비전 기반 제스처 인식을 이용한 사용자 인터페이스 구현

고민삼 ○, 이광희, 김창우, 안준호, 김인중
한동대학교 전산전자공학부
e-mail : {komiins, gist1226, ahimahas }@naver.com
swanjh@nate.com, ijkim@handong.edu

An Implementation of User Interface Using Vision-based Gesture Recognition

Min-Sam Ko ○, Kwang-Hee Lee, Chang-woo Kim, Jun-Ho Ahn, In-Jung Kim
School of Computer Science & Electronic Engineering
Handong Global University

요 약

컴퓨터 비전을 기반으로 한 영상처리 및 제스처인식 기술을 이용하여 편리하게 응용프로그램을 제어 할 수 있는 인터페이스를 구성하였다. 카메라로 얻어진 영상을 HSV 색상계로 변환 후 색상 정보를 이용하여 손 영역을 추출한다. 추출된 손 영역의 무게중심과 크기 변화를 추적하여 제스처를 인식한다. 인식한 제스처는 응용프로그램의 동작을 제어하거나 마우스의 입력을 대체하는데 사용된다. 이는 별도의 입력장치 없이 컴퓨터 비전 기술만으로 제스처를 인식하여 응용프로그램을 실행할 수 있어 자연스러우며 사용자 친화적인 인터페이스 환경을 제공한다.

Keywords: 비전, 제스처 인식, 패턴 인식, HSV

1. 서 론

컴퓨터의 빠른 발전과 카메라의 대중화로 인하여 영상처리 기술이 적용되는 분야가 늘어나고 있다. 카메라만을 사용하여 제스처를 인식하는 방법을 비전 기반 제스처 인식이라고 한다[1]. 이러한 비전 기반 제스처 인식은 자연스러우며 사용자 친화적인 인터페이스를 구성 할 수 있어서 유비쿼터스 시스템의 인터페이스로도 유용하게 사용될 수 있다.

본 논문에서는 비전 기반의 제스처 인식을 이용하여 사용자가 편리하게 응용프로그램을 제어할 수 있는 인터페이스를 제안한다. 영상을 HSV 좌표계로 변환한 후 색상정보를 이용하여 손의 영역을 추출하고 추출한 손의 크기와 무게중심의 좌표 변화를 추적하여 제스처로 인식한다. 인식된 제스처의 정보는 응용프로그램을 제어하는데 사용된다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개한다. 3장에서는 영상에서 손의 영역을 추출하고 추적하는 방법에 대해서 설명한다. 4장에서는 인식의 대상이 되는 제스처를 정의하고 제스처를 인식하는 방법에 대해 설명한다. 5장에서는 3, 4장에서 소개한 제스처 인식 기술을 이용하여 응용프로그램을 제어하는 사용자 인터페이스를 소개한다. 6장에서는 실험 결과와 분석을 서술하며, 7장에서 결론을 맺는다.

2. 관련 연구

비전 기반 제스처 인식을 위해서는 손 영역 추출 및 추적과 손 궤적으로부터 제스처를 인식하는 단계가 필요하다. 김태수 외 1명은 RGB 좌표 계를 임계값을 가지고 이진화 하여 손 영역을 추출 및 추적하는 방법을 제안하였다[2]. 최유주 외 4명은 HIS 색상공간에서 색상(Hue) 및 색상 기울기(Hue-Gradient)를 기반으로 정의된 배경모델을 구축하고, 실시간으로 입력되는 영상과의 배경차분(background subtraction)을 이용하여 손 영역을 추출하여 추적하는 방법을 제안하였다[3]. 김종민 외 1인은 2차원 형상의 불변량에 해당하는 경계선의 방향성 히스토그램을 이용하여 손 영역을 추출 및 추적하는 방법을 제안하였다[4]. 환영환은 체인코드와 무게중심 프로필 방법을 이용하여 제스처를 인식하는 방법을 제안하였다[5]. 이용재 외 1인은 연속성을 가지는 모델영상들을 HMM을 사용하여 포즈들의 시공간적 특성을 매칭함으로써 제스처를 인식하는 방법을 제안하였다[6].

3. 손 영역추출 및 추적

손 영역추출 및 추적이란 영상 중 우리가 사용 할 손의 영역을 추출하고 궤적을 추적하는 과정으로써 제스처 인식을 위한 정보를 제공하는 단계이다. 카메라를 통해 입력되는 영상 정보는 연속된 영상들의 스트림으로 구성된다. 이 때 영상 정보는 조명에 민감해진다. 이와 같은 문제를 줄이기 위해 본 논문에서는 RGB 좌표 계에서 HSV 좌표 계로 변환하여 조명 변화의 적응력을 개선하였다. H,S,V는 각각 색상,채도,명도를 의미한다. [4]

RGB 좌표 계는 다음 식 (1)에 의해 HSV 좌표 계로 변환 한다.

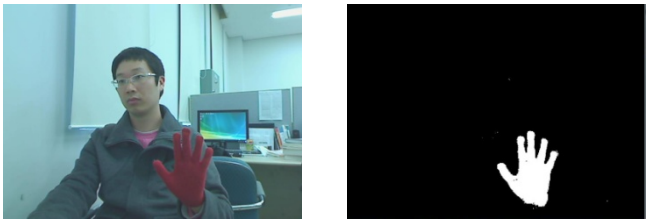
$$\mathrm{H} = \cos^{-1}\left(\frac{\frac{1}{2}[(\mathrm{R}-\mathrm{G})+(\mathrm{R}-\mathrm{B})]}{[(\mathrm{R}-\mathrm{G})^2+(\mathrm{R}-\mathrm{B})(\mathrm{G}-\mathrm{B})]^{1/2}}\right)$$

$$\mathrm{S} = 1 - \frac{3}{(\mathrm{R}+\mathrm{G}+\mathrm{B})}[\min(\mathrm{R},\mathrm{G},\mathrm{B})] \qquad (1)$$

각 화소의 H의 값과 S의 값에 대한 임계 값을 설정하여 식 (2)와 같은 규칙에 의해 손 영역을 추출한다.

$$(\theta_{\mathrm{low}}^{\mathrm{H}} < H < \theta_{\mathrm{high}}^{\mathrm{H}}) \,\&\, (\theta_{\mathrm{low}}^{\mathrm{S}} < S < \theta_{\mathrm{high}}^{\mathrm{S}}) \qquad (2)$$

식 (2)의 $\theta_{\mathrm{low}}^{\mathrm{H}}$은 H의 최소 임계값이며 $\theta_{\mathrm{high}}^{\mathrm{H}}$은 H의 최대 임계 값이다. $\theta_{\mathrm{low}}^{\mathrm{S}}$와 $\theta_{\mathrm{high}}^{\mathrm{S}}$은 각각 S의 최대, 최소 임계값이다. 각 임계 값은 실험으로 설정하였다.



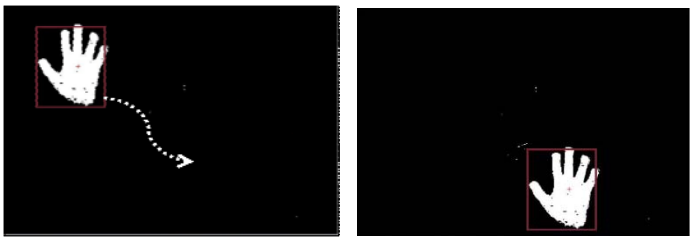
(a) (b)

<그림 1. 입력된 영상 및 손 추출한 결과>

그림 1(a)는 입력된 영상이며 그림 1(b)는 그림 1(a)에서 손을 추출한 결과이다.

명도에 비교적 안정적인 HSV 좌표 계를 쓰지만 연속적인 색상 정보가 시간에 따라 변화하기 때문에 이진화된 영상의 정보도 연속적으로 변화하게 된다. 이 때 생기는 노이즈로 인해 추출 오류가 일어날 수 있다. 이와 같은 문제를 줄이기 위해 본 논문에서는 Glassfire 알고리즘을 사용한다. Glassfire 알고리즘은 이진화된 영상을 레이블링하기 위해 많이 사용되는 방법 중 하나이다[7]. Glassfire 알고리즘은 손 추출된 결과에서 손 영역을 추적 할 수 있으며, 레이블링하는 수치가 임계 값보다 적으면 손 영역으로 추적하지 않고 노이즈로 인식하기 때문에, 노이즈를 제거하는 효과를 볼 수 있다. 이로써 잡영에 둔감하면서도 효율적으로 손 영역을 추적할 수 있다.

하지만 카메라 영상의 화소수가 많을 경우 계산량이 증가하게 된다. 위와 같은 문제를 개선하기 위해 본 논문에서는 블록화를 사용하여 계산량을 줄였다. 블록화는 M*N의 영상을 p*q(1≤p≤M, 1≤q≤N) 크기로 다시 나누는 것을 말한다 [8]. 한 블록 안에는 (M/p)*(N/q)개의 화소로 구성된다. 각 블록 내에서 식 (2)를 만족하는 화소의 수가 임계값보다 크다면 해당 블록을 손 영역으로 판단한다. 이로써 손 영역으로 판단한 블록만 레이블링 하면 되기 때문에 계산량이 감소한다.



(a) 손의 궤적 (b)추적된 손

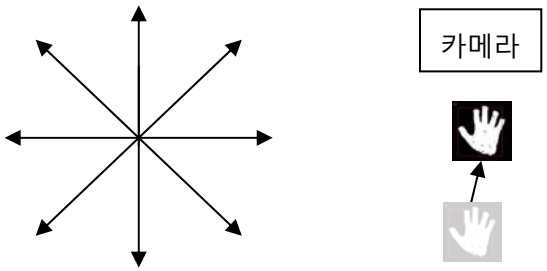
<그림 2. 손을 추적하는 모습>

그림 2(a)는 손의 궤적을 나타낸 사진이며, 그림 2(b)는 손 영역을 추적한 결과이다.

4. 제스처 인식

제스처를 인식하기에 앞서 사용할 제스처를 정의하였다. 제스처를 정의할 때 다음과 같은 것을 고려하였다. 정의할 제스처는 사용자에게 쉬워야 하는 동시에 시스템이 인식하기 좋아야 한다.

본 논문에서는 그림 3(a)같이 일반제스처를 정의하였다. 일반 제스처는 상, 하, 좌, 우, 우상, 우하, 좌상, 좌우 등 8방향의 동작을 말한다.



(a) 일반 제스처 (b) 시작 제스처

<그림3. 본 논문에서 정의한 제스처>

하지만 같은 동작이라도 목적에 따라 다르게 해석할 수 있는 경우가 있다. 예를 들어 사용자가 오른쪽으로 움직였을 때, 이것은 제스처일 수도 있고 의미 없는 동작일 수도 있다. 본 논문에서는 그림 3(b)와 같이 손을 앞으로 내미는 동작을 시작 제스처로 정의하여 사용함으로써 이러한 모호성을 해결하였다. 시작 제스처는 모든 일반 제스처가 인식하기 전에 먼저 인식되어야 하는 제스처이다.

제스처를 인식하는 과정은 다음과 같다. 시작 제스처가 인식되면 일반 제스처를 인식한다. 만약 시작 제스처가 인식된 후, 제한된 시간 안에 일반 제스처가 인식되지 못하면 일반 제스처의 인식을 중지하고 시작 제스처를 기다린다.

제스처는 손 추적을 통해 얻어진 x 좌표와 y 좌표, 그리고 손 영역의 크기의 변화로부터 인식한다. 각 순간 손의 영역 P_t는 식 (3)과 같이 세 가지 정보로 구성되어 큐에 저장된다. 큐는 연속적으로 추출된 손 영역을 임시로 저장하는 공간으로 사용된다.

$$P_t = (x_t, y_t, a_t) \quad (3)$$

식 (3)의x_t, y_t, a_t는 각각 시간 t일 때의 x, y 좌표, 그리고 추적된 손의 크기를 의미한다. 알고리즘 (1)에서처럼 좌표의 변화는 없이 넓이가 증가하였을 경우, P_t는 큐에 저장된다. 만약 조건을 만족시키지 못하면 큐는 초기화된다. $|x_t-x_{t-1}|$와 $|y_t-y_{t-1}|$은 현재 좌표와 이전 좌표의 변화 절대 값이며, 이 값이 각각 임계 값 θ_{error} 보다 적으면 변화가 없다고 인식한다. $|a_t-a_{t-1}|$은 넓이의 변화로서 임계 값 θ_{max}보다 클 경우 의도적인 움직임으로 해석하여 큐에 P_t를 저장한다.

알고리즘 (1)

Do loop
 IF ($|x_t-x_{t-1}|$, $|y_t-y_{t-1}| < \theta_{error}$ && $|a_t-a_{t-1}| > \theta_{max}$)
  THEN (Put P_t into Queue)
 ELSE
  THEN (Initialize Queue)

큐는 증가하는 손 영역을 담고 있다. 큐의 처음과 끝에 있는 손 영역이 식 (4)를 만족할 경우 손을 앞으로 내미는 제스처로 판단한다. a_0/a_t은 큐의 처음과 끝의 면적 변화 비율 값이며, 이 값이 임계 값 θ_{rate}보다 크고, 현재 손 영역의 크기가 임계 값 θ_{area}보다 크면 손을 앞으로 내미는 동작인 시작 제스처를 인식한다.

$$(a_0/a_t > \theta_{rate}) \& (a_t > \theta_{area}) \quad (4)$$

시작 제스처가 인식되면 큐는 초기화 되고 일반 제스처를 인식하기 위해 연속적으로 들어오는 손 영역 정보를 저장하기 시작한다. 일반 제스처는 식 (5), 식 (6)에 따라 인식된다. 먼저 식 (5)에서 Δx, Δy은 처음 큐에 저장된 손 영역의 x, y좌표와 현재의 x, y좌표와의 차이이

다. Δx, Δy를 가지고 처음 저장된 손 영역을 원점으로 하여, 현재 손 영역이 움직인 각도 θ를 구한다. θ의 값을 가지고 그림 3(a)와 같이 정의한 8가지 방향을 구분하여 동작을 인식한다.

$$\Delta x = x_{queue[0]} - x_{queue[t]}$$

$$\Delta y = y_{queue[0]} - y_{queue[t]}$$

$$\theta = \tan^{-1}\frac{\Delta y}{\Delta x} \quad (5)$$

하지만 식 (5)에서 사용한 θ는 큐의 처음과 끝에 저장된 좌표만 이용하기 때문에 그 사이 동작에 따라 오차가 발생할 수 있다. 식 (6)은 이러한 오차에 대한 해결방법으로, $C(x)$와 $C(y)$는 큐에 저장된 첫 번째 손 영역을 기준으로 하여 t시간까지 x 좌표와 y좌표의 변화를 나타낸다.

$$C(x) = \sum_{k=1}^{t}(x_{queue[0]} - x_k)$$

$$C(y) = \sum_{k=1}^{t}(y_{queue[0]} - y_k) \quad (6)$$

예를 들어 왼쪽으로 움직이다 오른쪽으로 크게 움직였을 경우에는 인식하지 않아야 한다. 하지만 각도만으로 인식하면 큐의 처음과 끝에 저장된 좌표 비교를 통해 오른쪽 제스처로 인식하게 된다. 각도와 변화량을 함께 사용하면 왼쪽 움직임으로 인해 $C(x)$가 감소하게 된다. 감소된 $C(x)$가 정의한 임계 값을 넘지 못하면 동작은 인식되지 않는다.

5. 제스처를 이용한 사용자 인터페이스 구현

인식된 제스처를 이용하여 응용프로그램을 제어할 수 있는 사용자 인터페이스를 구현하였다. 제안하는 인식기를 이용하여 실제 응용프로그램을 제어하기 위해서는 먼저 그림 3(a)의 일반 제스처를 응용프로그램의 명령어에 대응시켜야 한다. 본 시스템에서는 시작 시 응용프로그램들을 등록하도록 구성하였다.

이와 같이 등록된 응용프로그램은 대기상태에서 입력된 상, 하, 좌, 우의 제스처에 의해 실행된다. 특정 응용프로그램 활성화 후에는 그림 3(a)의 일반 제스처를 인식하여 사용자가 지정한 명령을 수행한다. 각 방향에 관한 명령은 Shift, Ctrl, Art 키와 키보드 한자리 키의 조합으로 구성할 수 있게 하였다.

제안하는 사용자 인터페이스에서는 제스처 외에도 손의 움직임을 추적하여 마우스 커서를 모사할 수 있게 하였다. 이 때, 손의 움직임에 따라 마우스 커서가 이동하여야 하는 때와 그렇지 않을 때로 나누어야 한다. 이는 손의 크기인 a_t가 임계 값 이내에 있을 때 영상을 추적하여 커서로서 사용하게 되며, 임계 값을 넘어갈 때는 노이즈로 인식하여 마우스 커서로 인식하지 않는다. 또한 마우스 추적 기능을 활성화, 비활성화 형식으로 설정할 수 있게 하였다. 마우스 추적을 비활성화한 상태에서는 손의 움직임에 따라 마우스를 모사하지는 않지만, 영상과 제스처는 계속 인식하므로 추적을 비활성화한 상태에서도 사용자가 제스처를 입력하면 응용프로그램 제어가 가능하다.

6. 실험 결과 및 분석

6.1. 실험 환경

영상 입력을 위하여 Logitech Quickcam Pro 9000을 이용하였고, 손 추출의 인식률을 높이기 위해 빨간색 장갑을 착용하였다. 제스처는 카메라로부터 1m 떨어진 지점에서 촬영되었다. 실험에 사용한 조명은 표 (1)과 같으며, 각각 조명에 대하여 8방향 제스처를 각각 300번씩 실시하였다.

	어두움	보통	밝음
밝기(LUX)	70	200	1,000
밝기 예	호텔복도	시청각실	일반사무실

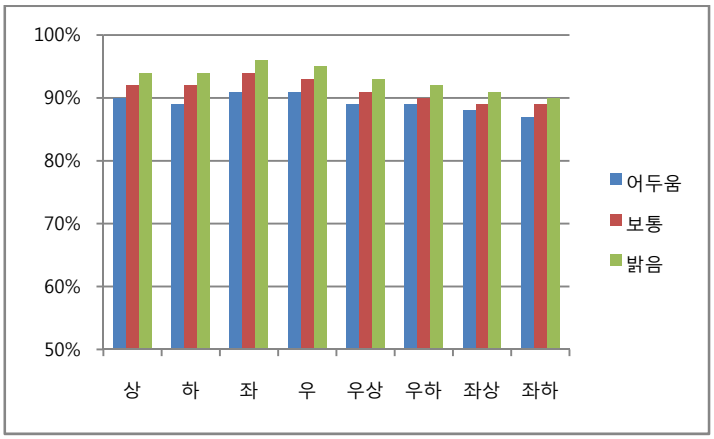
<표 1. 실험에 사용한 조명의 구성>

6.2. 실험 결과 및 분석

그래프1의 결과에 의하면 각 8방향의 제스처 중 가장 잘 인식된 좌방향은 평균 96%, 가장 인식되지 않은 좌하방향은 평균 87%의 인식률을 보였다. 이는 8방향의 제스처가 고르게 인식됨을 알 수 있다. 또한 그림4의 결과에 의하면 조명이 밝을 때는 인식률이 최고 96%, 최

하 90%를 유지하여 조명이 밝을 때는 잘 인식됨을 알 수 있었다.

그러나 조명이 어두울 때 인식률은 평균 89.2%로 인식률이 최고 91%, 최하 87%를 유지하는 것을 알 수 있다. 이는 조명 변화에 민감하지 않음을 알 수 있다.



<그림4. 조명에 따른 8방향의 인식률>

7. 결 론

본 논문은 비전 기반의 제스처 인식을 이용하여 응용프로그램을 제어 할 수 있는 인터페이스를 제안하였다. 사용자가 특정 입력 장치를 사용하지 않고 비전 기반의 제스처나 마우스 이동을 모사하여 응용프로그램을 제어할 수 있는 인터페이스를 구현하였다. 손 영역 추출 및 추적은 카메라에서 입력받은 영상을 HSV로 변환 후 색상 정보를 이용하여 손을 추출하고 Glassfire 알고리즘을 사용하여 손을 추적한다. 제스처는 손을 내미는 동작을 시작제스처로 인식하고 시작제스처가 인식된 후 일반제스처가 인식된다. 일반제스처는 8가지 방향 동작으로 정의하였는데 추후 이들을 조합할 경우 도형 등 더 복잡한 제스처로 확장 할 수 있다.

참고문헌

[1] A. Mulder, "Hand gesture for hci," Technical Report 96-1, Simon Fraster University, 1996.

[2] 김태수, 전중창, "화상처리에 의한 한국어수화인식시스템 개발을 위한 인식 방법," 02 한국멀티미디어학회 추계학술대회논문집, pp.69-72, 2002.

[3] 최유주, 이제성, 유효선, 이정원, 조위덕, "조명 변화에 안정적인 손 형태 인지 기술," The KIPS Transactions: PartB,1598-284X, 제15B권1호,pp.25-36, 2008.

[4] 김종민, 이칠우, "윈도우 플레이어 제어를 위한 에지 방향성 히스토그램 손 형상 인식," 한국정보과학회 03 가을학술발표논문집(2), pp.628-630, 2003.

[5] 환영환, "움직임 정보를 이용한 제스처 인식 시스템," The KIPS Transactions:PartB,1598-284X, 제10B권4호,pp.473-478, 2003.

[6] 이용재, 이칠우, "저차원 특징 공간에서 HMM을 이용한 제스처 인식," 2001년도 추계 학술발표 논문집 pp.849-853, 2001.

[7] 강동중, 하종은, Visual C++을 이용한 디지털 영상처리, 사이텍 미디어, pp.250-259 2005.

[8] 윤영근, 이석룡, 박호현, 정진완, "그레이 영상에서의 영역 확장에 기반한 영상 세그멘테이션 기법," 한국정보과학회논문지,데이터베이스 제32권 제1호, pp1-12, 2005.