

# CAMShift와 이중 원형 추적법을 이용한 손 동작 게임 컨트롤러 구현

## Implementation of Finger-Gesture Game Controller using CAMShift and Double Circle Tracing Method

이우범\*

Woo-beom Lee\*

### 요약

본 논문에서는 단일 카메라를 사용하여 사용자 검지 손가락 움직임과 손가락 개수의 인식을 기반으로 하는 손동작 게임 컨트롤러 인터페이스를 구현한다. 구현한 손동작 게임 컨트롤러 인터페이스는 검지 손가락 끝점의 위치 추적을 위해서 CAMShift(Continuously Adaptive Mean Shift) 알고리즘을 적용하고, 손가락 개수 인식을 위해서는 손 영역의 중심으로 부터 이중 원형 추적(Double Circle Tracing)에 의한 교차정보를 이용한다. 이때 성능 향상을 위해서 RGB 입력 영상에 대해서 CAMShift 알고리즘 적용에는 HSI 컬러모델을 이중 원형 추적을 위해서는 YCbCr 컬러모델을 사용한다. 또한 인텔사의 OpenCV 라이브러리를 기반으로 C++언어를 사용하여 사격 시뮬레이터 게임을 제작하여 손동작 게임 컨트롤러 인터페이스의 성능을 평가하고 사용자 명령 컨트롤러로서의 유효성을 검증하였다. 그 결과 각 게임 컨트롤 모드별 평균 90% 이상의 인식률을 보였다.

### ABSTRACT

A finger-gesture game controller using the single camera is implemented in this paper, which is based on the recognition of the number of fingers and the index finger moving direction. Proposed method uses the CAMShift algorithm to trace the end-point of index finger effectively. The number of finger is recognized by using a double circle tracing method. Then, HSI color mode transformation is performed for the CAMShift algorithm, and YCbCr color model is used in the double circle tracing method. Also, all processing tasks are implemented by using the Intel OpenCV library and C++ language. In order to evaluate the performance of the proposed method, we developed a shooting game simulator and validated the proposed method. The proposed method showed the average recognition ratio of more than 90% for each of the game command-mode.

**Keywords** :CAMShift Algorithm, Double Circle Tracing, Finger-Gesture Game Controller, Shooting Simulator Game, Shape Variation

## I. 서론

컴퓨터 비전 인식 기술을 기반으로 인간의 손동작이나 손가락을 이용하여 기계에게 명령을 부여하는 사용자 인터페이스(UI: User Interface) 기술은 최근 스마트폰 또는 태블릿 PC의 보급으로 가장 주목 받고 있는 기술 중에 하나이다[1].

특히, 손에 어떠한 색상 마커(marker) 또는 기기 장비도

착용하지 않는 비장착형 손동작 인터페이스 기술은 카메라의 성능에 영향을 많이 받는 단점에도 불구하고, 사용자 편의성과 추가적인 장치 없이 인식이 가능하며 구현에도 추가적인 비용이 발생하지 않기 때문에 마이크로소프트 XBOX의 키넥트(Kinect)와 같은 영상 기반의 게임기에서 사용자 컨트롤러로서 많이 연구되고 있다[2,3].

그리고 이러한 대부분의 연구들이 시스템 적용 UI를 위한 최적화된 제스처 정의와 인식에 있어서 어느 정도의 기능과 성능을 보장하고 있다. 그러나 동일한 손동작일지라도 행하는 사람에 따라서 다양한 표현이 가능하기 때문에 범용적 인식에 어려움이 있으며, 빛 상태와 같은 조명 조건에 따른 제약에 의해서 확실한 해법을 제시하지 못하고 있는 실정이다.

따라서 본 논문에서는 게임 시뮬레이터에서 최적화된 컨

\* 상지대학교 컴퓨터정보공학부

투고 일자 : 2014. 4. 10 수정완료일자 : 2014. 4. 29

게재확정일자 : 2014. 5. 2

\* 이 논문은 2012-2013년도 상지대학교 교내 연구비 지원에 의한 것임.

트roller 기능을 보이는 손동작 인터페이스를 구현한다. 구현하는 손동작 게임 컨트롤러는 검지 손가락의 끝점 추적과 손가락 개수의 변화 인식을 기반으로 한다.

검지 손가락의 끝점 추적은 CAMShift 알고리즘에 의해서 사용자 컨트롤러의 움직임(moving) 제어와 연동되며, 손가락 개수의 인식은 사용자 명령(user command)과 연동된다. 특히, 인식의 성능을 높이기 위해서 각각의 경우에 HSI와 YCbCr로의 칼라 모델 변환을 수행한다. 그리고 구현한 손동작 컨트롤러는 사격 시뮬레이터 게임에서 사용자 명령 입력기로 적용하여 그 유효성을 보인다.

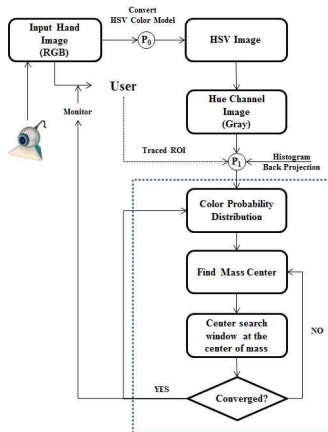


그림 1. 검지 손가락 추적 처리과정  
Fig. 1. Index Finger Tracing Processing Task.

## II. CAMShift 손가락 끝점 추적

본 논문에서 구현하는 손동작 게임 컨트롤러는 손가락 끝점 추적을 위한 모듈과 손가락 제스처에 연동되어서 사용자 명령을 수행하는 손가락 제스처 인식 모듈로 구성된다.

손가락 끝점 추적 모듈은 (그림 1)과 같이 객체 추적에 효율적으로 사용되고 있는 Mean Shift 알고리즘[4] 기반의 CAMShift 알고리즘[5,6]을 이용하여 손가락 영역을 추적하고 그 이동 방향을 연동하여 사용자 게임 모니터에 제시한다.

실시간으로 입력되는 비디오 프레임으로부터 검지 손가락의 이동 위치를 검출하기 위해서는 먼저 입력 RGB 영상으로부터 HSI 칼라 모델 변환을 하여 색상(Hue) 채널의 영상을 생성한다. 특히, 사용되는 HSI 칼라 모델은 RGB 컬러모델과는 다르게 색상 정보와 밝기 정보를 분리하는 것이 가능하기 때문에 손 영역 검출에 있어서 빛의 영향을 최대한 줄여주는 것이 가능하다. 이 때 칼라모델 변환에서 사용되는 r, g, b 값은 실제 영상 출력이나 특징값 계산을 위해서 선형 변환 후의 값을 이용한다.

입력 영상의 Hue 채널이 생성되면 Hue 채널의 확률 분포를 이용하여 손가락 검지를 추적하기 위한 CAMShift 알고리즘 적용한다. CAMShift 알고리즘은 모드 탐색(Mode seeking) 알고리즘으로 특정 데이터의 중심 이동(Mean shift)을 이용하여 관심영역

을 추적하는 Mean Shift 알고리즘[6]을 기반으로 하고 있다. 그리고 Mean Shift 알고리즘의 각 단계는 아래와 같다[4].

1. 크기  $s$ 의 탐색 윈도우  $W$ 를 설정한다.
2. 초기 탐색 윈도우의 중심값 :
3. 탐색 윈도우  $W$ 에서의 평균 위치는 아래와 같이 계산한다.

$$p_k(\cdot) = \frac{1}{|W|} \sum_{j \in W} p_j \quad (1)$$

그리고 중심 이동은 아래와 같이 함수  $f(p)$ 의 경사를 따른다. 여기서, 함수  $f(p)$ 는 중심 값의 분포함수이다.

$$\hat{p}_k(W) - p_k \approx f'(p_k) \quad (2)$$

4. 윈도우를  $\hat{p}_k(W)$ 에 중심을 설정한다.
5. 단계 3과 4를 수렴할 때까지 반복한다.

이상의 과정에 의해서 Mean Shift 알고리즘은  $f'(p) \approx 0$ 에 수렴하게 된다. 즉, 선택된 관심영역 범위 안에서 밀도가 가장 큰 곳을 검색한 후에 검색된 곳을 새로운 중심으로 설정하여 다시 검색 범위를 설정하면서 미리 설정된 임계 범위까지 중심 설정과 관심 영역의 설정 작업을 반복한다.

본 논문에서 CAMShift 알고리즘 적용을 위하여 입력 영상의 Hue 채널이 생성되면 사용자로부터 추적을 위한 객체 영역, 즉 추적 관심영역(ROI: Region Of Interest)을 입력 받아서 해당 영역을 추적하기 위한 색 확률 분포(Color Probability Distribution)를 구한다. 색 확률 분포는 추적할 객체를 구성하는 색상의 히스토그램 분포에 따라서 입력 영상의 각 화소를 그에 상응하는 확률 값으로 변환한다. 이때 사용되는 방법이 히스토그램 역투영(Back-projection) 기법이다[8].

히스토그램 역투영 기법은  $n$  영상 위치  $x_i, i=1, \dots, n$ 와  $m$  개의 저장소(bin)을 갖는 히스토그램  $\{q_u\}_{u=1, \dots, m}$ 가 정의되었을 때, 추적 객체의 화소에 상응하여 생성되는 색 확률 분포 값은 아래 식 (3)에 의해서 생성된다.

$$p_u = \min \left[ \frac{255}{\max(q)} q_u, 255 \right] \quad (3)$$

$$\text{where, } q_u = \sum_{i=1}^n \delta c(x_i^*) - u$$

식에서  $c(\cdot)$ 는  $x_i^*$  위치의 화소를 연관된 히스토그램 저장소 인덱스  $c(x_i^*)$ 로  $R^2 \rightarrow \{1, \dots, m\}$  맵핑하는 함수이다. 이상의 처리를 거치고 나면 0부터 1까지의 확률 범위에 따라서 8 비트 Hue 채널의 각각의 화소는 0과 255 사이의 값으로 맵핑되게 되며, 이 화소값의 강도는 ROI 값과 유사도가 클수록 강한 강도(Intensity) 값을 보이게 된다. 따라서 이 색 확률 분포 값에 따라서 Mean Shift 알고리즘을 적용하여 원하는 손가락 객체를 추적한다.

색 확률 분포가 완성되면 손가락 객체의 움직임에 따른 객체의 중심 값을 새로 계산하고, 그에 따라 Mean Shift 알고리즘에

서 사용되는 탐색 윈도우(Search window)의 중심값을 재설정한다. 탐색 윈도우의 중심 값  $(x_c, y_c)$ 은 아래 식 (4)를 이용하여 계산한다.

$$x_c = \frac{M_{10}}{M_{00}}; \quad y_c = \frac{M_{01}}{M_{00}} \quad (4)$$

여기서,

$$M_{00} = \sum_x \sum_y I(x,y), \quad M_{10} = \sum_x \sum_y xI(x,y), \\ M_{01} = \sum_x \sum_y yI(x,y)$$

이다. 식에서  $M_{00}$ 는 0차 모멘트이고,  $M_{10}$ 와  $M_{01}$ 는 각각  $x, y$ 에 대한 1차 모멘트이다. 그리고  $I(x,y)$ 는 탐색 윈도우 상의  $(x,y)$  화소 위치에서의 강도 값을 의미한다.

연속적으로 제공되는 비디오 프레임에 대해서 위 알고리즘을 적용하고, 실시간으로 손가락의 움직임 방향이 검출되면 사용자 화면의 그 움직임을 표시하면서 사용자의 게임 명령을 대기한다.

### III. 이중 원형 추적에 의한 손동작 인식

실시간으로 모니터에 제시되는 손가락 움직임을 보면서 사용자는 손가락 개수의 변화에 의한 게임 조작 명령을 입력한다. 본 논문에서는 손동작에 의한 게임 컨트롤러의 구현을 위해서 우리가 이전에 연구한 이중 원형 추적법(Double Circle Tracing Method)[9]을 이용하여 (그림 2)와 같은 처리 과정으로 손가락 개수와 형상을 인식한다.

손동작 인식을 위한 전처리 단계로서 손 영역을 추출하기 위해서는 밝기 변화에 민감한 RGB 칼라모델 보다 밝기 변화의 영향을 효과적으로 감소시키기 위하여 밝기(Y: Luminance)와 색차신호(Cb: 푸른색의 강도, Cr: 붉은색의 강도)에 기반을 두는 YCbCr 컬러모델로 변환하여 사용한다.

그러나 YCbCr 컬러모델을 이용하여 손 영역의 살구색(Flesh color)에 해당하는 값으로 임계처리하면 RGB 칼라 모델보다는 나은 성능을 보이지만 밝기에 의한 영향으로 인해 주변 잡음이 남게 된다. 따라서 이러한 잡음을 제거하기 위해서 영상 내의 잡음 제거 또는 구성 요소들의 결합 또는 분리에 효율적인 모폴로지 변환(morphological transformation)의 팽창(dilation)과 침식(erosion) 연산을 이용한다[10].

$$I(x,y) = \underset{(x',y') \in K}{MIN} I_S(x+x', y+y') \quad (5)$$

$$I_D(x,y) = \underset{(x',y') \in K}{MAX} I_S(x+x', y+y') \quad (6)$$

식 (5)와 (6)에서  $I_S(\cdot)$ 는 입력 영상의 화소값을 의미하며,  $I_E(\cdot)$ 과  $I_D(\cdot)$ 는 각각 침식과 팽창 변환된 영상을 나타낸다. 그리고  $MIN(\cdot)$ 와  $MAX(\cdot)$ 는 각각 형태소  $K$ 에 상응

하는 화소들 가운데 최대값과 최소값을 추출하는 함수이다.

잡음이 제거된 영상이 생성되면, 손가락 개수를 인식하기 위한 이중 원형 추적법을 적용하기 위해서 먼저 이중 원의 중심으로 사용되는 손 영역의 중심값을 계산한다. 손 영역의 중심값  $C(x,y)$ 는 앞의 Mean Shift 알고리즘의 식 (4)를 이진영상에 적용한 아래 식 (7)로 계산할 수 있다.

$$C(x,y) = \left( \begin{array}{c} \left( \sum_x \sum_y xB(x,y) \quad \sum_x \sum_y yB(x,y) \right) \\ \left( \sum_x \sum_y B(x,y) \quad \sum_x \sum_y B(x,y) \right) \end{array} \right) \quad (7)$$

식에서  $B(x,y)$ 는 임계 처리된 이진영상을 의미하기 때문에 손 영역으로 인지된 화소 부분만 계산에 포함된다. 손 영역의 중심을 획득한 후에는 이중 원형 추적을 위한 두 원의 반경을 설정한다.

두 원의 반경은 단일 카메라에서 거리에 의한 손 영역의 크기 영향을 제거하기 위해서 손 영역의 중심에서부터 손 영역의 우측 끝까지의 거리를 정규화 상수  $N_C$ 로 이용하여 아래 식 (8)에 의해서 손가락 제스처 인식이 되는 손 영역을 정규화(normalize)한다.

$$S_N = S_R \frac{N_C}{N_R}, \quad \text{where, } S_R = \text{height or width} \quad (8)$$

식 (8)에서  $S_N$ 은 정규화 후의 손 영역의 높이(height) 또는 폭(width)을 나타내고,  $S_R$ 은 실제 카메라 거리에 의해서 다양한 크기로 획득된 높이 또는 폭을 의미한다. 그리고 정규화 상수  $N_C$ 는 손 영역의 중심값으로부터 우측 끝까지의 거리를 의미하는데, 이것은 손가락 형상의 변화와 상관없이 그 값의 차이가 없기 때문에 사전에 지정된 값을 사용할 수 있다.

따라서 정규화 과정 후에는 카메라 거리에 상응하여 확대 혹은 축소된 정규화된 손 영역 영상의 획득이 가능하며 사전 실험에 의해서 설정한 이중 추적원의 반경을 모든 영상에 대해서 수정 없이 적용할 수 있다.

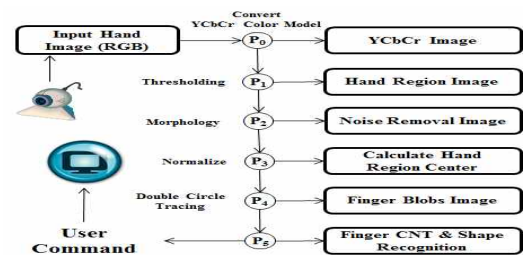


그림 2. 손가락 제스처 명령 인식 과정  
Fig. 2. Finger-Gesture Command Recognition Task.

정규화 후에 손가락 인식을 위해서는 반지름의 길이가 다른 두 개의 원을 사용하여 손가락 끝점의 단서(clue)가 되는 흔적(Blob)들을 검색한다. 이 방법은 단일 원형 추적에 의해서 발생하는 잡음에 의한 원의 교차 문제를 교차점이 아닌 이중 원에 의한 영역의 흔적으로 검색하기 때문에

오 검출되는 경우를 효과적으로 줄일 수 있다.

(그림 3)은 손 영역의 중심과 반지름의 길이가 다른 두 개의 원(각각 반지름이  $r$ ,  $r'$ 인 원)을 이용한 이중 원형 추적에 의해서 손가락 단서 영역을 추출한 예이다. 손가락 단서 영역이 추출된 후에는 이 단서 영역으로부터 손가락의 개수를 인식하고 인식된 정보를 사용자 게임 컨트롤 명령과 연동한다.

그러나 본 논문에서는 사격 게임 시뮬레이터에서 보다 사용자의 자연스러운 손가락 명령을 위하여 검지 손가락의 펴짐(조준 명령)과 굽힘(발사 명령)의 손가락 개수가 동일한 유사한 형태의 손가락 명령을 포함하고 있다. 이 두 명령을 분류하여 인식하는 문제는 손가락 제스처의 명령 입력 영상이 전 방향으로부터 입력되고 있기 때문에 손가락 개수에 의해서는 그 구분이 쉽지 않다.

따라서 이 조준과 발사 명령을 분류 인식하기 위해서는 단일 형상의 특징만으로는 그 한계가 있기 때문에 실제 게임 적용에 있어서 손가락 제스처 명령의 장면 전환에 의한 형상 변화 정보를 부가적으로 이용한다.

형상 변화 정보를 이용하기 위해서는 CAMShift 알고리즘의 식 (7)에 의해서 손가락 제스처 영역의 중심이 구해지면, 중심의 바로 위 부분에 검지 손가락 형상 변화 인지를 위한 검색 윈도우 영역  $w$ 를 설정한다. 그리고 아래 식 (9)의 판단 함수  $J(\cdot)$ 에 의해서 두 명령을 분류 인식한다.

$$J(w) = \varphi \left[ \sum_{x,y \in w} H(x,y) \right] \quad (9)$$

$$\text{where, } H_D(x,y) = \varphi [H_n(x,y) - H_{n-1}(x,y) - T]$$

식 (9)에서  $(x,y)$ 는 검색 윈도우  $w$ 에서의 화소의 위치이고,  $H(x,y)$ 는 해당 위치에서의 색상 값을 나타낸다. 그리고  $T$ 는 현재 프레임( $H_n$ )과 이전 프레임( $H_{n-1}$ )간의 화소 차이가 있다고 인식될 수 있는 임계값을 의미하며,  $\varphi[\cdot]$  함수는 양극형(Bipolar) 임계 계단 함수이다. 따라서 식 (9)에 의해서 검색 윈도우에서 색상의 양(+의 변화)와 음(-)의 변화를 분석할 수 있으며, 각각의 변화를 검지의 펴짐과 굽힘으로 분류 인식한다.

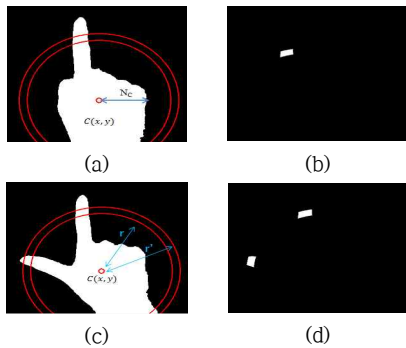


그림 3. 이중 원형 추적법의 예

- (a) 손가락 영역 추출 1 (b) 손가락 단서 영역 1
- (c) 손가락 영역 추출2 (d) 손가락 단서 영역2

Fig .3. Example of Double Circle Tracing Method.  
 (a) Finger region extraction I (b) Finger clue region I  
 (c) Finger region extraction II (d) Finger clue region II

#### IV. 게임 시뮬레이션

본 논문에서는 구현한 실시간 손동작 게임 컨트롤러의 성능 검증을 위해서 윈도우 기반의 사격 게임 시뮬레이터를 개발하였다. 입력 영상은 실시간으로 일반적 상용 웹캠인 마이크로소프트사의 LifeCam VX-800을 사용하고, 인텔사의 컴퓨터비전 라이브러리인 OpenCV를 기반으로 MS Windows 7 환경 하에서 C++언어를 사용하여 각 처리 단계를 구현하였다.

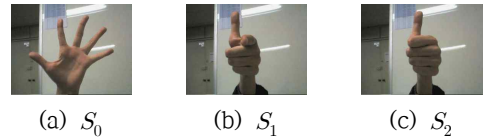


그림 4. 손동작 게임 컨트롤 명령

Fig .4. Finger-Gesture Game Control Command.

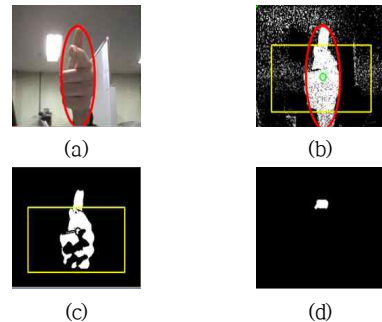


그림 5. 손동작 게임 명령 처리 과정  
 (a) 입력 영상 (b) 역투영 영상

(c) 손영역 추출 영상 (d) 명령 단서 영상  
 Fig .5. Finger-Gesture Game Command Processing Task.

- (a) Input Image (b) Back-projection Image
- (c) Hand Region Extraction Image,
- (d) Command Clue Image

구현한 사격 게임 시뮬레이터는 웹캠으로 입력 받은 사용자의 손동작 게임 명령을 CAMShift 알고리즘에 의한 검지 손가락을 추적하여 과녁에 표시하고, 손가락 개수의 변화를 사용자 명령으로 인지하여 슈팅(shooting) 명령을 수행하도록 구성되어 있다.

본 논문의 중요한 실험 목적은 특정 게임, 즉 사격 시뮬레이터에서의 사용자 편의성이 최적화된 손가락 명령의 가능성을 보이는 것이다. 따라서 실험에 있어서 정의된 사용자 명령은 손가락 개수가 차이가 많은 상이-특징(inter-feature)을 가진 제스처 명령과 손가락 개수의 차이가 없고 단지 형상의 변화만 있는 유사-특징(intra-feature)을 지닌 손가락 명령으로 설계하였다.

설계된 손동작 명령은 비교적 단순한 게임 시작( $S_0$ ). 조



준점 이동( $S_1$ ), 슈팅( $S_2$ )의 세 가지 명령으로 구성되며, 각각 게임 초기화(initialize), 마우스 이동(mouse move), 마우스 클릭(mouse Click) 이벤트와 연동된다. (그림 4)는 게임 컨트롤을 위해서 사용되는 손동작 명령의 예이고, (그림 5)는 게임 컨트롤러 처리 과정에서 발생하는 각 과정의 처리 결과 영상의 예이다. 그리고 CAMShift 알고리즘의 손가락 영역 추적을 위한 (그림 4)의 빨간색 표기 부분은 초기 관심 영역으로 게임을 시작할 때 사용자에게 의해서 게임 초기화 과정에서 마우스로 설정된다.

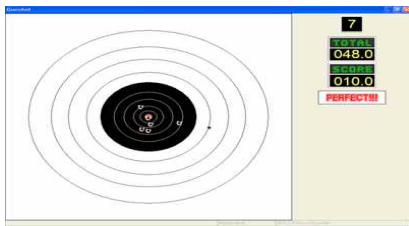


그림 6. 사격 게임 시뮬레이터  
Fig .6. Shooting Game Simulator.

표 1. 게임 컨트롤러 성능 평가 결과  
Table 1. Game Controller Performance Evaluation Result.

모드	INTER		INTRA		평균	실행 오류
	$S_0$	$S_1$	$S_2$			
인식률	100%	85%	91%	92%	10%	

성능평가 방법으로는 정의된 사용자 손동작 명령 제스처를 (그림 6)과 같이 구현한 사격 게임 시뮬레이터에서 게임 조작을 위한 각 명령 인식률을 분석하였다. 그 결과 (표 1)과 같이 평균 92%의 명령 모드 인식률과 10%의 게임 실행 오류를 나타냄으로써 게임 컨트롤러로서 성공적인 인터페이스 기능을 보였다. 여기서 게임 실행 오류는 하나의 게임이 완료되지 못하고 중간에 오류를 나타내는 것을 의미한다.

실행 오류의 발생 원인으로는 본 논문에서 사용하는 조준점 이동( $S_1$ )과 슈팅( $S_2$ ) 제스처 명령 인식이 손가락 개수의 있어서 동일하고, 단지 검지 손가락의 꺾짐과 굽힘에 의한 차이만을 인식하고 있는 유사 제스처 명령이기 때문에 색상 값에 의한 변별력이 약하기 때문이다. 그리고 두 제스처의 특징은 실영상으로부터 손가락 명령 전환에 의한 단일 형상 변화 정보를 이용하기 때문에 이전의 명령 인식 실패는 다음에 이어지는 명령의 연쇄적인 인식 실패로 연결된다. 그러나 본 논문에서는 하나의 손가락 명령을 손가락 개수가 다른 형상의 명령으로 사용하여 인식률을 높이기 보다는 사격 시뮬레이터 게임에 최적화된 자연스러운 손가락 명령의 사용에 그 의미를 두었다.

또한 단일 형상의 인식에 있어서도  $S_1$ 의 인식률이 낮은 이유는 손가락 개수에 있어서  $S_2$ 와 차이가 없고, 실제 카메

라 각도와 검지 손가락의 꺾짐 정도나 조명에 의한 색상차에 의해서 조준점 이동( $S_1$ )과 슈팅( $S_2$ ) 제스처 명령을 분류하는 검색 윈도우 상에서의 색상 변화가 뚜렷하지 않는 경우가 발생하기 때문이다.

그러나 이러한 오인식의 문제는 제스처 명령의 단일 형상에 있어서 손가락 개수가 다른 형상, 즉 주먹과 브이 같은 명령을 사용하거나, 검지 손가락에 색상 마커를 이용하면 쉽게 해결할 수 있다. 또한 실제 게임 중의 사용자 명령의 전후 형상 변화 관계를 고려하거나, 카메라를 다각도로 다중으로 사용하여 입력되는 손동작의 형상을 다중 인식 처리하면 보다 높은 성능을 보이게 된다.

### V. 결 론

본 논문에서는 사용자 손동작 기반의 게임 컨트롤러를 구현하였다. 구현된 게임 컨트롤러는 사용자 편의성뿐만 아니라, 저비용의 영상 입력 장치의 사용이 가능하다. 특히, 검증된 알고리즘을 적용하여 높은 인식 성능을 보임으로써 최적화된 게임 컨트롤러에 적용 가능성이 크다.

또한 본 논문에서는 게임 컨트롤러로서의 유효성을 보이기 위해서 구현한 게임 시뮬레이터에 최적화된 명령 모드만을 지원하고 있으나, 이 문제는 명령 모드의 정의와 입력 장비의 다중화 특히, 저비용 카메라의 다중 사용에 의해서 효율성을 높일 수 있다.

그리고 본 논문에서 제시된 방법에서의 탐색 윈도우의 크기 설정 문제, 정확한 검지 손가락 끝점 인식, 검지 손가락의 꺾짐과 닫힘 인식, 손가락 추적과 손동작 인식에 있어서의 중복성 제거를 통한 연동 문제 등의 부분적 성능 향상이 연구된다면, 향후 본 논문의 손동작 게임 컨트롤러의 유효성은 매우 클 것으로 예상된다.

### 참 고 문 헌

- [1] R. Z. Khan and N. A. Ibraheem, "Survey on Gesture Recognition for Hand Image Posture", Computer and Information Science Vol. 5, No. 3, pp. 110-121, 2012.
- [2] Hasan, M. M., Mishra, P. K. "Brightness factor matching for gesture recognition system using scaled normalization", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No. 2, pp. 35-46, 2011.
- [3] S. Mitra and T. Acharya, "Gesture Recognition: A Survey", IEEE Trans. on SMC Vol. 37, No. 3, pp. 311-324, 2007.
- [4] Y. Cheng, "Mean shift, mode seeking, and clustering", IEEE Trans. PAMI, 17:790-799, 1995.
- [5] Intel Corporation, Open Source Computer Vision Library Reference Manual, 123456-001, 2001.
- [6] Bradski, G. R. "Computer vision face tracking for use

- in a perceptual user interface”, Intel Technology Journal, 2<sup>nd</sup> Quarter, 1998.
- [7] A. R. Smith, “Color Gamut Transform Pairs”, SIGGRAPH 78, pp. 12-19, 1978.
- [8] John G. Allen, Richard Y. D. Xu, Jesse S. Jin, “Object tracking using CamShift algorithm and multiple quantized feature spaces”, Proceeding VIP '05 Proceedings of the Pan-Sydney area workshop on Visual information processing, Australian Computer Society, pp. 3-7, 2004.
- [9] M. S. Jang and W. B. Lee, “Implement of Finger-Gesture Remote Controller using the Moving Direction Recognition of Sing Shape”, The Journal of The IIBC, Vol. 13, No.4, pp. 91-97, 2013.
- [10] R. M. Haralick, et al, “Image analysis using mathematical morphology”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No.4, pp. 532-550, 1987.



**이우범(Woo-Beom Lee)**

正會員

1995년 2월 영남대 컴퓨터공학과(공학사)

1997년 2월 영남대 컴퓨터공학과(공학석사)

2000년 8월 영남대 컴퓨터공학과(공학박사)

2000년 3월~2004년 2월 대구과학대 컴퓨터공학과 교수

2004년 3월~2007년 2월 영남대 전자정보공학부 특임교수

2007년 3월~현재 상지대 컴퓨터정보공학부 교수

※주관심분야 : 뉴로비전컴퓨팅, IT융합의료기(한방의료영상), 제스처인식