Journal of The Korea Society of Computer and Information
Vol. 19, No. 8, August 2014
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2014.19.8.035

깊이영상에서 효율적인 핸드 마우스를 위한 3D 포인팅

주 성 일*, 원 선 희*, 최 형 일*

3D Pointing for Effective Hand Mouse in Depth Image

Sung-Il Joo*, Sun-Hee Weon*, Hyung-Il Choi*

요 약

본 논문에서는 효율적인 핸드 마우스를 위한 3D 포인팅 인터페이스에 대해 제안한다. 제안하는 방법에서는 조명과 환경 변화에 강건하기 위해 깊이영상을 이용하고, 손바닥의 법선벡터를 이용하여 3D 포인팅을 구성한다. 먼저, 손 영역 검출과 추적은 기존 방법을 이용하고, 이로부터 획득한 정보를 바탕으로 손바닥의 영역을 예측하여 관심 영역을 획득한다. 관심 영역을 획득하면, 해당 영역을 평면의 방정식으로 근사시키고, 법선 벡터를 추출한다. 다음으로, 안정적인 제어를 위해 추출한 법선 벡터를 이용하여 보간을 수행하고, 교점을 검출한다. 검출된 교점은 안정성과 효율성을 위해 시그모이드 함수를 이용한 동적 가중치가 적용되고, 최종적으로 2D 좌표계로 변환된다. 본 논문에서는 관심 영역, 방향 벡터 검출 방법에 대해 설명하고 안정적인 제어를 위한 보간 방법과 동적 가중치 적용 방법에 대해 제안한다. 마지막으로 제안된 3D 포인팅의 정성적, 정량적 분석을 통해 안정적인 제어 가능성을 입증한다.

▸ Keywords : 깊이영상, 핸드 마우스, 동적 가중치, 3D 포인팅

Abstract

This paper proposes a 3D pointing interface that is designed for the efficient application of a hand mouse. The proposed method uses depth images to secure high-quality results even in response to changes in lighting and environmental conditions and uses the normal vector of the palm of the hand to perform 3D pointing. First, the hand region is detected and tracked using the existing conventional method; based on the information thus obtained, the region of the palm is predicted and the region of interest is obtained. Once the region of interest has been identified, this region is approximated by the plane equation and the normal vector is extracted. Next, to ensure stable control, interpolation is performed using the extracted normal vector and the

•제1저자 : 주성일 •교신저자 : 최형일
•투고일 : 2014. 8. 11, 심사일 : 2014. 8. 18, 게재확정일 : 2014. 8. 24.
* 숭실대학교 글로벌미디어학과(Dept. of Global Media, Soongsil University)
※이 논문은 서울시 산학연 협력사업(SS110013)의 지원을 받아 수행된 연구임.

intersection point is detected. For stability and efficiency, the dynamic weight using the sigmoid function is applied to the above detected intersection point, and finally, this is converted into the 2D coordinate system. This paper explains the methods of detecting the region of interest and the direction vector and proposes a method of interpolating and applying the dynamic weight in order to stabilize control. Lastly, qualitative and quantitative analyses are performed on the proposed 3D pointing method to verify its ability to deliver stable control.

▸ Keywords : Depth image, Hand Mouse, Dynamic Weight, 3D Pointing

I. 서 론

기계가 인간의 행동을 탐지하고 자동으로 분석하여 인지하는 것은 인공지능 분야에서 풀어야 할 과제로서 지난 수십 년 동안 활발한 연구가 진행되고 있다. 이를 위해 언어나 행동, 얼굴 표정, 근육의 움직임이나 뇌파 등 인체로부터 획득 가능한 다양한 신호를 통해 인간의 행동과 그 의도를 분석하고 있다. 그 중 인간의 시각을 모방하는 컴퓨터 비전(Computer vision) 분야에서는 인체의 형태를 인지하거나 몸짓과 같은 행동을 인식하고자 하는 연구가 활발하게 이루어지고 있다. 특히 이러한 비전 기술을 이용한 인간의 행동 분석 연구들은 최근 스마트 기기의 급속한 발전 및 보급화로 인해 기기 제어를 위한 스마트 인터페이스 기술에 다양한 형태로 응용되어 사용자들의 관심이 급증하고 있다. 또한 스마트 기기 시장의 활성화로 인해 기술적인 요구 수준이 점차 높아지고 있는 추세이다. 지능형 스마트 인터페이스는 사용자들의 기기 조작을 위한 편의성과 직관성을 최대한 반영하는 기술이 중요하게 요구된다. 즉, 사용자들로부터 최소한의 신호를 획득하고 이를 분석 및 인식하여 기기와의 상호작용이 자연스럽게 이루어질 수 있도록 하는 것이 지능형 스마트 인터페이스 기술의 최대 목표라 할 수 있다.

최근 스마트 인터페이스 기술 중에서 키넥트(Kinect)와 립모션(Leap Motion) 등 저가형 센서의 보급으로 인해 제스처 인식 기술에 대한 연구가 활발히 진행되고 있다. 이러한 제스처 인식 기술을 사용한 다양한 인터페이스 기술들은 대부분 사용자의 손을 탐색하고 동작을 인식함으로서 보다 직관적으로 기기를 제어할 수 있도록 한다. 손동작은 인간이 비언어적으로 표현할 수 있는 효과적인 의사소통 방식으로서 사람들 간의 복잡한 상호작용을 위해 간단한 동작으로 의사 전달이 가능하다. 컴퓨터 비전 분야에서의 제스처 인식 기술은 카메라를 통해 입력된 영상으로부터 사용자의 손을 인지하거나 손동작을 인식함으로서 사용자 명령을 파악하기 위한 것이다. 기존의 연구들에서는 손의 형태를 분석하는 정적 제스처 인식 연구에서부터 시간의 흐름에 따라 변화하는 손의 궤적을 분석하는 동적 제스처 인식 방법까지 다양한 관점에서 연구되어오고 있다.

II. 관련연구

기존의 제스처 인식 연구들은 손 영상을 분석하기 위해 외부적인 장치를 사용하는 방법[1-6]과 손의 형태를 인식하는 정적 제스처 인식[7-12], 손의 시간적인 움직임을 인식하는 동적 제스처 인식[13-16]에 대한 연구가 활발히 진행되고 있다. 외부적인 장치를 사용하는 방법은 데이터 글러브(Data glove)[1-3], 모션 캡쳐(Motion capture) 장비[4], 손에 마커[5]를 부착하거나 웨어러블 형태의 장치[6]와 같이 사용자의 신체에 착용되는 센서로부터 제스처에 대한 정보를 획득하는 방식이다. 이러한 방식은 사용자의 제스처를 인식하기 위해 반드시 인식을 위한 장치를 착용해야 가능하다는 점과 환경적인 제약 조건이 필요하다는 점에서 자연스러운 인터페이스 구축에는 많은 문제점을 안고 있다. 또한 카메라의 영상으로부터 색상정보를 이용하는 방법은 손의 피부 색상을 모델링하여 검출하고 영상의 저수준 특징을 이용하는 방식으로서 인식 환경내의 조명 조건에 따라 인식 성능이 감소한다는 문제점이 있다. 한편 색상 정보와 깊이 정보를 동시에 사용하는 방식은 특징 정보량이 많아지기 때문에 정확한 검출이 수행될 수 있으나 두 영상을 제공할 수 있는 하드웨어가 필요할 뿐만

아니라 연산량이 증가하게 되는 문제점을 안고 있다.

손의 형태를 인식하는 정적 제스처 인식과 동적 제스처 인식은 사용자의 직관적인 움직임을 통해 명령을 수행할 수 있도록 도와준다. 그러나 이러한 기술들은 사전에 정의된 제스처에 대해 사용자가 기억하고 수행해야 한다는 점에서 편의성이 낮아질 수 있다. 또한 이러한 문제점의 대안으로 연구되는 핸드 마우스 방식[17-19]은 제어 가능한 공간에 대한 문제와 기기 조작을 위해 사용자의 움직임이 비교적 많기 때문에 이로 인한 신체적인 피로감이나 불편함이 동반된다는 문제점이 있다.[20] 따라서 본 논문에서는 조명과 환경 변화에 강건하고 사용자의 편의성을 최대화하기 위한 방법으로 깊이 정보를 이용한 핸드 마우스 방식의 3차원 포인팅 기법을 제안하고자 한다.

논문의 구성은 3장에서는 3D 포인팅을 위한 방향 벡터 검출 및 교점 검출, 동적 가중치 적용 방법에 대해 설명하고 4장에서는 정량적, 정성적 실험을 통해 3D 포인팅 방법의 우수성을 기술하며 5장에서는 결론과 향후 연구에 대해 기술한다.

III. 3D 포인팅

제스처를 이용한 유저 인터페이스는 크게 3가지 종류로 분류할 수 있다. 첫째는 손의 움직이는 궤적을 분석하여 명령을 인지하는 동적 제스처(Dynamic Gesture) 인식이며, 두 번째는 손의 형태(Pose)를 분석하여 명령을 인식하는 정적 제스처(Static Gesture) 인식이다. 마지막으로 손의 위치와 변위에 따라 마우스를 이동시키고 그 외에 추가적인 인식을 통해 마우스와 같은 역할을 수행하는 핸드 마우스 방식이 있다.[17-19] 본 논문에서는 손바닥의 방향을 검출하고 이를 이용한 핸드 마우스 방식을 제안하고자 한다.

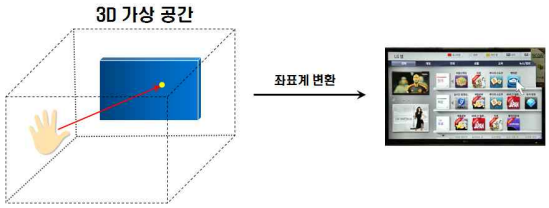
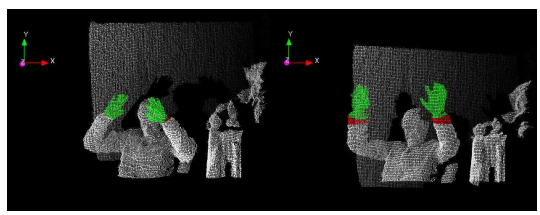


그림 1. 3D 포인팅 개념도
Fig. 1. 3D Pointing Conceptual Diagram

그림 1은 제안하고자하는 3D 포인팅의 개념도이다. 그림과 같이 추적된 손바닥의 방향으로 연장된 직선과 3D 가상공간 안에 있는 평면과의 교점을 계산하고, 획득한 교점을 실제 제어하고자하는 장치의 좌표계로 변환하는 과정을 통해 핸드 마우스 방식의 인터페이스가 구성된다. 기존 핸드 마우스 방식의 문제점은 마우스를 현재 위치로부터 많이 떨어진 위치로 이동시키기 위해서 상대적으로 많은 움직임이 필요하였다. 하지만 제안하는 3D 포인팅 방식의 경우 간편하게 손바닥의 방향과 손 위치의 이동을 이용하여 쉽게 큰 공간을 제어할 수 있다.

1. 평면 근사를 이용한 손바닥 방향 검출

본 절에서는 손바닥의 방향을 검출하는 방법에 대해 설명한다. 손바닥의 방향을 검출하기 위해서는 먼저 현재 손의 위치를 알아야 한다. 손 영역의 검출 및 추적은 Joo[21]가 제안한 방법을 사용하였다. 따라서 본 논문에서는 현재 프레임에서 손 영역의 위치는 주어졌다고 가정하고, 손의 방향 벡터를 결정하는 방법에 대해 설명한다.



(a) 영역 확장 결과가 관심 영역에 모두 포함된 경우 (b) 영역 확장 결과 중 일부만 관심 영역에 포함된 경우

그림 2. 관심 화소 선택
Fig. 2. Selection of the Pixels of Interest

그림 2는 Joo[21]가 제안한 영역 확장의 결과를 입력으로 하여 선택된 관심 화소를 보여준다. 녹색 또는 적색으로 표현된 영역은 영역 확장 결과이며, 녹색 영역은 영역 확장 결과 중 관심영역만 추출한 영역이다. (a)의 왼손은 영역 확장의 결과 영역이 모두 관심영역으로 포함된 상태이며, (b)는 약간의 영역을 제외한 영역이 관심영역으로 검출된 것으로 볼 수 있다. 이렇게 관심영역을 추출하는 이유는 그림과 같이 손바닥 외에 좀 더 많은 영역이 검출될 수 있기 때문이다. 따라서 추적점을 기준으로 유클라디언 거리를 이용하여 관심 화소를 선택한다. 관심영역 검출을 위한 거리 임계값은 카메라로부터 손의 거리를 이용하여 2차 다항 모델에 의해 α를 획득하여 계산된다.[21]

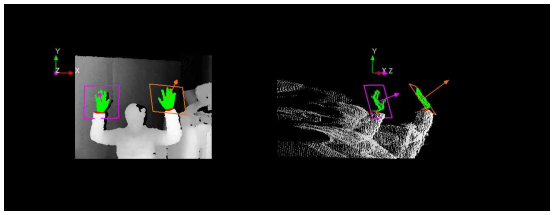
$$ax + by + c = z \tag{1}$$

$$A=\begin{bmatrix}a\\b\\c\end{bmatrix},\ X=\begin{bmatrix}x_1\ y_1\ 1\\x_2\ y_2\ 1\\.\ .\ .\\x_n\ y_n\ 1\end{bmatrix},\ Z=\begin{bmatrix}z_1\\z_2\\.\\z_n\end{bmatrix} \quad (2)$$

$$XA=Z \quad (3)$$

$$A=(X^TX)^{-1}X^TZ \quad (4)$$

〔수식 1〕~〔수식 4〕는 평면의 방정식을 검출하기 위해 최소 자승법(Least Square Method)을 이용한 수식이다.〔22, 23〕 먼저 〔수식 1〕과 같이 평면의 방정식을 세우고, 여러 개의 데이터들을 이용하기 위해 〔수식 2〕와 같이 행렬식으로 표현한다. 따라서 〔수식 1〕을 행렬식으로 표현하면 〔수식 3〕과 같은 형태로 정리할 수 있다. 마지막으로 〔수식 3〕을 의사 역행렬(pseudo inverse)을 이용하여 정리한 〔수식 4〕를 계산함으로써 최종 평면의 방정식을 획득할 수 있다. 따라서 손바닥의 방향 벡터는 평면의 방정식에 의해 $\vec{d}=(a,b,-1)$으로 정의된다.



(a) 정면 (b) 측면

그림 3. 평면 근사 결과
Fig. 3. Result of Plane Approximation

그림 3은 위에서 설명한 수식으로 구한 평면과 방향(법선 벡터) 결과를 보여준다. 방향은 평면의 법선 벡터로 관심 화소의 평균 위치를 기준으로 표현하였다. 그림과 같이 손바닥이 평평하지 않아도 안정적으로 동작함을 볼 수 있다.

평면의 방정식만으로도 방향 벡터를 계산하여 가상 스크린과의 교점을 계산할 수 있다. 하지만 매 프레임마다 평면의 법선 벡터를 방향 벡터로 정의하면 깊이 영상의 잡음으로 인해 떨림 현상이 발생할 수 있으며 너무 급격한 교점의 이동을 초래할 수 있다. 따라서 가중치를 두어 시간적으로 보간을 수행함으로써 점진적으로 방향이 결정되도록 한다.

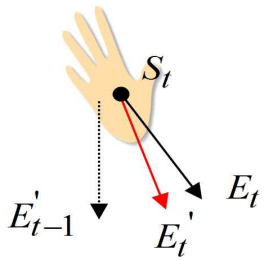


그림 4. 방향 벡터 보간
Fig. 4.Interpolation of the Direction Vector

그림 4은 방향 벡터를 보간하는 방법을 보여준다. S_t는 관심 화소의 중심 좌표이며, E_t는 S_t를 기준으로 t번째 프레임에서 검출된 방향 벡터의 끝점이다. 그림과 같이 이전 프레임의 E'_{t-1}과 현재 프레임의 E_t를 보간하여 E'_t를 계산한다.

〔수식 5〕는 E_t를 구하는 수식으로 $\vec{d}$는 평면의 방정식으로부터 획득한 법선 벡터이다. 또한 l은 방향 벡터의 길이이다. 〔수식 6〕은 이전 방향 벡터와 현재 방향 벡터를 이용하여 보간하는 수식으로 α에 의해 보간된다. 본 논문에서는 $\alpha=0.8$, $l=100$을 사용하였다. 따라서 최종적인 방향 벡터는 S_t를 원점으로 하는 $\vec{n}=\dfrac{E'_t-S_t}{\|E'_t-S_t\|}$이다.

$$E_t=S_t-l\times\frac{\vec{d}}{\|\vec{d}\|} \quad (5)$$

$$E'_t=\alpha E_t+(1-\alpha)E'_{t-1} \quad (6)$$

2. 교점 검출 및 좌표계 변환

그림 5는 3.1절에서 획득한 방향 벡터를 이용해서 교점을 구하는 방법을 보여준다. 먼저 방향 벡터로 직선의 방정식을 $S_t+\delta\vec{n}$으로 표현한다. S_t는 이전 3.1절에서 구한 관심 화소의 평균 좌표이며, $\vec{n}$은 이전 항에서 최종적으로 획득한 손바닥의 방향을 나타내는 노멀 벡터이다. 또한 $P0$와 $P1$은 가상 스크린 위의 임의의 점으로, 평면의 방정식을 $\vec{N}\bullet(P0-P1)=0$으로 표현한다. 여기서 $\bullet$은 내적 기호이며, $P0$와 $P1$이 가상 스크린의 평면 위에 있으므로 $(P0-P1)$ 또한 평면 위의 벡터이다. 즉, 가상 스크린의 법선 벡터 $\vec{N}$과 평면 위의 $(P0-P1)$ 벡터는 수직이므로 0을 만족한다. 따라서 손바닥의 방향 벡터 위에 존재하며, 가상

스크린의 평면의 방정식을 만족하는 c를 구하면 가상 스크린 위의 좌표를 찾을 수 있다.

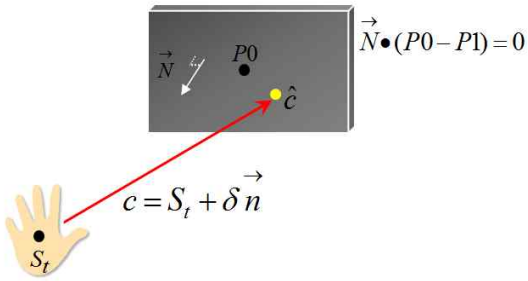


그림 5. 가상 스크린과 방향 벡터의 교점
Fig. 5. The Intersection Point of the Virtual Screen and the Direction Vector

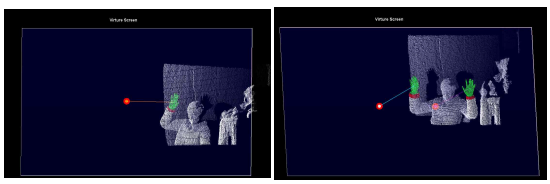
〔수식 7〕은 그림 5에서 $P1$에 c를 대입한 식이다. 〔수식 7〕에서 c는 손바닥의 방향을 나타내는 직선 위의 한 점이며 동시에 평면 위의 점이므로 〔수식 8〕로 정리되며, 최종적으로 〔수식 9〕를 얻을 수 있다. δ는 S_t점으로부터 $\vec{n}$방향으로 δ만큼의 거리의 위치가 교점이라는 것을 의미한다. 따라서 교점의 위치는 $\hat{c}=S_t+\delta\vec{n}$로부터 얻을 수 있다. 본 실험에서는 $P0=\{0,0,0\}$, $\vec{N}=\{0,0,-1\}$을 사용하였다.

$$\vec{N}\bullet\ (P0-c)=0 \tag{7}$$

$$\vec{N}\bullet\ P0-\vec{N}\bullet\ S_t-\delta(\vec{N}\bullet\ \vec{n})=0 \tag{8}$$

$$\delta=\frac{\vec{N}\bullet\ P0-\vec{N}\bullet\ S_t}{\vec{N}\bullet\ \vec{n}} \tag{9}$$

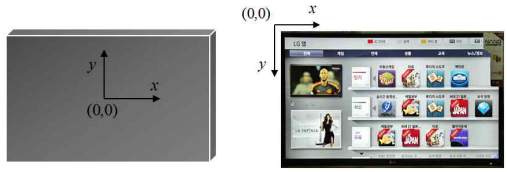
그림 6은 교점 검출 결과를 보여준다. 진청색 사각형이 가상 스크린이며, 적색 원이 가상 스크린과 손바닥 방향의 교점을 나타낸다.



(a) 한 손 교점 검출 결과 (b) 양 손 교점 검출 결과

그림 6. 교점 검출 결과
Fig. 6. Intersection Point Detection Result

$\hat{c}$은 가상 3차원 공간상의 손바닥의 방향과 가상 스크린 상의 3차원 교점을 나타낸다. 따라서 마우스의 좌표를 결정하기위해 실제 스크린 상의 2차원 좌표계로 변환해야 한다.



(a) 가상 스크린 좌표계 (b) 실제 스크린 좌표계

그림 7. 가상 스크린과 실제 스크린의 좌표 체계
Fig. 7.The Coordinate System of the Virtual Screen and the Actual Screen

본 논문에서는 3차원 공간상의 가상 스크린은 원점을 중심으로 z축에 수직인 V_w와 V_h의 길이를 갖는 평면으로 정의하였다. 따라서 간단하게 가상 스크린 상의 좌표는 스크린 상의 원점을 기준으로 x축과 y축의 좌표만을 이용해서 변환을 수행하면 된다.

$$R_x=\frac{(\hat{c_x}-vo_x+0.5\,V_w)}{V_w}R_w \tag{10}$$

$$R_y=R_h-\frac{(\hat{c_y}-vo_y+0.5\,V_h)}{V_h}R_h \tag{11}$$

〔수식 10〕과 〔수식 11〕은 실제 스크린 상에 좌표계 (R_x,R_y)를 계산하는 수식이다. $\hat{c_x}$과 $\hat{c_y}$는 가상 스크린과 손바닥 방향의 교점 $\hat{c}$의 x, y 좌표이며, vo_x와 vo_y는 가상 스크린의 중심 좌표이다. 또한 V_w와 V_h는 가상 스크린의 가로, 세로 길이이며, R_w와 R_h는 실제 스크린 상의 가로, 세로 해상도이다. R_y를 계산할 때 R_x와 다르게 R_h에서 감산을 수행한 이유는 그림 7과 같이 가상 스크린의 y좌표계와 실제 스크린 상의 y 좌표계가 서로 반대 방향이기 때문이다.

3. 시그모이드 함수를 이용한 동적 가중치

지금까지 손바닥 방향과 가상 스크린의 교점, 실제 스크린 상의 좌표를 계산하는 방법에 대해 설명하였다. 또한 손바닥의 방향 벡터를 시간 축으로 보간하여 민감한 변화를 완화시켰다. 하지만 손바닥의 방향으로 가상 스크린과의 교점을 계

산하기 때문에 깊이 영상의 잡음으로 인해 여전히 실제 스크린 좌표가 떨리는 현상이 발생한다. 이러한 이유로 사용자는 정확한 위치를 선택하는 것이 매우 어렵다. 따라서 이러한 단점을 해결하기 위해 이전 교점과 현재 교점의 거리 차이를 이용하여 동적 가중치를 계산하고 현재 교점을 구할 때 가중치를 적용하는 방법에 대해 설명한다.

$$x = \| \hat{c}^{t-1} - \hat{c}^{t} \| \tag{12}$$

$$w(x) = \frac{1}{1+e^{-\delta(x-\tau)}} \tag{13}$$

$$\hat{c}^{t} = w(x)\hat{c}^{t} + (1-w(x))\hat{c}^{t-1} \tag{14}$$

〔수식 12〕는 동적 가중치를 구하기 위한 파라미터를 구하는 수식이다. $\hat{c}^{t}$는 t 시점의 가상 스크린과의 교점으로 x는 손바닥이 향하는 방향으로 인해 생성되는 교점의 속도로 볼 수 있다. 〔수식 13〕은 동적 가중치를 구하는 수식으로 이전 교점과 현재 교점의 거리 차이 x에 의해 결정된다. 또한 δ는 경사도를 결정하는 상수이며, τ는 x가 항상 0보다 같거나 크기 때문에 가중치 함수 $w(x)$가 $x=0$일 경우 0에 근접하게 만들어주기 위한 상수이다. 〔수식 14〕는 최종 교점을 결정하는 수식으로 현재 계산된 교점에 동적 가중치를 곱하고 이전 교점에 1에서 동적 가중치를 감산한 가중치를 곱하여 최종 교점을 검출한다. 따라서 사용자가 손의 방향을 크게 변화시킨다면 동적 가중치가 1이 되어 검출된 교점 위치대로 이동하며, 손 떨림 현상으로 교점의 위치가 약간의 변화를 갖는다면 현재 검출된 교점의 영향력을 0에 가깝게 줄여 떨림 현상을 제거할 수 있다.

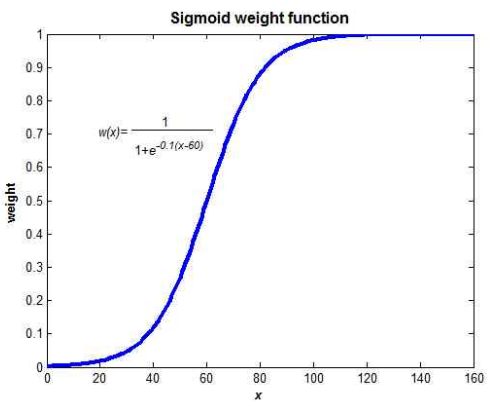


그림 8. 시그모이드 동적 가중치 함수 예
Fig. 8. Example of the Sigmoid Dynamic Weight Function

그림 8은 $\delta=0.1, \tau=60$일 때의 동적 가중치 함수의 예를 보여준다. 그림과 같이 시간에 따른 교점 $\hat{c}$의 변화가 클 경우 동적 가중치는 1의 값을 갖으며, 교점의 변화가 20 미만일 경우 0.1이하의 가중치를 적용함으로써 떨림 현상을 해결할 수 있다.

Ⅳ. 실험 결과

그림 9는 제어 정확도를 분석하기 위한 환경을 보여준다. 그림과 같이 카메라로부터 약 1m 정도 떨어진 위치에서 녹색 화살표 방향으로 조금씩 이동하여 가상 스크린과의 교점을 저장하였다.

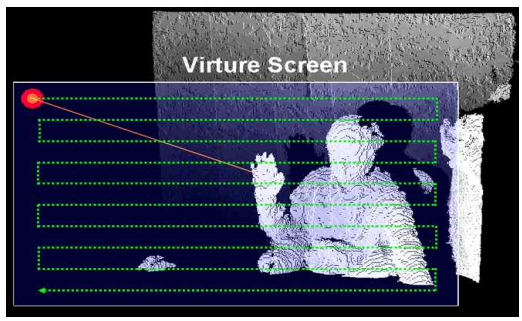


그림 9. 3D 포인팅 제어 정확도 실험을 위한 환경
Fig. 9.The Environment for Testing the Accuracy of 3D Pointing Control

그림 10은 시그모이드 함수를 이용한 동적 가중치를 적용하지 않고 평면 검출과 교점 검출만으로 그림 9와 같이 가상 스크린 전체 영역을 스캔한 교점의 히스토리를 보여준다. 그림과 같이 손바닥의 약간의 변화에도 민감하게 반응하여 교점의 위치가 매우 불안정하게 이동한 것을 볼 수 있다. 또한 교점의 거리가 촘촘한 이유는 동적 가중치를 적용하지 않았기 때문에 약간의 변화에도 민감하게 반응하여 정확한 제어를 위해 모든 이동에 대해 최대한 천천히 손바닥의 방향을 변화시켰기 때문이다. 반면 그림 11은 동적 가중치 ($\delta=0.1, \tau=60$)를 적용한 그림으로 비교적 안정적으로 이동된 것을 볼 수 있다. 또한 가로로 이동할 경우 빠르게 이동하였기 때문에 교점이 비교적 촘촘하지 않았으며 세로로 이동할 경우에는 정확하게 약간만 아래로 이동하기 위해 천천히 손바닥 방향을 변경하였기 때문에 교점이 촘촘한 것을 볼 수 있다.

따라서 동적 가중치를 적용함으로써 사용자의 의도에 따라 많은 이동에는 빠르게 이동할 수 있도록 하고 정확한 위치로 이동하기 위한 약간의 변화에는 천천히 이동하도록 돕기 때문에 좀 더 효과적으로 제어할 수 있음을 알 수 있다.

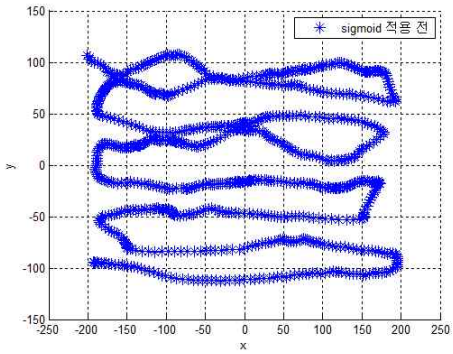


그림 10. 동적 가중치 적용 전
Fig. 10. Before Applying the Dynamic Weight

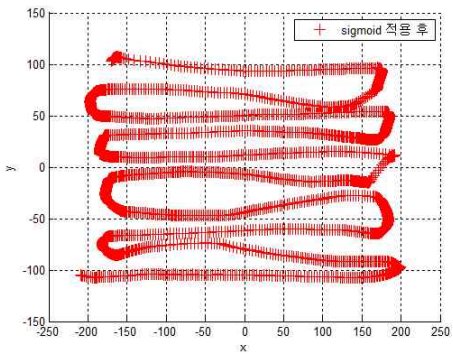


그림 11. 동적 가중치 적용 후
Fig. 11. After Applying the Dynamic Weight

그림 12는 교점을 임의의 위치에 이동시키고 손바닥을 최대한 움직이지 않은 상태로 약 2초(60프레임)간의 교점 위치를 표현한 그림이다. 그림과 같이 동적 가중치를 적용한 경우에는 가상 스크린과의 교점이 거의 변화가 없었으며, 동적 가중치를 적용하지 않은 경우에는 +7~-7 정도의 변화가 있었다. 이는 깊이 영상 자체의 잡음으로 인해 평면 근사에 영향을 주고, 이러한 결과로 교점에도 미세한 영향이 미치기 때문이다. 따라서 동적 가중치를 적용함으로써 비교적 정확한 제어가 가능함을 알 수 있다.

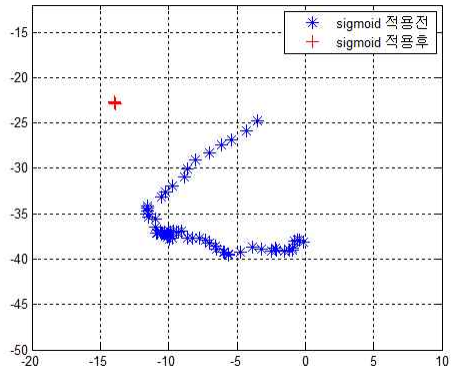


그림 12. 정지된 상태의 교점 좌표 변화
Fig. 12. Change in the Coordinates of the Immobilized Intersection Point

그림 13은 제어의 정확도 및 속도 평가를 위한 실험 방법을 보여준다. 가상 스크린의 범위는 x축으로 -192~+192 범위이며 y축으로는 -108~108로 설정하였다. 실험 방법은 초기에 가상 스크린 중앙에 목표 위치를 보여주고 사용자가 제어를 통해 교점을 이동시켜 목표 위치와 임계값 이하의 거리 안에 포함되면 성공한 것으로 인식하고 가상 스크린 내부의 랜덤한 위치로 목표 위치를 이동시켰다. (a)는 초기 상태를 보여주고, (c)와 같이 목표 위치에 도달하면 (d)처럼 랜덤한 위치로 목표 위치가 이동된다. 이와 같은 방법으로 성공 여부를 결정하는 거리 임계값을 달리하고 목표 위치 도달 후 다음 목표 위치까지 교점을 이동시켜 성공할 때까지 걸리는 평균 시간을 측정하였다. 실험은 5명에 대해 약 40번 시도하여 측정하였다.

(a) t frame (b) t+25 frame (c) t+30 frame

(d) t+42 frame (e) t+62 frame (f) t+67 frame

그림 13. 제어 정확도 및 속도 실험 방법
Fig. 13. Method of Testing the Accuracy and Speed of Control

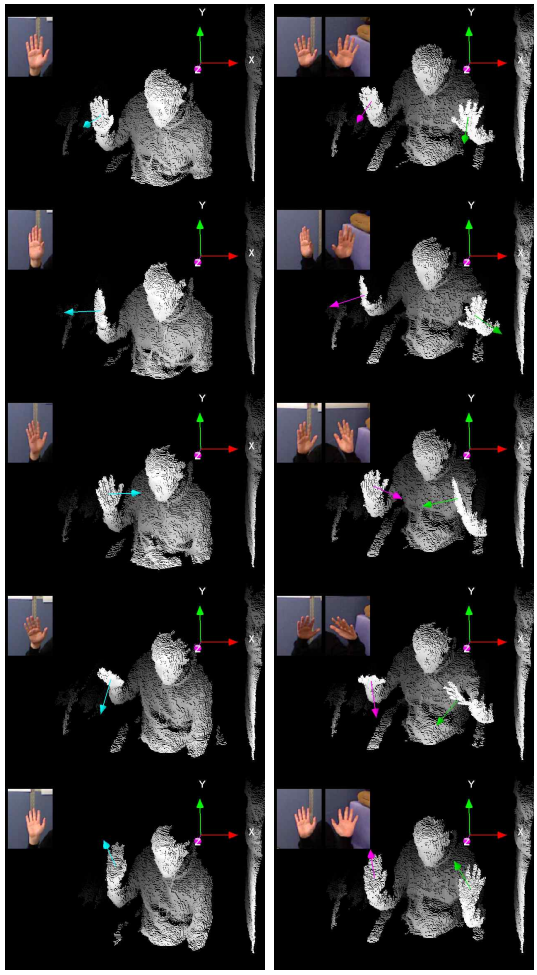
표 1은 위에서 설명한 실험 방법으로 평가한 결과이다. 거리 임계값을 10으로 설정한 경우에는 동적 가중치를 적용하

지 않은 방법이 더 빠른 결과를 보였다. 하지만 거리 임계값을 7~3으로 설정한 실험에는 동적 가중치를 적용한 방법이 더 우수한 결과를 보였다. 임계값을 10으로 한 경우, 동적 가중치를 적용하지 않은 방법이 더 빠른 결과를 보인 것은 사용자의 제어에 빠른 반응을 보여 목표 위치까지 빠르게 도달하였으며 교점이 불안정하게 떨려 쉽게 성공 조건을 만족 시킬 수 있기 때문이었다. 하지만 거리 임계값이 줄어들수록 동적 가중치를 적용하지 않은 방법은 정확하게 목표 위치에 도달하기 어려웠으며 그로 인해 좀 더 많은 시간이 소요되었다.

표 1. 제어 속도 평가 결과
Table 1. Control Speed Evaluation Results

거리 임계값	평균 시간(초)	
	동적 가중치 적용	동적 가중치 미적용
10	1.734	1.555
7	1.868	2.04
5	2.174	2.226
3	2.515	2.807

그림 14는 손바닥의 방향에 따른 검출된 방향을 보여준다. (a)는 한손에 대한 방향 검출 결과를 보여주고, (b)는 양손에 대한 검출 결과를 보여준다. 먼저 가장 첫 번째 행은 정면에 대한 방향을 나타내며, 두 번째 행은 몸 바깥 쪽 방향으로 손바닥을 회전시켰을 경우를 보여준다. 또한 세 번째 행은 몸 안쪽 방향이며, 네 번째는 아래 방향, 마지막 행은 위쪽 방향을 향했을 경우 검출 결과를 보여준다. 그림과 같이 모든 방향에 대해 약 -45~+45도까지 정상적으로 검출된 것을 볼 수 있다. 따라서 작은 공간에서 매우 큰 영역을 제어할 수 있으며, 다양한 제스처 인식 방법을 통해 많은 응용이 가능하다.



(a) 한 손 방향 검출 결과 (b) 양손 방향 검출 결과

그림 14. 포즈 별 손바닥 방향 검출 결과
Fig. 14. Detection Results for the Direction of the Palm of the Hand in Each Pose

V. 결 론

본 논문에서는 깊이 영상을 이용하여 좁은 공간에서 큰 공간을 효율적으로 제어하는 방법에 대해 제안하였다. 손바닥의 관심 영역을 검출하고, 검출된 영역을 평면의 방정식으로 근사시키고, 이를 이용하여 손바닥의 방향을 추출하였다. 이렇게 함으로써 사용자가 좀 더 적은 움직임으로 효율적으로 제어 가능함을 보였다. 또한 제안한 인터페이스 특성 상 약간의 움직임에도 큰 움직임을 보인 교점을 동적 가중치를 적용함으로써 안정적으로 동작함을 확인하였다. 향후 다양한 분야에 적용을 하기 위해 이벤트 처리 방법 및 적용 방법에 대한 추가적인 연구가 필요할 것이다.

참고문헌

[1] *http://www.cyberglovesystems.com/products/cyberglove-ii/overview*

[2] D. J. Sturman and D. Zeltzer, "A Survey of Glove-based Input", IEEE Computer Graphics and Applications, Vol. 14, No. 1, pp. 30-39, January 1994.

[3] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill, "A Hand Gesture

Interface Device", CHI'87 In proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface, pp. 189-192, May 1987.

〔4〕 *http://www.dh.aist.go.jp/en/research/centered/dhand-link2/*

〔5〕 R. Y. Wang and J. Popović, "Real-Time Hand-Tracking with a Color Glove", ACM Transaction on Graphics(TOG), Vol. 28, No. 3, Article 63, August 2009.

〔6〕 D. Kim, O. Hilliges, S. Izadi, A. Butler, J. Chen, I. Oikonomidis and P. Olivier, "Digits : Freehand 3D Interactions Anywhere Using a Wrist-Worn Gloveless Sensor", In proceedings of the 25th annual ACM symposium on User Interface Software and Technology, pp. 167-176, October 2012.

〔7〕 S. P. Priyal and P. K. Bora, "A robust static hand gesture recognition system using geometry based normalizations and Krawtchouk moments", International Journal on Pattern Recognition, Vol. 46, No. 8, pp. 2202-2219, August 2013.

〔8〕 Z. Song, H. Yang, Y. Zhao and F. Zheng, "Hand Detection and Gesture Recognition Exploit Motion Times Image in Complicate Scenarios", In proceedings of Advances in Visual Computing - 6th International Symposium, ISVC 2010, Vol. 6454, pp. 628-636, November 2010.

〔9〕 S. Malassiotis and M. G. Strintzis, "Real-time hand posture recognition using range data", International Journal on Image and Vision Computing, Vol. 26, No. 7, pp. 1027-1037, July 2008.

〔10〕 F. Kirac, Y. E. Kara and L. Akarun, "Hierarchically constrained 3D hand pose estimation using regression forests from single frame depth data", International Journal on Pattern Recognition Letters- Special Issue on Depth Image Analysis, September 2013.

〔11〕 P. Suryanarayan, A. Subramanian and D. Mandalapu, "Dynamic Hand Pose Recognition using Depth Data", In proceedings of the 2010 International Conference on Pattern Recognition, pp. 3105-3108, August 2010.

〔12〕 F. Dominio, M. Donadeo and P. Zanuttigh, "Combining multiple depth-based descriptors for hand gesture recognition", International Journal on Pattern Recognition Letters, October 2013.

〔13〕 N. D. Binh, E. Shuichi and T. Ejima, "Real-time hand tracking and gesture recognition system", In proceedings of International Conference on Graphics, Vision and Image Processing, pp. 362-368, December 2005.

〔14〕 A. Kurakin, Z. Zhang and Z. Liu, "A Real Time System for Dynamic Hand Gesture Recognition with a Depth Sensor", In proceedings of the 20th European Signal Processing Conference, pp. 1975-1979, August 2012.

〔15〕 M. Elmezain, A. Al-Hamadi and B. Michaelis, "Hand trajectory-based gesture spotting and recognition using HMM", In proceedings of the International Conference on Image Processing, pp. 3577-3580, November 2009.

〔16〕 X. Liu and K. Fujimura, "Hand Gesture Recognition using Depth Data", In proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 529-534, May 2004.

〔17〕 H. J. Park, "A Method for Controlling Mouse Movement using a Real-Time Camera", Brown University, Providence, RI, USA, Department of Computer Science, 2010.

〔18〕 D. D. Luong, S. Lee and T. S. Kim, "Human Computer Interface Using the Recognized Finger Parts of Hand Depth Silhouette via Random Forests", In proceedings of the 2013 13th International Conference on Control Automation and Systems, pp. 905-909, October 2013.

〔19〕 A. Aksaç, O. Öztürk and T. Özyer, "Real-time Multi-Objective Hand Posture/Gesture

Recognition by Using Distance Classifiers and Finite State Machine for Virtual Mouse Operations", In proceedings of the 2011 7th International Conference on Electrical and Electronics Engineering, pp. II-457-II-461, December 2011.

〔20〕 S. I. Joo, "Dynamic Soft Cascade : Application to Gesture Recognition", Ph.D. Thesis, University of Soongsil, June 2014.

〔21〕 S. I. Joo, S. H. Weon and H. I. Choi, "Real-Time Depth-Based Hand Detection and Tracking", The Scientific World Journal, Vol. 2014, Article ID 284827, pp. 1-17, March 2014.

〔22〕 S. J. Miller, "The Method of Least Squares", Mathematics Department Brown University (2006): 1-7.

〔23〕 D. Eberly, "Least Squares Fitting of Data", *http://www.geometrictools.com/*, 1999.

저 자 소 개

주 성 일

2008 : 한국산업기술대학교 컴퓨터공학과(공학사)
2010 : 숭실대학교 미디어학과(공학석사)
2014 ~ 현재 : 숭실대학교 미디어학과(공학박사)
관심분야: Image Processing, Computer Vision, Pattern Recognition, Machine Learning.
Email : sijoo82@ssu.ac.kr

원 선 희

2005 : 한경대학교 컴퓨터공학과(공학사)
2007 : 숭실대학교 컴퓨터학과(공학석사)
2012 : 숭실대학교 미디어학과(공학박사)
현 재 : 숭실대학교 미디어학과 Post Doc.
관심분야: Image Processing, Computer Vision, Pattern Recognition, Machine Learning.
Email : nifty12@ssu.ac.kr

최 형 일

1972 : 연세대학교 전자공학과(공학사)
1982 : 미시간대학교 전자공학과(공학석사)
1987 : 미시간대학교 전자공학과(공학박사)
1995 : 퍼지 및 지능시스템학회 이사
1996 : 정보과학회 컴퓨터비전 및 패턴인식 연구회 위원장
1997 : IBM Waston Lab 방문연구원
2005 : 한국정보과학회 이사
현 재 : 숭실대학교 미디어학과 교수
관심분야: 컴퓨터공학
Email : hic@ssu.ac.kr