산업기술연구(강원대학교 산업기술연구소 논문집), 제29권 A호, 2009.
Journal of Industrial Technology, Kangwon Natl. Univ., Korea, No. 29 A, 2009.

USB 카메라 영상에서 DP 매칭을 이용한 사용자의 손 동작 인식

Hand Gesture Recognition using DP Matching from USB Camera Video

하 진 영* 변 민 우** 김 진 식**
Ha, Jin-Young Byeon, Min-Woo Kim, Jin-Sik

Abstract

In this paper, we proposed hand detection and hand gesture recognition from USB camera video. Firstly, we extract hand region extraction using skin color information from a difference images. Background image is initially stored and extracted from the input images in order to reduce problems from complex backgrounds. After that, 16-directional chain code sequence is computed from the tracking of hand motion. These chain code sequences are compared with pre-trained models using DP matching. Our hand gesture recognition system can be used to control PowerPoint slides or applied to multimedia education systems.

We got 92% hand region extraction accuracy and 82.5% gesture recognition accuracy, respectively.

키워드 : *손 동작 인식, 체인코드, 동적 패턴 정합, 피부 색상*
Keywords : *hand gesture recognition, chain code, DP matching, skin color*

1. 서론

현대 기술이 발전함에 따라 보다 편리한 방법으로 컴퓨터와 사용자가 상호작용할 수 있는 많은 기술들이 연구되고 있다. 상호작용의 방법으로는 크게 음성을 이용하는 방법과 영상을 이용하는 방법이 있다. 현재 음성을 이용하는 방법은 어느 정도 안정화를 이루었으며, 실용화된 제품들도 많이 나오고 있다. 하지만 음성을 이용하는 방법의 경우 주변의 소음이 많은 공간에서는 사용하기 어려운 단점이 있다. 영상을 이용한 방법은 입력된 영상에서 사람의 몸짓이나 손, 얼굴 등을 인식하여 사람과 컴퓨터가 상호 작용하는 방법이다. 이 중 표현력이 뛰어나고 널리 이용되는 것이 손동작을 사용하는 것이다. 손동작은 손의 위치와 모양에 의해 정의되며 Hand Posture와 Hand Gesture로 분류할 수 있다. Hand Posture는 관찰되는 한 시점에서 손과 손가락의 위치와 모양에 의해 정의되어지고, Hand Gesture는 연속된 Hand Posture의 패턴에 의해 정의된다.

기존의 손동작 인식의 방법으로는 데이터 장갑(data glove) 같은 장비를 이용한 방법과 장비를 이용하지 않고 카메라를 통해 입력받은 영상을 이용하여 인식하는 방법이 있다. 장비를 이용한 방법은 사용자가 특수한 장갑을 착용하고 장갑에 부착된 센서로부터 입력된 신호를 이용하여 손동작을 인식하거나 손에 마크를 부착하여 마킹점을 추출함으로써 손동작을 인식하는 것이다[1].

본 논문에서는 특별한 장비를 이용하지 않고 카

* 강원대학교 IT특성화학부대학 컴퓨터학부 교수, 교신저자
** 강원대학교 IT특성화학부대학 전기전자정보통신공학부 학부과정

메라를 통해 입력받은 영상을 이용하여 실시간으로 손동작을 인식할 수 있는 방법에 대해 연구하였다. 연구한 내용을 바탕으로 실제 응용 프로그램을 구현하여 실생활에 적용하고 향후 기술을 적용한 활용 가능성에 대해 알아본다. 본 논문에서 설명하는 손동작 인식 시스템의 구성을 간략히 나타내면 그림 1과 같다.

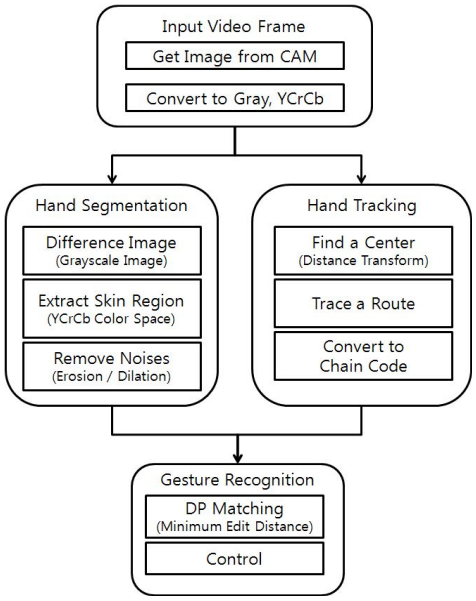


그림 1. 손동작 인식 시스템 구성도

본 논문의 구성은 다음과 같다. 본 논문의 2장에서는 카메라에서 입력받은 영상으로부터 손 영역을 검출하는 방법에 대해서 알아본다. 3장에서는 검출한 손 영역을 추적하는 방법에 대해 기술하였으며 4장에서는 손동작을 추적한 경로를 이용하여 체인코드를 구성하고 DP Matching을 적용하여 실제 동작을 인식하기 방법에 대해 알아본다. 5장에서는 본 논문에서 언급한 기술을 바탕으로 구현한 어플리케이션에 대하여 설명한다. 6장에서는 논문에서 제안한 손동작 인식 시스템에 대한 실험 결과와 문제점 및 향후 연구 방향에 대해 기술한다.

2. 손 영역 검출

손 영역을 검출하기 위한 가장 간단한 방법은 색상 공간에서의 피부색을 정의하고 범위에 해당하는 값을 피부색으로 판별하는 것이다. 본 논문에서도 이와 같은 방법을 이용하였으며, 사람의 배경에 피부색으로 정의된 색상이 존재할 경우 인식률을 떨어뜨릴 수 있으므로 차영상을 이용하여 배경에 피부색이 존재하더라도 실제 피부 색상만을 검출할 수 있도록 하였다.

2.1 차영상을 이용한 움직임 감지

차영상은 원 영상에서 대상 영상 간의 뺄셈 연산을 수행한 결과로서, 영상 내에 있는 물체의 움직임 감지 및 추적에 주로 이용된다. 이 방법을 이용하여 피부색 영역을 검출하기 위한 전처리 과정을 수행한다. 현재 프레임에 대해 차영상을 구하는 과정을 수행함으로써 보다 배경에서 피부색 영역을 제외한 정확한 손 영역을 검출할 수 있다.

(1) 명암도(grayscale) 영상으로 변환

카메라로부터 입력받은 영상은 3채널 RGB 이미지 이다. 차영상을 구하는데 있어서 3채널 RGB 이미지에서는 256^3레벨을 갖기 때문에 뺄셈 연산을 수행하는 것은 정확하지 않은 결과가 나올 수 있다. 따라서 RGB 색상 이미지를 1채널 명암도 영상으로 변환하여 수행하면 보다 정확하게 차영상을 구할 수 있다.

식 (1)은 RGB 색상에서 명도를 계산하여 그레이로 변환하는 일반적인 방법이다. 본 논문에서도 이것을 이용하여 그림 2와 같이 컬러 영상을 그레이 영상으로 변환하였다.

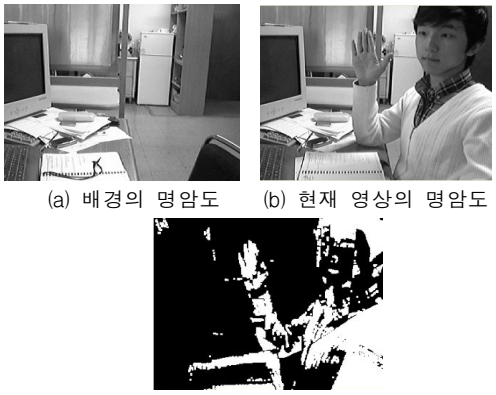
$$Gray(x,y) = 0.299 * R + 0.587 * G + 0.114B \quad (1)$$

그림 2. RGB 이미지를 Gray 이미지로 변환

(2) 차영상

사용자의 움직임을 감지하기 위한 차영상을 얻기 위해서는 먼저 배경 영상을 지정해야 한다. 미리 지정된 배경 $Bg(x,y)$에서 현재 프레임 $F_i(x,y)$과의 차이가 임계치(Threshold) 이상일 경우 사용자의 움직임이 감지된 영역이라고 판단하며, 그 외에는 움직임이 없다고 판단한다. 이 방법으로 이진화된 i번째 프레임의 차영상 $Diff_i(x,y)$을 구하는 방법은 식 (2)에 나타나 있다.

$$Diff_i(x,y) = \begin{cases} 1, & |Bg(x,y) - F_i(x,y)| \geq threshold \\ 0, & otherwise. \end{cases} \quad (2)$$



(a) 배경의 명암도 (b) 현재 영상의 명암도

(c) 차영상을 적용한 이진화

그림 3. 차영상을 적용한 이진화

2.2 피부색 영역 검출

손의 위치를 찾기 위한 많은 방법이 연구되었지만 실시간 인식 시스템에 적용하기 위해서는 연산량을 적게 필요로 하는 방법을 사용해야 한다. 따라서 본 논문에서는 특정 색상공간에서 피부색 영역의 경계선을 정의하는 방법을 이용하여 피부색 영역을 검출한다. 이 방법은 영상을 변환한 후, 정의된 피부 색상 값과 비교하여 피부색을 판별하기 때문에 연산이 적고 실시간 인식과 같은 빠른 시스템을 구현해야 하는 인식 기술에서 많이 응용되고 있다.

피부색 영역을 정의하여 손이나 얼굴을 검출하는 방법은 연산량이 적고 구현이 간단하다는 점 때문에 RGB, YCrCb, HSV 등 여러 색상 공간에서의 피부색 영역을 정의하고자 많은 연구 논문들이 발표되었다[2].

(1) RGB

빛의 3원색(Red, Green, Blue)을 이용하여 영상을 표현한 방법으로 카메라에서 영상을 입력받는 기본 값이다. 따라서 특별한 변환 없이 사용할 수 있다는 장점이 있다. 식 (3)을 통해 피부색 영역을 정하여 이진화 된 영상 $Skin_i(x,y)$를 얻을 수 있다.

$$Skin_i(x,y)=\begin{cases}1, R>95 \text{ and } G>40 \text{ and } B>20 \text{ and} \\ \quad \max(R,G,B)-\min(R,G,B)>15 \\ \quad \text{and} |R-G|>15 \text{ and } R>G \\ \quad \text{and } R>B \\ 0, \ otherwise\end{cases} \tag{3}$$

(2) HSV

HSV는 색조(Hue), 채도(Saturation), 농도(Value)를 이용하여 색을 표현하는 방법이다. 외곽선 검출 등 Value값이 중요한 곳에서 많이 사용되며, 식 (4)을 이용하여 RGB영상을 HSV영상으로 변환할 수 있으며, 식 (5)는 피부색 영역 $Skin_i(x,y)$을 구하는 식이다.

$$\begin{aligned} &V=\max(R,G,B) \\ &S=\begin{cases}(V-\min(R,G,B))*255/V, \text{ if } (V\neq 0). \\ 0, \ otherwise.\end{cases} \\ &H=\begin{cases}(G-B)*60/S, \text{ if } V=R \\ 180+(B-R)*60/S, \text{ if } V=G \\ 240+(R-G)*60/S, \text{ if } V=B\end{cases} \\ &\text{if } H<0 \ then \ H=H+360\end{aligned} \tag{4}$$

$$Skin_i(x,y)=\begin{cases}1, \ H\leq 19 \text{ and } S\geq 48 \\ 0, \ otherwise\end{cases} \tag{5}$$

(3) YCrCb

YCrCb는 이미지 압축 업무에서 주로 사용되는 압축된 비선형 RGB 신호이다. 컬러는 luma(비선형 RGB로부터 계산된 휘도, Y)로 표현되고 RGB의 가중된 요소들의 합과 RGB red와 blue 요소에서 luma를 뺀 Cr과 Cb 요소로 구성된다. 식 (6)을 이용하여 RGB영상을 YCrCb영상으로 변환할 수 있다.

$$\begin{aligned} &Y=0.299*R+0.587*G+0.114*B \\ &C_r=(R-Y)*0.713+\delta \\ &C_b=(B-Y)*0.564+\delta\end{aligned} \tag{6}$$

2.3 노이즈 제거

차영상 $Diff_i$와 피부색 영역 $Skin_i$는 픽셀 정보에 기반하여 이진화를 수행한 것이므로 노이즈가 발생하기 쉽다. 이를 제거하기 위하여 침식과 팽창 연산을 이용하였다.

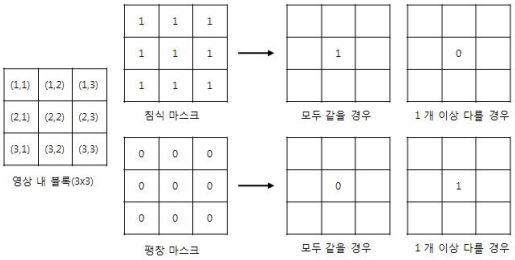


그림 4. 침식 연산과 팽창연산

침식 연산은 영상 내 블록에서 침식마스크의 해당 블록과 비교하여 일치할 경우 1로 하고, 한 개 이상 불일치 할 경우 0으로 설정한다. 팽창연산은 영상 내 블록에서 팽창 마스크의 해당 블록과 비교하여 일치할 경우 0으로 하고, 한 개 이상 불일치 할 경우 1로 설정한다. 그림 4는 침식연산과 팽창연산의 동작을 설명하고 있다. 침식 연산을 수행하게 되면 이진화된 영상에서 객체를 축소시키는 효과를 얻게 되는데 이를 이용해서 노이즈를

제거할 수 있으며, 팽창연산은 침식연산과 반대로 이진화된 영상에서 객체를 확대시키는 효과를 얻을 수 있다.

침식→팽창연산의 과정을 거치면 노이즈를 제거하면서 원하는 부분을 확장 시킬 수 있다. $Diff_i$와 $Skin_i$에 대해서 침식과 팽창 연산을 통해 노이즈를 제거하고 피부 영역을 확장 시킬 수 있다.

(a) RGB 이미지-1

(b) 피부색 영역 $Skin_a$

(c) 침식 연산 수행

(d) RGB 이미지-2

(e) (d)에 대한 침식 연산

그림 5. 피부색 영역과 침식 연산으로 노이즈 제거

그림 5(b)는 식 (4)를 적용하여 피부색 영역을 검출한 것이고, 그림 5(c)는 피부색 영역에 대하여 침식연산으로 노이즈를 제거한 그림이다.

2.4 차영상과 피부색 영역을 이용한 손 영역 검출

차영상 $Diff_i$와 피부색 영역 $Skin_i$의 이용하면 배경에 구애받지 않는 손 영역을 검출할 수 있다.

손 영역 $Hand_i(x,y)$는 식 (7)과 같이 정의한다.

$$Hand_i(x,y) = Diff_i(x,y) \wedge Skin_i(x,y) \qquad (7)$$

프레임의 각 (x,y)에 대해서 $D_i(x,y)=1$이고 $S_i(x,y)=1$인 부분은 현재 움직임이 있으며 피부색 영역이라고 정의할 수 있으므로 손 영역일 가능성이 높다고 판단할 수 있다.

그림 6은 그림 5(d)에 대하여 식 (7)을 적용한 그림이다.

그림 6. 그림 5(d)에 대한 $Hand_i$

3. 손의 움직임 추적

영상에서 손 영역을 검출하고 손동작을 인식하기 위해서는 우선 추적해야 할 손의 중심점을 찾고, 특정 프레임 간격마다 구한 손의 중심점을 바탕으로 인식을 위한 16방향의 체인코드(Chain code)를 생성할 수 있다.

3.1 손 후보 영역에서 손의 중심점 찾기

차영상과 정의된 피부색을 이용하여 손 영역 을 검출한 후, 손의 움직임을 추적하기 위해서는 손 영역의 중심을 찾아야 한다. 손 영역의 중심을 구하기 위해서, 거리변환(distance transform)을 이용하여 손의 중심을 구할 수 있다. 거리변환은 이미지상의 한 픽셀에서 가장 가까이 놓인 물체의 경계까지 거리를 구하는 연산이다. 표 1의 거리변환 행렬을 이용하여 나온 결과 값들 중에서 가장 큰 값의 좌표를 찾으면 손 영역의 중심을 찾을 수 있다[4].

표 1. 거리변환행렬[5,6]

4.5	4	3.5	3	3.5	4	4.5
4	3	2.5	2	2.5	3	4
3.5	2.5	1.5	1	1.5	2.5	3.5
3	2	1	0	1	2	3
3.5	2.5	1.5	1	1.5	2.5	3.5
4	3	2.5	2	2.5	3	4
4.5	4	3.5	3	3.5	4	4.5

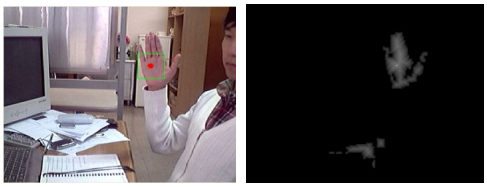


그림 7. 거리변환을 이용한 손 중심 찾기

3.2 중심점을 연결하는 손의 좌표 추적

거리 변환을 이용하여 손의 중심점을 찾은 다음 특정 프레임 간격으로 좌표들을 이동 경로에 추가

한다. 손의 중심점을 찾는 과정에서 노이즈 때문에 손이 아닌 부분을 손의 중심으로 인식하는 경우가 있다. 이 경우는 이전 좌표의 손의 위치와의 거리를 비교해서 특정 거리 보다 클 경우에는 손의 경로 추적에 포함시키지 않았다.

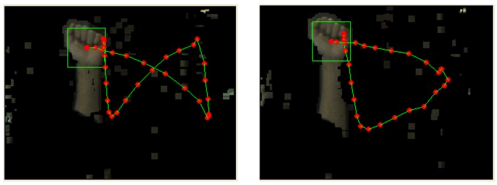


그림 8. 손의 위치를 추적

본 논문에서는 손의 위치를 추적하는 과정에서 인식하기 위한 손동작을 구분하기 위해 손이 특정 시간동안 정지해 있는 것으로 인식할 행동의 시작과 끝을 구분하였다. 그림 8은 손을 추적하면서 손의 중심점을 연결하는 좌표들을 추적하는 과정을 그림으로 나타낸 것이다.

3.3 체인코드로 변환

손의 중심을 추적한 좌표들의 경로를 바탕으로 인식에 필요한 16방향을 갖는 체인코드로 변환한다. 16방향에 대한 코드는 그림 9와 같이 정의하며, 0번에서 부터 1씩 증가할 때마다 방향 코드의 각도는 22.5°씩 증가한다.

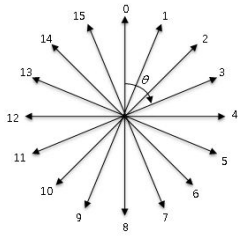


그림 9. 체인 코드를 구성하는 16방향 코드

손의 이동 좌표가 $p_i=(x_i, y_i)$, $p_{i+1}=(x_{i+1}, y_{i+1})$을 이룰 때, p_i와 p_{i+1}이 이루는 각도 $\theta_i(0 \leq \theta_i < 2\pi)\theta_i$는 그림 10과 같이 나타낼 수 있으며, 식 (6)(7)을 적용하여 θ_i를 구할 수 있다.

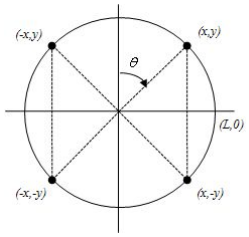


그림 10. 양의 y축으로부터 (x,y)가 이루는 각도

$$L\cos\theta = y \tag{8}$$
$$\therefore \theta = \begin{cases} \cos^{-1}\frac{y}{L}, & x \geq 0 \\ 2\pi - \cos^{-1}\frac{y}{L}, & x < 0 \end{cases}$$

$$L_{i,i+1} = \sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}$$
$$\theta_i = \begin{cases} a\cos(\frac{x_{i+1}-x_i}{L_{i,i+1}}), & x_i \leq x_{i+1} \\ 2\pi - a\cos(\frac{x_{i+1}-x_i}{L_{i,i+1}}), & x_i > x_{i+1} \end{cases} \tag{9}$$

식 (8)(9)을 이용하여 좌표가 이루는 각도 θ_i를 구하고 θ_i와 가장 근접한 방향코드(0-15)를 선택하여 체인코드에 추가한다.

16방향 코드를 이용한 체인 코드를 생성할 때에는 단위 코드 당 길이가 일정해야 인식률을 높일 수 있으며, 체인코드로부터 원래의 손의 움직임을 복원할 수도 있다. 즉, 체인코드 $Q=\{q_1, q_2, \cdots, q_{n-1}, q_n\}$에서 식 (11)을 만족해야 한다.

$$Length(p,q) = \sqrt{(p_x-q_x)^2+(p_y-q_y)^2} \tag{10}$$
$$Length(q_i, q_{i+1}) = Length(q_{i+1}, q_{i+2}), \quad 1 \leq i \leq n-2 \tag{11}$$

체인 코드를 이루는 단위 길이 L_c를 일정하게 하기 위해 p_i와 p_{i+1}의 길이를 L_c로 나누어 체인 코드를 생성하도록 하였다.

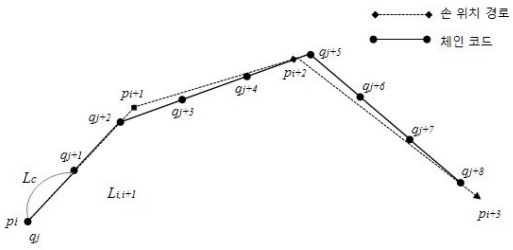


그림 11. 손의 위치 경로로 체인코드 구성

p_i와 p_{i+1}에 대하여 N_c개의 방향 코드를 생성하고, p_{i+1}은 q_{j+N_c}로 대체하여 다음 체인 코드를 생성하도록 한다.

$$N_c = \frac{L_{i,i+1} + L_c * 0.5}{L_c} \tag{12}$$

L_c의 길이는 체인코드 내에 연속적인 방향 코드에 대해서 중복되는 부분을 간략화 할 수 있는 적당한 범위 내에서 정하도록 한다.

그림 11은 식 (12)을 적용하여 손의 위치 경로를 바탕으로 체인코드를 구성한 그림이며, 그림 12는 실제 영상에서 체인코드를 생성하고 생성한 체인코드로 손 움직임의 좌표를 재구성한 화면이다. 정확히 같도록 재구성할 수는 없지만 같은 패턴으로 체인코드가 구성되었음을 확인할 수 있다.

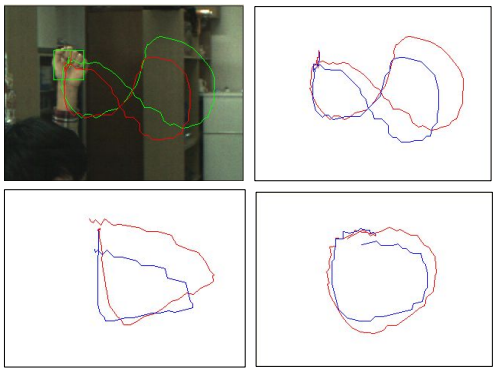


그림 12. 원래의 손 움직임과 생성된 체인코드를 이용하여 원래의 손 움직임을 재구성.

4. 손동작 인식

16 방향의 체인코드에 기반을 두어 손동작을 인식하기 위해 본 논문에서는 빠르고 쉽게 적용할 수 있는 DP Matching을 이용하였다. DP Matching은 동적프로그래밍기법을 이용한 문자열 최소 편집 거리를 계산하는 방법에 기초한다.

4.1 거리 계산

16 가지 방향 코드로 이루어진 두 체인 코드 $u = \{a_1, a_2, a_3, \cdots, a_{n-1}, a_m\}$, $v = \{b_1, b_2, b_3, \cdots, b_{m-1}, b_m\}$의 최소 거리(distance)를 정의하는 함수 $Dist(u,v)$는 다음과 같이 정의한다[7,8].

$$\textit{Initial values}: \quad \begin{cases} g_k(i,0) = i, & \text{for} 0 \le i \le n \\ g_k(0,j) = j, & \text{for} 0 \le j \le m \end{cases} \tag{13}$$

$$\textit{Recurrence Formula}: \quad g_k(i,j) = \min \begin{cases} g_k(i-1,j-1) + d(i,j) \\ g_k(i,j-1) + 1 \\ g_k(i-1,j) + 1 \end{cases} \tag{14}$$

$$Dist(u,v) = g_k(n,m)$$

식 (13)을 이용하여 초기 값을 설정하고, 식 (14)를 적용하여 동적프로그래밍을 구현한다. $Dist(u,v)$를 구하는데 드는 시간 복잡도는 $O(nm)$으로 빠른 시간에 편집 거리를 구할 수 있다. $d(i,j)$는 코드 a_i^k와 b_j가 같아지기 위한 비용을 나타낸다. 코드가 같아지기 위한 비용은 각도의 차이에 비례한다는 점에 기인하여 본 논문에서는 각도 차이에 의한 두 체인코드의 차이를 적용하였다. 구하고자 하는 두 체인 코드 간의 최소 거리는 $g_k(n,m)$을 참조하여 얻을 수 있다.

4.2 정의된 동작 별 인식 모델 생성

DP Matching을 이용하여 정의된 동작을 인식하기 위해서는 k개의 동작에 대해 r개의 인식 모델을 확보하도록 한다. 본 논문에서는 실험을 위해 8가지 동작에 대해 10개의 인식모델을 확보하여 DP Matching을 적용하였다.

체인 코드 V를 인식하기 위해 $k*r$개의 모델에 대해 Match Function을 적용하여 각 거리를 측정한다. $k*r$개의 모든 인식 모델에 대해 최소 거리를 갖는 동작 모델을 찾으면 체인코드 V에 해당하는 동작을 인식할 수 있다.

4.3 정의되지 않은 동작에 대한 인식

각 모델에 대해 Match Function을 적용하여 최소 비용을 갖는 모델을 선택한다면 정의되지 않은 동작을 입력하는 경우에도 정의된 동작에서 인식을 하는 결과를 초래 할 수 있다. 정의되지 않은 동작을 판별하기 위하여 본 논문에서는 식 (15)와 같은 정의된 동작별 인식에 허용되는 최대 비용 E_k를 정의하였다.

$$E_k = \max\{Dist(M_i^k, M_j^k), 1 \le i,j \le r\} \tag{15}$$

M_i^k는 동작 k로 정의된 i번째 체인코드를 나타낸다.

4.4 DP Matching을 이용한 동작 인식

DP Matching을 적용하여 최소 비용 $D_{\min}$을 구하고 동작 k에 대한 최대 비용 한계값 E_k를 적용하여 식 (16)의 조건을 만족할 때 체인코드 u가

정의된 동작 k로 인식 하도록 하였다.

$$D_{\min} = \min\left\{Dist(u, M_j^i), \quad (1 \leq i \leq k, 1 \leq j \leq r)\right\}$$
$$\text{단}, Dist(u, M_j^i) \leq E_k \tag{16}$$

즉, 체인 코드 u에 대해 DP Matching을 이용하여 동작 k와 가장 유사하다고 인식하였지만 $Dist(u, M_i^k) > E_k$일 경우에는 동작 인식 모델에서 허용하는 최대 비용을 초과하였으므로 정의되지 않은 동작으로 인식한다. 식 (16)에서 정의된 DP Matching을 이용한 동작 인식 알고리즘은 시간복잡도 $O(r*k*n*m)$을 요구한다.

5. 실험 및 평가

본 논문에서 제안한 피부색 검출과 동작 인식 방법을 실험하기 위해 크게 두 가지 실험을 하였다.

첫 번째 실험은 색상공간에 따른 피부색 인식률을 알아보는 것으로, 5명의 실험자가 각각 RGB, HSV, YCrCb에서의 동작실험을 실시하여 동작변화에 따른 피부색 검출이 정확하게 이루어지는지 확인하는 방법으로 실험을 진행하였다. 각 실험자별로 10번씩 실험하여 다음과 같은 결과를 얻었다.

표 2. 색상공간에 대한 정확도 실험 결과

	정상적인 동작 횟수		
	RGB	HSV	YCrCb
실험자 1	1/10	6/10	9/10
실험자 2	0/10	7/10	8/10
실험자 3	1/10	8/10	10/10
실험자 4	1/10	6/10	9/10
실험자 5	0/10	9/10	10/10
결과	3/50	36/50	46/50

RGB 색상공간의 경우 50번의 테스트에서 3번의 정상동작을 하여 6%의 정확도를 보여주었다. HSV 색상공간의 경우 50번의 테스트에서 36번의 정상동작을 하여 72%의 정확도를 보여주었다. 마지막으로 YCrCb 색상공간의 경우 50번의 테스트에서 46번의 정상동작을 하여 92%의 정확도를 보여주었다. 피부색 검출을 위해서는 휘도와 색차를 분리하여 연산하는 YCrCb 방식이 가장 정확도가 높은 것을 알 수 있는데, 이는 피부가 다른 사물에 비하여 휘도의 변화에 민감하게 반응하기 때문이다.

두 번째 실험은 본 논문에서 제안한 동작 인식의 정확도를 알아보는 것으로, 실험을 위해 그림 13과 같이 8가지 동작과 각 동작별 10가지 인식 모델을 구성하였다.

인식을 위한 실험 데이터는 본 저자가 각 동작에 대해 반복적으로 10회씩 수행하였으며, 인식을 위한 동작 인식 모델 데이터는 2명이 각각 5번씩 수행하여 8가지 동작에 대해 80개의 모델을 생성한 후 DP Matching에 적용하였다.

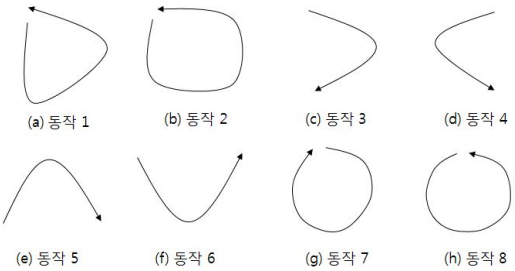


그림 13. 인식을 위한 정의된 8가지 동작

각 동작별로 10번씩 반복적인 테스트를 수행한 결과 다음과 같은 결과를 얻었다.

표 3. 정의된 동작에 대한 손 동작 인식 결과

	결과(올바른 인식/수행횟수)
동작 1	10 / 10
동작 2	7 / 10
동작 3	8 / 10
동작 4	9 / 10
동작 5	9 / 10
동작 6	8 / 10
동작 7	7 / 10
동작 8	8 / 10

정의된 동작에 대해서 반복적으로 실험한 결과 80개의 테스트 동작 중 66개를 올바르게 인식하여 82.5%의 인식률을 보였으며, 정의되지 않은 동작에 대해서 80개의 테스트 동작을 실험한 결과 53개를 정의되지 않은 동작으로 인식하여 66.25%의 인식률을 보였다.

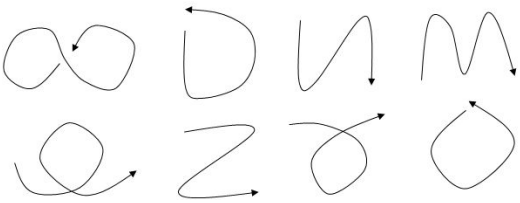


그림 14. 임의의 인식에 정의되지 않은 동작

Pentium4 3.0GHz CPU, 1.25GB Memory환경에서 개발 및 실험을 진행하였으며, 카메라는 Philips PCVC680 USB VGA Camera를 사용하였다.

6. 결론

컴퓨터의 성능 및 정보 표현에 대한 기술 발달로 인해 컴퓨터와의 상호 작용을 위한 새로운 방법들이 요구되고 있다. 손은 신체의 다른 어떤 부분보다도 움직임이 자유로우며, 많은 것을 표현할 수 있으므로 컴퓨터와의 상호작용을 위해 손을 이용하는 것은 적절한 방법이 된다. 본 논문에서는 PowerPoint™ 제어를 위한 8가지의 손 제스처를 정의하고, 아무런 장치 없이 명령을 내릴 수 있는 방법을 제안하였다.

3절에서 언급한 색상공간에서 피부색의 범위를 정의하여 피부색을 검출하는 방법은 빛에 민감하다고 알려져 있다. 자연광을 받는 상태에서는 만족할만한 결과를 보였지만 형광등이나 자연광이 적게 비치는 곳에서 실험한 결과 제대로 인식하지 못하였다. 이와 같은 경우는 영상 자체의 RGB 색상 값을 수동으로 가감하여 빛의 양을 임의로 조절하거나 노출되는 빛에 적응하는 방법으로 피부색을 범용 적으로 정의할 수 있는 기술을 개발하는 연구를 필요로 한다.

4절에서 언급한 DP Matching을 이용한 동작 인식에서는 정의된 동작에 대해서는 높은 인식률을 보이지만, 정의되지 않은 임의의 동작에 대해서 무효화하는 과정에서 낮은 정확도를 나타냈다. 이 문제를 해결하기 위해서 정의되지 않은 동작을 처리할 수 있는 더 효과적인 방법을 연구해야 할 것이다.

본 논문의 연구를 바탕으로 영상처리 분야에서 응용될 수 있으며, 사용자의 손 동작을 인식한 다양한 시스템이 개발될 수 있을 것이다. 손 동작을 이용한 멀티미디어 교육용 시스템이나, 영상 감시 시스템, 사용자의 행동 분석 시스템 등 여러 가지 영상 인식 기술에서 활용될 수 있을 것으로 기대한다.

참 고 문 헌

[1] 이길만, 문대성, 김성옥, 김민환, “실생활 환경에서의 손 동작 인식 및 추적 방법”, 한국멀티미디어학회 *'99추계 학술발표논문집*, 성남, pp. 577-582, 1999.11.

[2] PEER, P., KOVAC, J., AND SOLINA, F. “Human skin colour clustering for face detection”, *EUROCON 2003 - International Conference on Computer as a Tool*, 2003.

[3] Vezhnevets V., Sazonov V., Andreeva A., “A Survey on Pixel-Based Skin Color Detection Techniques”, *Proc. Graphicon - 2003*, pp. 85-92, Moscow, Russia, September 2003.

[4] Taejin Ha, Woontack Woo, “Video see-throught HMD based Hand Interface for Augmented Reality”, *HCI 2006*, vol. 1, pp. 169-174, 2006.

[5] Pedro F. Felzenszwalb and Daniel P. Huttenlocher, “Distance Transforms of Sampled Functions”, *Cornell Computing and Information Science TR2004*, 1963.

[6] Gunilla Borgefors, “Distance Transformations in Digital Images”, *Computer Vision, Graphics and Image Processing*, Vol. 34, pp. 344-371, 1986.

[7] Miyao, H. Maruyama, M, “An online handwritten music symbol recognition system”, *IJDAR*, Vol. 9, No. 1, pp. 49-58, March 2007.

[8] M. Tanaka, Y. Ishino, H.Shinmada, T. Inoue, “Dynamical Scaling in on-line hand written characters' matching”, *9th International Conference on Neural Information Processing(ICONIP'02)*, Vol 5, 2002.