CamShift 알고리즘의 Hand Tracking 기법을 응용한 Hand Motion 입력 장치 시스템

전유나*, 김수지**, 이창훈***, 김형률****, 이성구*****

요약

기존의 대표적인 입력장치는 키보드, 마우스 등으로 한정적이었으나 최근 들어 사용자들의 다양한 요구에 따라 새로운 형태의 입력장치들이 개발되는 추세이다. 이러한 추세에 맞춰 특수한 device 없이 영상상의 hand motion을 분석해 명령을 부여하는 새로운 형태의 입력장치를 제안한다. Cam-Shift 기법으로 skin color 영역을 이진화 하여 tracking 한 후, labeling을 통해 분리한 손가락 영역과 손 중심점과의 각도를 동서남북으로 구분해 counting하여 손동작을 인식한다. 손동작에 대한 입력은 맨손에 배경처리를 하지 않은 경우 약 76.8%의 낮은 인식률을 보였으나, 붉은색 장갑을 착용하고 배경을 지정해 줄 경우 잡영 제거의 영향으로 인식률이 90.2%까지 향상된다.

키워드 : CamShift 알고리즘, 핸드트래킹, 핸드모션, OpenCV

The input device system with hand motion using hand tracking technique of CamShift algorithm

Yu-na Jeon*, Soo-ji Kim**, Chang-hoon Lee***, Hyeong-ryul Kim****, Sung-koo Lee*****

Abstract

The existing input device is limited to keyboard and mouse. However, recently new type of input device has been developed in response to requests from users. To reflect this trend we propose the new type of input device that gives instruction as analyzing the hand motion of image without special device. After binarizing the skin color area using Cam-Shift method and tracking, it recognizes the hand motion by inputting the finger areas and the angles from the palm center point, which are separated through labeling, into four cardinal directions and counting them. In cases when specific background was not set and without gloves, the recognition rate remained approximately at 75 percent. However, when specific background was set and the person wore red gloves, the recognition rate increased to 90.2 percent due to reduction in noise.

Keywords : CamShift algorithm, Hand Tracking, Hand motion, OpenCV

1. 서론

※ 교신저자(Corresponding Author): Sung-koo Lee

Received : January 26, 2015

Revised : February 26, 2015

Accepted : February 28, 2015

* Hanshin University, School of Computer Engineering

email: wjsdbk545@nate.com

** Hanshin University, School of Computer Engineering

email: todo1302@nate.com

*** Hanshin University, School of Computer Engineering

email: leech230@naver.com

초기 컴퓨터로부터 현재까지 주요 입력장치로 키보드와 마우스를 이용하였으며, 최근에는 터치 방식의 입력방법이 등장하였다. 그러나 현재에는

**** Hanshin University, School of Computer Engineering

email: hyungryul@naver.com

***** Hanshin University, School of Computer Engineering

email: sklee@hs.ac.kr

■ 본 연구는 한신대학교의 2015학년도 교내연구비 지원에 의해 수행되었음

키보드, 마우스, 터치에 의한 입력 방법보다 사람의 손가락 동작을 인식하여 사람이 직접 컴퓨터를 제어할 수 있는 방법들이 제시되기 시작했다.

이에 본 논문에서는 CamShift 알고리즘의 Hand Tracking 기법을 응용하여 손동작을 인식하고 세부 동작(입력)에 따라 단축키 기능을 대신할 수 있는 OpenCV 기반의 입력 장치 시스템을 제안한다. 영상처리를 통하여 원격 제어와 같은 다양한 상황에서의 키보드 및 마우스를 제어할 수 있는 즉, 손 추적과 손동작 인식을 이용하여 키보드 대신 컴퓨터를 제어할 수 있는 HDS (Handmotion Device System) 시스템을 구현하였다.

2. 관련연구

2.1 모션 인식의 대표적인 시스템

영상처리 분야의 발전으로 얼굴인식, 홍채인식, 지문인식 방법이 생활 속에 다양하게 접목되었다. 손동작 인식에 의해 게임 시스템에 응용되는 대표적인 장치인 Wii 와 KINECT는 컨트롤러라는 별도의 디바이스가 존재하여야 작동이 가능하다. 이렇듯 손동작 인식은 현재 게임 디바이스 분야에서 많이 응용되고 있지만 PC 활용을 목적으로 하는 시스템에서 찾기는 어렵다[1,2,7].

2.2 HSV color 모델을 이용한 영역 추출

행렬로 입력받은 영상정보에는 영상의 RGB 색상 값이 저장된다. 그러나 RGB색상으로는 skin color를 추출하기 어려우며, 한정된 피부색의 범위로 인해 미세한 오차에도 예민하게 반응할 수 있다. 같은 피부색일지라도 흑인과 백인, 황인에 따라서 오차가 존재하고, 영상의 밝기에도 영향을 받는다. 이를 극복하기 위해 인종의 차이나 영상의 밝기에 비교적 민감하지 않은 HSV color model을 사용하는 것이 바람직하다.

HSV color model은 색상(Hue), 채도(Saturation), 밝기(Value)로 구성된다. 색상은 그 색의 원색을 나타내고, 채도는 색의 순수도를 나타내는 것으로 원색에 어느 정도의 흰색이 혼합되었는지를 나타낸다. 밝기는 색상의 밝은 정도를 나타낸다. HSV 컬러모델은 (그림 1)과 같이 원뿔

모양의 좌표계로 표현된다. 색상은 0도에서 360도 사이의 각도로 표현된다. 0도는 빨강색, 120도는 초록색, 240도는 파란색을 각각 나타낸다. 채도는 0에서 1까지의 값을 가지며 원뿔 중심으로부터의 수평거리 (z축과의 거리)로 표현된다. 명도는 세로축에 해당하며, 가장 아래쪽이 명도 0인 검정색을 나타내고, 가장 위쪽이 명도 1인 흰색을 나타낸다.

(Figure 1) HSV color model

이러한 HSV의 특징을 이용할 경우 채도와 밝기를 무시하고 순수한 원색의 포함 정도를 사용하여 오차를 줄일 수 있다. 사람의 피부는 핏줄의 영향으로 인해 일정한 양의 붉은 색을 포함하고 있는데, 어두운 살색(흑인 또는 어두운 곳)과 밝은 살색(백인 또는 밝은 곳) 모두 채도와 밝기를 제외하면 같은 영역의 붉은 원색에 포함되어 있기 때문이다. 본 논문에서는 채도와 밝기를 일정한 값으로 한정해 주고, Hue채널에서 붉은색 영역의 범위를 지정하여 피부색을 추출한다.

2.3 CamShift 알고리즘에 의한 관심영역 추출

Mean-Shift 알고리즘은 데이터 집합의 밀도 분포(특징점, 코너, 색상)를 기반으로 관심영역(ROI) 객체를 고속으로 추적하는 알고리즘으로 초기의 검색 영역의 크기와 위치를 지정하면 반복되는 색 분할 계산에 의해서 색상 클러스터가 발생되고 초기 지정한 색 영역에 기반 하여 경계를 결정하여 관심 물체를 추출할 수 있다(그림 2)[4].

그러나 Mean-Shift 알고리즘의 영상 처리에는 차분 영상을 이용한 방법과 optical flare 벡터를 이용하는 방법이 있지만, 이러한 방법은 대부분 이미 획득한 비디오 스트리밍 혹은 고정된 카메라로부터 획득한 비디오 스트리밍 상에 존재하는 객체의 움직임을 추적하는데 주안점을 두고 있다. 따라서 객체가 이동하고 카메라도 이동하는 경우에는 적용할 수 없는 부분적인 문제점이 있다. 그러므로 객체가 이동함에 따라 카메라의 방향도 이동하는 환경에 응용할 수 있도록 이동 객체의 움직임을 검출하고 이동 객체를 따라 카메라의 방향을 이동함으로써 지속적으로 객체의 움직임을 추적 감시할 필요가 있다. OpenCV에서는 이를 위한 대안으로 객체추적을 위한 알고리즘인 CAMShift를 구현한 함수인 cvCamShift ()함수를 제공하고 있다(그림 3)[3].

(그림 2) Mean-Shift

(Figure 2) Mean-Shift

(그림 3) CamShift 알고리즘의 동작

(Figure 3) Operation of CamShift Algorithm

3. 연구내용

3.1 추출 영역의 동작 인식

3.1.1 이중 Labeling에 의한 손가락 개수 Counting

추출 영역은 흰색 물체의 외곽을 추출해오기 때문에 손 모양에 대한 구분이 되는 것은 아니다. 따라서 행렬로 저장된 흰색 픽셀을 부분적으로 가져와 동작을 구분할 필요가 있다. 본 논문에서는 손가락 개수를 count하고 손가락의 방향 정보를 얻어와 이를 구분한다.

(그림 4) 레이블링과 카운팅

(Figure 4) Lablieng and Counting

Labeling된 손 중심에서 가장 긴 세 번째 손가락 기준으로 모든 손가락이 잡힐 수 있고 손바닥 부분은 잡히지 않을 크기, 대략(2/3)로 (그림 4)의 빨간 부분과 같이 원을 그린다. 원을 기반으로 원과 흰색을 갖는 픽셀들이 겹치는 부분들을 다시 각각 Labeling하면 손목과 펼쳐있는 손가락 마디의 개수의 count가 가능하다. 여기서 손목을 손가락 count에서 제외하기위해 Labeling된 각각의 사각형의 넓이를 구한 뒤 값이 가장 큰 사각형을(주황색 테두리) 손목으로 간주하여 제거한다.

3.1.2 손가락 방향 정보에 의한 동작 인식

<표 1> 손가락 방향 정보의 예

east1	east0	east1
west1	west1	west1
north0	north0	north0
south0	south1	south3

<Table 1> examples of finger direction information

labeling된 손가락 마디 영역의 좌표와 손 중

심 좌표를 이용하여, 손 중심에 대한 마디 영역의 상대적 위치를 구해 동작을 구분한다. labeling된 한 사각형 영역의 첫 점의 좌표가 기본 값으로 설정되는 것을 이용하여 손가락 마디 영역의 좌표를 labeling 사각형의 첫 점으로 지정하였다. 우선 손의 중심을 기준으로 하여 동서남북으로 방향을 나눈 뒤, 각 방향에 들어오는 손가락의 개수를 count한다. 예를 들어 <표 1>의 세 번째 손동작은 손의 중심을 기준으로 서쪽에 한개, 북쪽에 3개, 동쪽에 1개의 손가락이 count 된다.

(그림 5) 삼각함수의 이용

(Figure 5) Using of Trigonometrical Function

손마디 영역의 상대적 방향은 (그림 5)와 같이 삼각함수를 이용하여 구한다. 마디 영역의 좌표(IX,IY)와 손 중심 좌표(mX,mY)가 그리는 삼각형의 대각선 길이를 피타고라스 정리($\sqrt{(IX-mX)^2+(IY-mY)^2}$)를 이용해 구하고, Y좌표의 차(IY-mY)로 높이를 구한다. 높이를 대각선으로 나누어 $\sin\theta$ 를 구한다. $\sin 45^\circ \approx 0.7$ 로 가정할 경우 북쪽($45^\circ \sim 135^\circ$) 영역의 범위는 $0.7 \leq \sin\theta \leq 1$ 이 된다. 남쪽($225^\circ \sim 270^\circ$)은 그 반대에 위치하므로 $-1 \leq \sin\theta \leq -0.7$ 이 되고, $-0.7 \leq \sin\theta \leq 0.7$ 의 범위에는 동쪽과 서쪽 영역이 속한다. 동, 서는 여기서 X좌표 차(IX-mX)의 부호로 한 번 더 구분해 준다. X좌표의 차가 양수일 경우 동, 음수 일 경우 서가 된다.

그러나 입력 영상의 행렬식은 좌우를 반전해 값을 넘겨주는 특성이 있으므로 동과 서, 북과 남이 반대인 결과가 출력된다. 뽀렘 연산의 부호 값이 모두 반대로 나오기 때문이다. 이를 해결하기 위해서 입력영상을 반전한 후 count 하는 방

법과, count 값의 좌우와 상하를 반전해 출력하는 방법이 있다. 본 논문에서는 출력영상의 가시성을 높이기 위해 전자의 방법을 사용하였다.

3.2 동작 정보에 의한 명령어 전달

3.2.1 프레임 누적 정보에 의한 인식 판단

동작이 구분된 후 실제로 명령을 전달하려면 해당 동작이 유효한 동작인지 판단할 필요가 있다. 따라서 영상에 들어오는 손의 모양 중 지정된 데이터에 해당하는 정보와 일정 시간 일치할 경우 유효한 동작이라고 판단한다.

(그림 6) 프레임 비교

(Figure 6) Comparison of frame

각도에 따른 방향정보를 이용하여 프레임의 비교를 통해 일정 시간 일치하는 지 확인한다. (그림 6)를 보면 A에 기존의 방향정보를 입력시키고 B에는 비교할 프레임의 방향정보를 입력시켜 A와 B의 값을 대조한다. 이러한 비교에서 참 값이 계속 유지할 경우 동작으로 인식한다. 본 시스템에서는 1초간 10프레임을 받아오기 때문에 동작 인식 유효시간을 1.5초 즉, 15프레임으로 지정하였다. 아래 (그림 7)에서와 같이 15프레임동안 같은 동작의 방향정보가 계속된다면 동작을 인식한다.

(그림 7) 프레임 누적

(그림 7) Accumulation of frame

각도를 이용한 방향정보를 저장하여 비교하기 때문에 미세한 손의 떨림에도 동작의 연속적인 유지를 받아들이는 반면 손이 아닌 잡영으로 labeling 되는 경우에는 방향정보와 일치하는 데이터가 없는 것으로 판단되어 동작을 인식하지 않는다.

3.2.2 방향 정보에 의한 Key event 명령

Key event는 사용자가 키보드와 같은 입력 장치로 컴포넌트에 키를 입력 하였을 경우 발생하는 이벤트이다. Key event는 키가 눌러지거나(Pressed), 키가 눌러졌다 떼어지거나(Released), 키가 입력(Typed)될 때 List를 제외한 컴포넌트 객체에 의해 생성 된다. 키가 입력되는(Key Typed) 이벤트는 고수준의 이벤트이며 싱글키 입력('a') 이나 연속 키 입력('shift' + 'a')에 의해 발생 된다. 키가 눌러지거나(Key Pressed) 떼어지는(Key Released)이벤트는 저수준의 이벤트로서 키보드 레이아웃이나 플랫폼에 의존한다. 다음은 Key event의 주요 상수 필드와 메서드이다.

본 논문에서는 손의 중심위치에서 각도를 계산 한 후 동, 서, 남, 북 네 방향의 손가락 개수를 count 한다. 그리고 네 방향의 count 된 수의 만큼 Key Event를 발생시킨다. 예를 들어 동=1, 북=1을 count하게 되었을 경우 동=1, 북=1에 해당하는 Key Event를 발생하여 키보드를 컨트롤 한다.

프레임의 누적 정보와 일치하는 키 명령을 찾아 실행하는 컨트롤에서 발생하는 순서도는 아래 (그림 8)과 같다.

(그림 8) 프레임과 키 명령의 흐름도

(Figure 8) Flow chart of frame and key command

4. 구현 결과

4.1 시스템 구성도

(그림 9) 시스템 구성도

(Figure 9) System diagram

1. 사용자는 Hand-Motion에 대한 단축키를 개인의 필요에 맞게 변경한다.
2. 웹캠과 시스템을 연결한 후 웹캠으로부터 영상자료를 받아온다.
3. HSC로 변환한 영상의 Hue값에서 살색을 추출하여 이진화 한다.
4. Cam-Shift 알고리즘을 이용하여 손의 중심값을 찾고 관심영역을 설정한다.
5. 관심영역 안에서 이중 레이블링을 통해 손가락의 개수를 카운팅하고, 손목을 제거한다.
6. Tracking한 손의 중심과 레이블링한 손가락 사이의 각도를 삼각함수를 이용하여 계산하고, 계산 결과에 따른 동작을 인식한다.
7. 프레임의 누적시켜 잡영을 제거하고 인식된 동작에 해당하는 단축키를 응용프로그램에 전달한다.

4.2 구현 내용

프로그램의 왼쪽 상단 부분에는 손가락으로 만들 수 있는 간단한 동작을 리스트 컨트롤로 나열하였고 왼쪽 하단 부분에는 사용자가 해당 동작에 대한 단축키 등을 설정할 수 있는 기능을 구현하였다. 프로그램의 오른쪽 상단 부분에는 카메라를 영상으로 연결하는 connect 버튼을 통해 연결의 활성화 기능을 추가하였다. (그림

10)은 동0서0남0북1 의 방향정보를 갖는 볼륨업 동작이 인식되는 것을 보인다. 동작이 인식되면 동영상이나 음악 플레이어의 볼륨이 컨트롤 될 수 있다.

(그림 10) 프로그램 화면

(Figure 10) Screen of Program

4.3 인식률 테스트

인식률은 각 항목에 대해 100번의 시도를 하여 성공 횟수로 계산 하였다. 맨손의 경우 배경 잡영의 영향을 크게 받아 인식률이 평균 76.8%에서 그쳤으나, 배경을 지정해 주거나 붉은 장갑을 착용할 경우 인식률이 90%에 가까운 결과를 보였다.

(그림 11) 손동작 예시

(Figure 11) example of handmotion

* 왼쪽상단->오른쪽->왼쪽하단->오른쪽 순서

<표 2> 인식을 변화

* 단위 : % (100회 기준)

	1	2	3	4	5	6	A
Type 1	74	79	70	78	78	82	76.8
Type 2	77	85	73	88	86	90	83.2
Type 3	75	87	72	88	84	91	82.8
Type 4	86	92	80	94	90	96	89.7

Type 1 : Bare hand
 Type 2 : Red gloves
 Type 3 : Bare hand + Specifying background
 Type 4 : Red gloves + Specifying background
 A : Average

<Table 2> Changes of recognition rate

5. 결론

영상처리의 가장 흥미로운 분야 중 하나인 영상인식의 응용 기술로 모션 인식, 홍채 인식, 얼굴 및 표정 인식, 지문 인식, 제스처 인식 등이 있으며 최근에 개인의 독특한 신체적 특징이나 습관 등을 이용하여 개인의 신원을 확인하는 방법인 생체인식(Bio Metrics)에 대한 관심과 중요성이 더욱 증대되고 있다.

본 논문에서는 영상처리를 통해 OpenCV를 이용하여 손동작을 인식하는 입력장치 시스템 HDS(Handmotion Device System)은 구현하였다. 기존 손동작 인식 프로그램은 PC환경이 아닌 별도의 디바이스로 구동이 되어 게임 분야 혹은 모바일 분야에서 활용이 되었으나 본 논문에서 구현된 HDS 시스템은 윈도우 PC 환경에서 구동된다. 또한 사용자 마다 다른 피부색을 인식하기 위해 독립적인 HSV컬러의 Hue값을 이용하여 흑인, 백인, 황인에 구분 없이 피부색을 검출할 수 있으며, 정확한 손가락 개수의 count를 위해 손바닥의 중심위치를 기점으로 labeling하여 손가락 마디를 제외한 손목을 제거함으로써 정확도를 높였다. labeling 값에 따른 손동작 인식은 동, 서, 남, 북 네 방향으로 범위를 정하여 범위에 들어오는 손가락 개수를 통해 구분하였다.

본 논문에서 제안된 시스템은 다음과 같은 기대효과를 가질 수 있다.

- 동영상 시청 시 원거리에서 손동작만으로 컴퓨터 볼륨 제어, 재생, 되감기

- 사용자에게 의한 단축키 직접 설정, 변경 손동작에 적용하여 간단한 동작만으로 빠르게 키 입력 전달
- 사용자가 필요한 입력 명령어들을 모두 추가한다면 키보드나 마우스 없이도 입력장치의 역할 대행
- 손동작 입력에 의한 새로운 개념의 게임 프로그램 개발

향후 연구로 HDS 시스템이 배경에 대한 잡영 제거를 통해 손동작 인식률을 높인다면 이는 내장 캠을 통해서 키보드나 마우스를 통한 컨트롤을 대신 할 수 있을 것이다. (그림 11)에서와 같이 단지 손 모양을 바꿈으로써 PC 환경에서 동영상 시청하거나 혹은 문서작업을 하는 경우 Power On/Off 또는 Volume Up/Down 등을 자유롭게 이용할 수 있다. 이는 PC에 부속되어 있는 기존 입력장치인 키보드나 마우스로부터의 해방을 의미한다. 이렇듯 기본적인 연구들이 밀받침된다면 최근 대두되고 있는 다양한 모션 인식 분야에서 인간의 편의를 위한 손동작 인식 분야의 다양하고 정확한 모션 인식이 가능해질 것이다.

References

- [1] Bai yu, Sang-Yun Park, Yun-Sik Kim, In-Gab Jeong, Soo-Yol Ok, Eung-Joo Lee, Park Sang-Yun, Kim Yun-Sik, Jeong In-Gab, Ok Soo-Yol, Lee Eung-Joo, "Hand tracking and hand gesture recognition for human computer interaction", Korea Multimedia Society Conference, vol.14, no.2, pp.182-193, 2011
- [2] Gwangha Jeon, Jongseok Um, "(A)Study on online real-time strategy game by using hand tracking in augmented reality", Korea Multimedia Society Conference, vol.12, no.12, pp.1761-1768, 2009
- [3] Ho-Sik Park, Cheol-soo Bae, "Face and Hand Tracking Algorithm for Sign Language Recognition", Communications and Networks, vol.31, no.11, pp.1071-1076, 2006
- [4] Shan Caifeng, Tan Tieniu, Wei Yucheng, "Real-time hand tracking using a mean shift embedded particle filter", Pattern Recognition, vol.40, no.7, pp.1958-197

0, 2006

- [5] Yonjae Kim, Simon Leonard, Azad Shademan, Axel Krieger, Peter C. W. Kim, "Kinect technology for hand tracking control of surgical robots: technical and surgical skill comparison to current robotic masters", Surgical Endoscopy, vol.28, no.6, pp.1993-2000, 2013
- [6] Kwang-woon Choi, Da-woon Jeong, Suk-han Lee, Jong-soo Choi, "Interaction Augmented Reality System using a Hand Motion", Korea Multimedia Society Conference, vol.15, no.4, pp.425-438, 2012
- [7] Sae-Bom Lee, Il-Hong Jung, "A Design and Implementation of Natural User Interface System Using Kinect", Journal of Digital Contents Society, vol.15, no.4, pp.473-480, 2014

전 유 나

1991년 5월 31일 서울 생
2010년: 작전여자고등학교 졸업
2015년: 한신대학교 컴퓨터공학부 졸업

김 수 지

1991년 11월 28일 인천 생
2010년: 신명여자고등학교 졸업
2015년: 한신대학교 컴퓨터공학부 졸업

이 창 훈

1987년 8월 17일 서울 생
2006년: 독산고등학교 졸업
2015년: 한신대학교 컴퓨터공학부 졸업

김 형 룰

1989년 8월 12일 수원 생
2008년: 권선고등학교 졸업
2015년: 한신대학교 컴퓨터공학부 졸업

이 성 구

1980년: 중앙대학교 전자계산학과
학사

1989년: 중앙대학교 전자계산학과
공학석사 - 인공지능

1993년: Arizona State University,
공학석사

1998년: Arizona State University, 공학박사 - 소프
트웨어공학

1999년~현재: 한신대학교 컴퓨터공학과 교수

관심분야: 소프트웨어공학, 재사용, 컴포넌트개발, 검
색시스템, 유비쿼터스 컴퓨팅, 라이브러리시스템