구성할 수 있다[2].

2.1 손가락 자세 탐지

손가락의 펼침(stretch)과 굽힘(bend) 여부를 알기 위한 가장 단순한 방법은 그림 1에서 특정 손가락 F_i일 때, F_i의 끝 부분인 TIP과 첫 번째 마디인 DIP의 좌표값을 비교하는 방법이다. 하지만 단순히 TIP과 DIP을 비교해 손가락이 펼침 여부를 탐지하는 알고리즘은 손을 돌려서 손 끝 부분이 바닥을 향한 상태에서는 인식이 어려운 단점이 있다. 본 논문에서는 손가락의 관절 각도를 이용해 손가락이 펼침 여부를 탐지하는

방법[2]을 채택하고 있다. 그림 1과 같이 손가락의 특정 특징점(landmark)을 C로 지정하고, C를 기준으로 연결된 위, 아래 특징점을 각각 x, y로 지정한다. 이후 C에서 x로 향하는 3차원 벡터 v_1과 C에서 y로 향하는 3차원 벡터 v_2를 구한다.

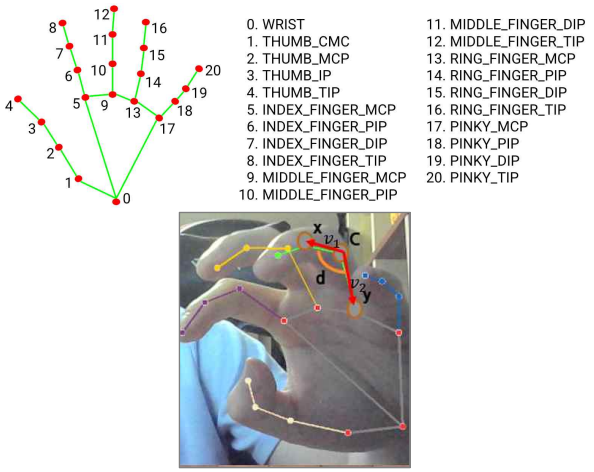


그림 1. 손가락에서 두 벡터를 이용한 굽힘 판단

두 벡터의 각도(d)를 구하기 위해 식 (1)과 같이 벡터의 내적을 활용한다. 이후, 각 손가락마다의 각도 변화를 참고해 검지(index finger), 중지(middle finger) 및 약지(ring finger)는 100°보다 작으면 손가락의 굽힘으로 판단하고 엄지(thumb)와 소지(pinky)의 경우에는 150°보다 작으면 굽힘으로 판정한다.

$$d = \cos^{-1}\left(\frac{x_1x_2 + y_1y_2 + z_1z_2}{|v_1||v_2|}\right) \quad (1)$$
$$where\ \ v_1 = (x_1, y_1, z_1),\ \ v_2 = (x_2, y_2, z_2)$$

2.2 손의 정적 자세

손의 정적 자세는 동적 제스처를 탐지하기 위한 손의 모양으로, 미디어파이프 핸즈 모델의 출력값인 특징점과 손의 정적 자세 정보를 이용해 구분한다.

2.2.1 손 포인트 자세

손 포인트 자세는 손의 상하좌우 동적 제스처를 입력으로 받기 위한 손의 모양이다. 손 포인트 자세의 조건은 그림 2와 같이 나머지는 접고 검지와 중지만을 펼친 상태에서 이 두 손가락들의 맞닿음(touch)이다. 중지와 검지의 TIP 거리를 이용해 두 손가락들의 맞닿음 여부를 조사한다. 즉, 두 TIP들 간의 거리를 구한 다음에 min_len1과 비교해 이 거리보다 작은 경우, 두 손가락이 맞닿은 것으로 판정한다. 이때 카메라 줌 인/아웃(zoom in/out)에 따라 손가락 길이가 가변되는 것을 감안해 그림 3과 같이 손 포인트 자세의 맞닿음 기준값(min_len1)은 손 내부에서 구할 수 있는 상대적 거리(중지의 DIP과 TIP 간 거리)를 사용한다.


그림 2. 손 포인트 자세

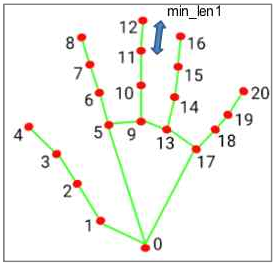
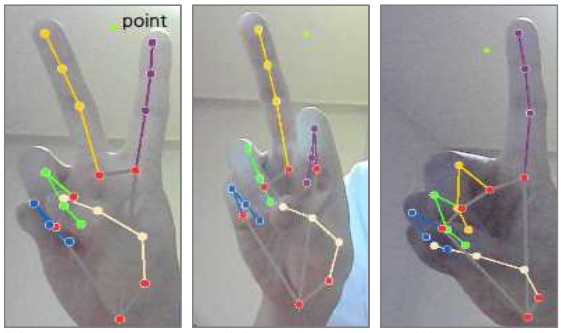


그림 3. 맞닿음 기준값(min_len1)

2.2.2 손 마우스 자세

손 마우스 자세는 실제 마우스와 유사하게 좌 클릭 및 우 클릭이 가능하도록 입력하기 위한 손의 정적 자세를 의미한다. 마우스의 좌 클릭과 우 클릭을 제스처로 이용하기 위해서 마우스 커서 모드, 좌 클릭 및 우 클릭 총 3 가지 자세로 구분된다.

그림 4와 같이 실제 마우스를 사용하는 것과 유사하게 중지와 검지를 펼친 상태에서 벌린 손 모양이 마우스 커서 모드로의 진입을 의미한다. 마우스 커서 모드에서 그림 4(b)와 같이 검지를 굽히면 좌 클릭으로 인식하고, 또한 그림 4(c)처럼 중지를 굽히면 우 클릭으로 인식한다. 마우스 커서의 위치인 마우스 포인트(mouse point)는 손가락을 굽혔을 때, 값이 크게 변하지 않는 특징점인 검지와 중지의 MCP들 간의 중점을 기준으로 손목(WRIST)의 좌표를 점대칭시킨 위치로 정한다.



(a)마우스 커서 모드 (b) 좌 클릭 (c) 우 클릭
그림 4. 손 마우스 자세

2.2.3 손 다이얼 자세

손 다이얼 자세는 다이얼 버튼과 같이 손을 돌려 각도 증감(+/-) 값을 다이얼 눈금의 증감 입력값으로 받기 위한 손의 모양이다. 손 다이얼 자세의 모양은 그림 5와 같이 손 포인트 자세에서 엄지를 펼친 상태이고 손 다이얼 제스처 모드 탐지 시, 각도의 변화를 화면에 보여하기 위해 검지의 TIP과 엄지의 TIP을 연결한 청색(blue) 직선을 표시한다.


그림 5. 손 다이얼 자세

2.3 손의 동적 제스처

손의 동적 제스처는 앞서 구한 정적 자세를 토대로 움직임이 판단되고 최종적으로 손의 제스처 입력으로 활용된다. 동적 제스처를 탐지하는 방식은 OpenCV를 통해 카메라의 초당 프레임 수(frame per second)가 n일 때, n×2을 n_frame으로 지정해 최근 n_frame 동안의 정적 자세 시퀀스를 연이어 저장한다. 즉, 저장된 프레임 수가 n_frame이 됐을 때부터 정적 자세 시퀀스의 변화를 확인해 지속적으

로 동적 제스처를 탐지한다. 본 논문에서는 앞에서 구한 정적 자세를 이용해 손 포인터 제스처, 손 마우스 제스처, 손 다이얼 제스처 등 총 3가지 동적 제스처를 탐지한다.

2.3.1 손 포인트 제스처

그림 2와 같이 손 포인트 자세를 취했을 때 생기는 녹색 포인트를 이동시키면, 상하좌우 움직임을 파악해 입력 시퀀스 데이터로 받는다. 손 포인트 제스처를 탐지하기 위한 조건은 다음과 같다.

·조건 1. 현재 프레임의 정적 자세는 손 포인트 자세이다.
·조건 2. 전체 n_frame 중 손 포인트 자세의 비율이 80% 이상이다.
·조건 3. 첫 번째 포인트와 n_frame 번째 포인트의 거리 차이가 min_len2 이상이다.

조건 2를 살펴보면 전체 프레임 중 80% 이상이 손 포인트 자세로 입력돼야 손 포인트 제스처로 인식한다. 그 이유는 미디어파이프 핸즈 모델(MediaPipe Hands Model)에서 발생하는 오차를 완충해 더 높은 확률로 동적 제스처를 탐지하기 위함이다. Hands 모델은 동작할 때, 중간중간 오인식으로 인해 실제와 다르게 탐지되는 경우가 종종 발생한다. 이런 경우엔 n_frame 동안의 정적 자세 시퀀스는 그림 6과 같이 저장되는데, n_frame 동안 부분적으로 오인식이 발생해 동적 제스처에 오판이 발생하거나 혹은 동작 제스처 자체가 탐지되지 않는 문제가 발생한다. 이를 해결하기 위해 전체 정적 자세 시퀀스 중 특정 비율 이상, 즉 80% 이상이 손 포인트 자세일 때만 충분한 손 포인트 자세들이 입력됐다고 판단해 포인트 제스처로 인식한다.

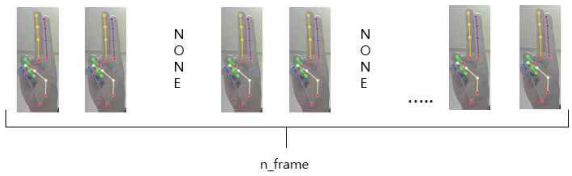


그림 6. 손 포인트 자세 시퀀스 입력 중 오인식에 의해 NONE 값 발생한 사례

조건 3에서는 포인트의 이동 여부를 측정하기 위해 min_len2라는 값을 이용한다. 손 포인트 제스처에서는 min_len2 이상의 거리를 이동했을 때, 손 포인트 제스처가 입력된 것으로 인식한다. 그림 7과 같이 손 포인트 제스처의 이동거리 기준값(min_len2)은 카메라 줌 인/아웃에 따라 가변되도록 손목 좌표인 WRIST와 검지의 MCP 간 거리로 정하고 있다.

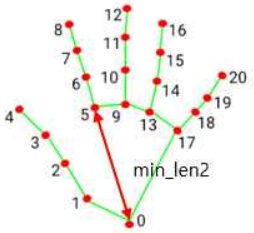


그림 7. 이동 거리 기준값(min_len2)

2.3.2 손 마우스 제스처

손 마우스 제스처는 그림 3과 같이 마우스 커서 모드에서 좌 클릭 혹은 우 클릭을 탐지해 이미지 내 마우스 포인트 위치에서 마우스 클릭을 발생시킨다. 이러한 손 마우스 클릭을 발생시키기 위한 조건은 다음과 같다.

·조건 1. 현재 프레임의 정적 자세는 좌 클릭 또는 우 클릭 자세이다.
·조건 2. n_frame/2 프레임 동안 마우스 커서 모드의 정적 자세 비율이 60% 이상이다.
·조건 3. 최근 n_frame/4 프레임 동안 좌 클릭 혹은 우 클릭 자세의 비율이 50% 이상이다.

조건 2와 조건 3을 보면 전체 n_frame 중 앞의 절반은 마우스 커서 모드의 정적 자세를 확인하고, 뒤의 최근 1/4은 클릭 정적 자세를 확인하는 것이다. 이러한 조건을 통해 손 마우스 제스처를 탐지하는 이유는 그림 8을 보면 알 수 있다.

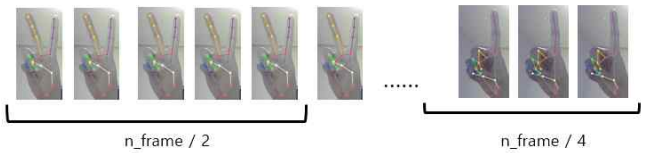


그림 8. 손 마우스 제스처 입력의 연속 동작

앞의 1/2 프레임들 동안 마우스 커서 모드 자세를 확인하는 이유는 마우스 커서 모드 상태에서 원하는 위치로 마우스 포인트를 이동시킨 후, 좌 클릭 혹은 우 클릭을 위한 손가락의 움직임이 있었음을 확인하기 위함이다. 그리고 뒤의 최근 1/4 프레임들 동안 클릭 자세를 확인하는 이유는 마우스 커서 모드 상태에서 움직임이 있었다면 그 움직임이 어떤 클릭 자세인지를 확인하기 위한 것이다.

2.3.3 손 다이얼 제스처

손 다이얼 제스처는 손 다이얼 자세를 취했을 시를 시작 각도로 지정해 손을 돌릴 때마다 회전 방향에 따른 각도 증감(+/-)을 30° 단위로 변환해 다이얼 눈금의 증감 입력값으로 받는다. 그림 9와 같이 사용자가 손 다이얼 자세를 취하면 검지의 TIP과 엄지의 TIP 간의 중심인 center를 구하고, center와 검지 TIP을 이용해 식 (2)의 시작 각도(start_deg)를 구한다. 이후 연속적으로 입력된 프레임들에서의 손 다이얼 자세에서 동일한 과정을 수행해 현재 각도를 구하고 시작 각도와 비교해 각도 변화량이 +30°를 초과할 때마다 1만큼 다이얼 눈금을 증가시키고 반대로 -30°를 초과할 때마다 1만큼 다이얼 눈금을 감소시킨다. 한편 각도 변화량이 +60°를 초과하면 2만큼 다이얼 눈금을 증가시키고 반대로 -60°를 초과하면 2만큼 다이얼 눈금을 감소시킨다. 즉, 각도 변화량이 30° 혹은 60°를 초과할 때마다 회전 방향에 따른 각도 증감(+/-)을 30° 단위로 변환해 다이얼 눈금의 증감 입력값으로 삼는다.

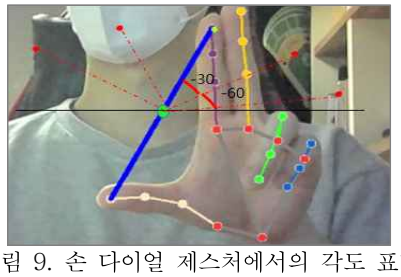


그림 9. 손 다이얼 제스처에서의 각도 표현

$$rad = -\text{atan2}(TIP.y - center.y,\ TIP.x - center.x) \quad (2)$$
$$start\_\text{deg} = \frac{(rad \times 180)}{\pi}$$

손 다이얼 제스처에서 각도 증감(+/-)을 다이얼 눈금의 증감 입력값으로 받기 위한 조건은 다음과 같다.

·조건 1. 현재 프레임의 정적 자세는 손 다이얼 자세이다.
·조건 2. 최근 n_frame/4 동안 손 다이얼 자세의 비율은

80% 이상이다.

·조건 3. 시작 각도(start_deg)를 기준으로 값의 변화량이 30° 또는 60° 이상이다.

Ⅲ. 시뮬레이션 결과 및 고찰

본 논문에서는 JetBrains PyCharm, Google MediaPipe Hands, OpenCV 4.6과 Logitech C930e WebCam, NVIDIA GTX 1070Ti GPU 환경에서 가상의 스마트홈 기기를 제어하는 컴퓨터 시뮬레이션을 수행하였다.

3.1 손 제스처 인식률

손 제스처를 이용한 입력 데이터 생성에 대한 인식률을 측정하기 위해 실험자가 100회씩 제스처를 이용해 데이터를 입력하였다. 그림 10은 손 포인트 제스처의 상하좌우(up/down/lef/right) 이동, 손 다이얼 제스처의 각도 증감(+/-), 그리고 손 마우스 제스처의 좌 클릭과 우 클릭(left_click/right click)을 수행하였을 때의 손 제스처 인식률을 나타낸 그래프이다.

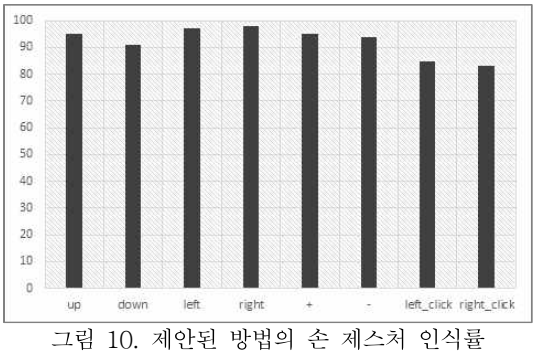


그림 10. 제안된 방법의 손 제스처 인식률

사용자가 정확한 입력할 수 있게 점과 선으로 현재 손의 정적 자세를 화면에 표시하기 때문에 전체적으로 양호한 정확도로 데이터를 입력할 수 있다. 하지만 손을 탐지하기 위해 1개의 단안 카메라(웹캠)만을 사용하기 때문에 손이 카메라 정면에서 벗어나게 되면 정확한 탐지가 어려운 경우도 발생한다. 대표적으로 마우스 클릭에서 클릭 과정에서 손가락을 접을 때 손이 아래로 기울기 때문에 다른 입력 방식보다 정확도가 낮은 것을 볼 수 있다.

3.2 가상의 스마트홈 기기 제어

본 논문에서는 제안된 손 제스처 인식 방법을 이용해 가상의 스마트홈 기기를 제어하는 시나리오를 중심으로 시뮬레이션을 수행하였다. 그림 11은 제안된 손 제스처 인식 방법을 이용해 구현한 메뉴로, Light, Memo, TV, Air Cleaner, Exit 총 5가지 기능을 수행하는 메뉴를 나타낸 것이다.

그림 11. 제안된 손 제스처 인식 방법을 이용한 가상의 스마트홈 기기 제어 메뉴

초기 상태에서 메인 메뉴를 on/off를 수행하기 위해서 사용자는 손 포인트 제스처를 통해 'down/up'을 입력한다. down 입력 및 up 입력을 통해 각각 메인 메뉴를 활성화 및 비활성화시킬 수 있다. 이후 사용자는 left/right 입력을 통해 원하는 메뉴(Light, Memo, TV, Air Cleaner, Exit)를 선택한다. 예컨대, TV 메뉴에서는 스마트홈에 연결된 TV의 채널과 볼륨을 조작한다. 메인 메뉴에서 TV 메뉴를 선택하면 TV의 볼륨과 메뉴가 출력된다. 사용자가 볼륨과 채널을 조작하기 위해 각각 왼손과 오른손의 다이얼 자세를 이용해 회전 방향에 따른 각도 증감(+/-)을 입력하면, 이를 30° 단위로 변환해 다이얼 눈금의 증감 입력값으로 받는다. 그림 12는 오른손 다이얼 제스처를 통해 +60° 이상 증가시켜 TV 채널을 2만큼 증가한 사례를 예시한 것이다.

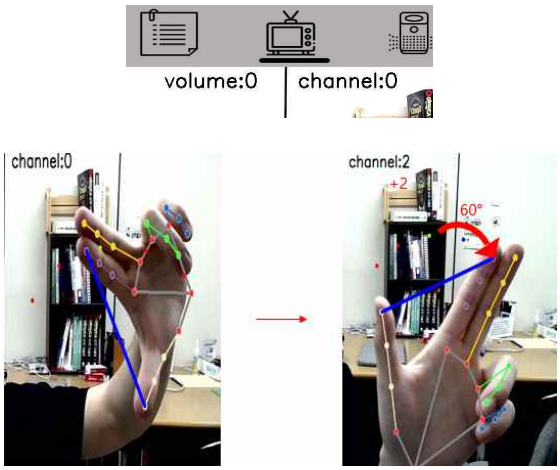


그림 12. 손 다이얼 제스처를 이용한 TV 채널 2만큼 증가

Ⅳ. 결론 및 향후 연구

본 논문에서는 미디어파이프 핸즈 모델을 이용해 손 제스처 기반의 사용자 인터페이스 기술을 개발하고 가상의 스마트홈 기기를 제어하는 컴퓨터 시뮬레이션을 수행하였다.

제안된 손 제스처 인식 방법은 미디어파이프 핸즈 모델을 이용해 손을 탐지하고 특징점을 사용해 정적 자세를 탐지한 후, 최종적으로 정적 자세의 변화를 계층적으로 감지해 동적 제스처를 입력 데이터로 탐지한다. 이를 토대로 손 포인트 제스처, 손 마우스 제스처, 손 다이얼 제스처 등의 인식 방법을 제안하였고, 가상의 스마트홈 기기 제어 시뮬레이션을 통해 그 유용성을 살펴보았다. 수화 같은 정교한 자세와 제스처에 미숙한 사람들이 적지 않고 복잡한 손가락 관절 표현은 피로도를 증가시키고 직관성도 떨어지기 때문에 직관성이 탁월하면서도 표현이 단순한 손 제스처 인식 방법을 고안하기 위해 노력하였다. 향후 다양한 장비들과 작업자가 혼재한 소부장 산업현장에서도 유용한 제스처 기반 사용자 인터페이스를 개발하기 위해 연구력을 집중할 것이다.

ACKNOWLEDGMENT

본 논문은 2022년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과이다.(2021RIS-004)

참고문헌

[1] 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, Feb. 2020.

[2] 허경용, 송복득, 김지홍, “손 표현 인식을 위한 계층적 손 자세 모델”, 한국정보통신학회논문지, 제25권, 제10호, pp. 1323-1329, 2021. 10.

[3] 허경용, 김명자, 송복득, 신범주, “가상 칠판을 위한 손 표현 인식”, 한국정보통신학회논문지, 제25권, 제12호, pp. 1770-1776, 2021. 12.
[4] Google MediaPipe, https://mediapipe.dev.