論文99-36S-4-8

인터액티브 가상 환경을 위한 손 제스처 인식 시스템

(A Hand Gesture Recognition System for Interactive Virtual Environment)

曺 浯 永 *·**, 金 炯 坤 *, 高 聖 濟 **, 安 相 喆 *

(Oh-Young Cho, Hyoung-Gon Kim, Sung-Jea Ko, Sang Chul Ahn)

요 약

본 논문에서는 복잡한 영상에서 추출해낸 손 영역으로부터 제스처를 인식하여 인간과 컴퓨터의 상호작용(HCI, Human-Computer Interaction)을 위한 보다 자연스러운 인터페이스를 제공한다. 제안하는 방법은 정규화된 RGB 색상 공간에 정의한 피부색의 가우시안 분포를 이용해 조명의 변화나 개인의 차이에도 안정적으로 손 영역을 추출하며, 배경에 대한 상대적인 움직임을 이용해 복잡한 영상에서도 효과적으로 손 영역을 추출해 낸다. 추출된 손 영역은 RBF(Radial Basis Function) 신경망을 이용해 각 제스처로 인식된다. 가상 환경과의 상호작용을 제공하기 위해 두 종류의 기본적인 정적 제스처들을 정의하며 간단한 구문론적 규칙을 사용해 하나 이상의 인식 결과들을 조합함으로써 적은 수의 제스처들만으로 보다 효율적이고 다양한 상호작용이 가능하게 한다. 제안하는 시스템은 TMS320C80 DSP 칩을 사용하여 구현되었으며 320×240 영상을 12Hz로 처리함으로써 빠른 속도로 가상 환경의 인터페이스를 제공한다.

Abstract

This paper presents more natural interface for Human-Computer Interaction(HCI), that is provided by recognizing hand gestures extracted from images with complex background. By using Gaussian distribution of skin color in the normalized RGB color space the proposed method extracts a hand region regardless of luminance change and individual differences. It also applies to complex images efficiently by using relative motion to background. The extracted hand shape is recognized as a specific gesture through a RBF(Radial Basis Function) neural network. For interactions with virtual environment, two types of elementary gestures and simple syntactics rules to combine them are defined. They make it possible to provide more efficient and various interactions even with a small number of gestures. The proposed system is implemented to provide fast interface with virtual environment as it processes 320×240 images at 12Hz using TMS320C80 DSP chip.

Ⅰ. 서 론

컴퓨터를 기반으로 하는 많은 기술들이 발전함에 따라 사람들은 점차 키보드나 마우스 또는 조이스틱과 같은 장치들을 직접 다루는 데에서 벗어나 좀 더 자유롭고 편리한 HCI를 요구하게 되었다. 이를 동기로 인간의 가장 기본적인 의사 소통 수단인 음성을 인터페이스로 사용하기 위한 연구가 먼저 진행되었으며 최근에 이르러서는 시각적인 정보에 기반한 손 또는 팔의 제스처(gesture)로 관심을 돌리게 되었다[1] [2] [3]. 제스처는 의사 소통에 있어서는 음성의 보조 역할을 하거나 수화 또는 수신호와 같이 독립적인 수단으로 사용되며, 물체를 조작하거나 지시하는데 있어서는 가장 직관적인 방법으로 사용된다. 이미 2D/3D 마우스,

* 正會員, 韓國科學技術研究院 映像미디어研究센터
(Imaging Media Research Center, KIST)
** 正會員, 高麗大學校 電子工學科
(Department of Electronic Eng., Korea University)
接受日字:1998年11月13日, 수정완료일:1999年2月23日

TV 제어, 그리고 윈도우 관리기와 같은 응용을 통해 제스처는 실세계에서 컴퓨터와 인간의 상호작용(interaction)을 위한 직관적이며 효율적인 수단으로 사용될 수 있는 가능성을 보여주고 있다[4] [5] [6]. 뿐만 아니라 컴퓨터 기술의 발달에 따라 급격히 성장하고 있는 가상 환경 또는 가상 현실 분야에서 제스처는 가상의 물체를 조작하고 대화하는데 가장 적합한 수단으로 보여진다[7].

손 제스처는 보통 손의 자세(posture) 즉, 공간적인 정보만을 사용하는 정적(static) 제스처와 움직임 즉, 시간적인 정보를 사용하는 동적(dynamic) 제스처로 나눌 수 있다. 정적 제스처를 사용하는 경우 정의하는 제스처의 수가 많아질수록 구분할 수 있는 형태의 차가 적어지므로 각 제스처를 분류해내기가 어렵다. 동적 제스처는 정적 제스처에 비해 표현이 자연스럽고 사용할 수 있는 제스처의 수도 더 많지만 움직임 중에서 실제로 의미를 갖는 부분을 추출해내기가 힘들다는 단점이 있다.

손의 자세 및 움직임을 분석하는 손 제스처 인식의 방법은 일반적으로 처리하는 정보의 입력 방식에 따라 글러브 기반의 방법과 컴퓨터 비전을 기반으로 하는 방법으로 나눌 수 있다. 전자는 센서가 장착된 기계적인 글러브를 사용하여 관절의 움직임과 손의 위치를 정확히 알아낼 수 있으나, 항상 번거로운 장치를 착용해야 하므로 사용자 입장에서는 불편하고 자연스럽지 못하다. 후자는 한 대 이상의 카메라로 얻은 영상으로부터 컴퓨터 비전 기법을 사용해서 손 제스처를 인식하는 방법으로 본질적으로 편하고 자연스러운 상호작용을 할 수 있다. 그러나 이러한 방법은 복잡한 영상에서 손 영역을 추출해내고 손의 움직임을 분석하거나 손의 자세를 인식하기가 어렵다. 따라서 손에 특정한 색의 표식(marker)을 붙이거나 표식이 있는 장갑을 사용하기도 하며, 단일한 배경으로 제한하기도 한다.

본 논문에서는 손 제스처의 인식을 통해 가상 환경에서 컴퓨터와의 상호작용을 위한 인터페이스를 제공한다. 기존의 손 제스처 인식을 이용한 응용들은 제공하는 동작들마다 각각 하나의 동적 제스처 또는 정적 제스처를 할당함으로써 많은 기능을 제공하기 어렵고 확장하기도 힘들다는 단점이 있다. 제안하는 방법에서는 크게 두 종류의 기본적인 정적 제스처를 정의하고 이들을 간단한 구문론적 방법으로 결합함으로써 그러한 단점을 극복하고 효율적이며 다양한 기능의 인터페이스를 제공한다. 제안하는 방법은 컴퓨터 비전 기법을 기반으로 하면서도 배경에 대한 제약이나 표식을 사용하지 않고 손 영역을 추출해 낸다. 또한 RBF 신경망을 사용하여 추출된 손 영역의 형태로부터 손의 자세를 인식해 낸다. .

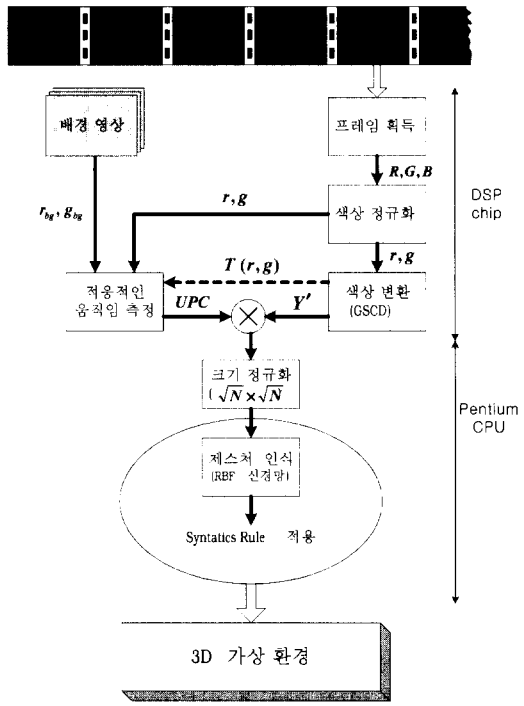


그림 1. 제안하는 손 제스처 인식 방법의 구성도
Fig. 1. Block diagram of the proposed algorithm of hand gesture recognition.

그림 1에서와 같이 실시간으로 입력되는 영상은 정규화된 RGB 색상 공간에 정의된 피부색 모델을 통해 색상 변환되며, 여기에 배경 영상과의 차이를 이용한 움직임 가중치를 적용하여 손 영역을 추출해 낸다. 추출된 손 영상은 미리 학습된 RBF 신경망을 거쳐 제스처로 인식된다. 인식된 제스처는 구문론적 규칙에 의해서 가상환경 시스템이 요구하는 입력의 형태로 변환되며 실제로 렌더링 등의 과정을 통하여 시각적으로 보이게 된다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 카메라로부터 연속적으로 입력되는 컬러 영상에서 피부색 및 움직임 정보를 이용해 손 영역을 추출하는 방법을 설명하고, Ⅲ장에서는 추출된 손 영역을 이용해 제스처

를 인식하는데 사용되는 RBF 신경망의 구조 및 학습 방법에 대해서 기술한다. Ⅳ장에서는 가상 환경에서 사용되는 제스처와 구문론적 분석 방법에 대해 설명하고 Ⅴ장에서는 제안하는 손 제스처 인식 방법에 대한 실험 및 결과를 보이고 마지막으로 Ⅵ장에서 결론을 맺는다.

Ⅱ. 손 영역 검출 방법

본 장에서는 복잡한 배경을 가진 컬러 영상에서 손 영역을 검출하는 방법에 대해 설명한다. 제안하는 방법은 카메라로부터 입력되는 컬러 영상으로부터 정규화된 RGB 색상 공간에 미리 정의되어 있는 피부색 영역을 검출하고, 배경 영상으로부터의 차영상을 구한 다음 두 결과로부터 원하는 손 영역을 검출해낸다.

1. 피부색의 가우시안 분포를 이용한 색상 변환

피부색의 분포를 이용한 색상 변환은 정규화된 RGB 색상 공간에 미리 피부색 영역을 정의해 두고, 입력 영상에서 각 화소가 해당 영역의 중심에 가까울수록 높은 값으로 변환하는 것을 기본 원리로 한다. 카메라에서 얻은 영상의 각 화소는 RGB 색상 공간의 점들로 표현되는데 RGB 색상 공간에서 피부색은 개인에 따라 차이가 있을 뿐만 아니라 조명의 변화에 대해서도 다른 값을 가지므로 일반화된 영역을 정의하기가 어렵다. 보통 단일 조명 즉 휘도(luminance)에 의한 R, G, B 값의 변화는 식 (1)과 같이 각 성분에 동일한 계수의 곱으로 나타난다. 따라서 기존의 RGB 색상 공간을 식 (2)와 같이 정규화 함으로써 휘도에 의한 색의 변화를 제거할 수 있다.

$$(R', G', B') = (\alpha R, \alpha G, \alpha B) \tag{1}$$

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B},$$

$$b = \frac{B}{R+G+B} \tag{2}$$

$$r + g + b = 1 \tag{3}$$

이 때, 정규화된 색상 성분 r, g, b 사이에는 식 (3)의 관계가 성립하므로 정규화된 색상 공간은 r, g 성분만으로 표현할 수 있다. 따라서 본 논문에서는 정규화된 RGB 색상 공간에 피부색을 정의함으로써 휘도에 의한 화소 값의 변화를 제거할 수 있을 뿐만 아니라 r, g 두 성분만을 사용하므로 처리하는 색상 정보의 양도 줄일 수 있다. HSI 색상 모델에서도 이와 유사한 개념으로 H(hue)와, S(saturation) 성분만을 사용해서 색 정보를 처리할 수 있는데, H와 S값을 얻기 위해서는 R, G, B 성분에 대한 복잡한 변환식을 거쳐야 하므로 정규화된 RGB 색상보다 더 많은 계산 시간이 소요되는 단점이 있다.

정규화된 r, g 색상 평면상에서 피부색은 표현되는 색상 정보의 양이 감소하였음에도 불구하고 개인, 몸의 부위 또는 환경에 따라서 약간씩 다른 분포를 보인다. 따라서 이러한 변화량을 충분히 고려하기 위해 실험에서 얻은 피부색의 평균 및 분산 값을 이용하여 정규화된 색상 공간 위에 2차원의 Gaussian 분포로 피부색을 모델링하며 이를 일반화된 피부색 분포(Generalized Skin Color Distribution, GSCD)로 정의한다.

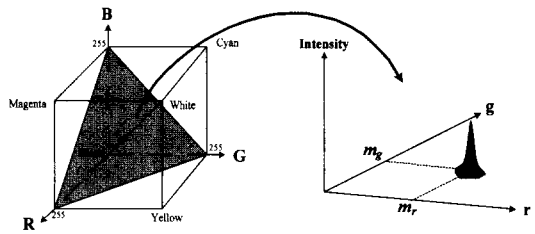


그림 2. 정규화된 색상 공간에서의 GSCD 모델링
Fig. 2. GSCD modeling in the normalized color space.

그림 2는 정규화된 색상 공간에 정의된 GSCD를 보여 주고 있다. 정규화된 색상 공간의 화소에 대해 GSCD를 적용하면 식 (4)와 같이 피부색의 평균에 가까울수록 높은 값을 갖고 멀수록 낮은 값을 갖게 된다. 정규화된 색상 공간의 영상을 GSCD를 통해 피부색에 가까울수록 높은 밝기값(intensity)을 갖는 영상으로 바꾸는 과정을 색상 변환(Color transform)이라 부르기로 한다.

$$\begin{aligned} Y(x,y) &= G(r(x,y), g(x,y)) \\ &= \frac{1}{2\pi\sigma_r\sigma_g} \exp\left[-\frac{1}{2}\left\{\left(\frac{r(x,y)-m_r}{\sigma_r}\right)^2 \right.\right. \\ &\quad \left.\left. + \left(\frac{g(x,y)-m_g}{\sigma_g}\right)^2\right\}\right] \end{aligned} \tag{4}$$

여기에서 $Y(\cdot)$는 색상 변환된 값이고, $G(\cdot)$는 차원의 Gaussian 함수이며, (x, y)는 각 화소의 위치

를 나타낸다. 이러한 색상 변환은 평균 및 분산 값을 변화시킴으로써 관심을 갖는 어떠한 색상에 대해서도 적용이 가능하다.

그런데, 정규화된 색상 공간에서의 색상 변환은 휘도가 낮은 경우 많은 오류를 일으키게 된다[10]. 예를 들어 각 성분의 합이 50 인 화소(R', G', B')의 정규화된 성분을(r', g') $r', g' \in \{0, 1, 2, \ldots, 255\}$라고 할 때, 같은 휘도를 갖는 화소 $(R'-1, G'+1, B')$는 $(r'-5, g'+5)$로 정규화 된다. 다시 말하면, 휘도가 낮은 화소는 RGB 색상 공간의 정규화 과정에서 잡음에 의한 작은 변화에도 쉽게 다른 색상으로 바뀌게 된다. 이로 인해 원래는 피부색인 화소가 색상 변환 후에는 낮은 밝기값을 가질 수 있으며, 반대로 피부색이 아닌 값이 색상 변환을 통해 높은 밝기값을 가질 수 있다. 따라서 R, G, B 성분의 합이 적절한 임계값보다 작은 경우에는 정규화 및 색상 변환을 적용하지 않는다. 보통 임계값은 색상 변환에서 사용되는 분산 값이 작을수록 더 큰 값으로 결정한다.

2. 움직임 가중치를 이용한 손 영역의 검출

GSCD를 이용한 색상 변환을 거친 입력 영상은 피부색에 가까울수록 높은 밝기값을 갖는 흑백 영상으로 변환된다. 따라서 손 영역뿐만 아니라 배경에서 피부색과 유사한 색상을 갖는 부분도 마찬가지로 높은 값을 갖게 된다. 일반적으로 고정된 카메라로부터 얻은 영상의 배경은 거의 일정하며 손 영역은 배경에 대해 상대적인 움직임을 가지고 있으므로 이를 이용해 피부색과 유사한 색상을 갖는 배경 부분을 제거하고 손 영역만을 얻을 수 있다. 제안하는 방법은 미리 고정된 물체만이 있는 영상, 즉 배경 영상을 구해놓은 다음 현재 입력되는 영상과의 차이를 이용해 움직임을 측정한다. 현재 입력된 영상의 손 영역은 배경 영상에서의 동일한 위치에 있는 영역과 다르므로 화소값의 차이가 비교적 크다. 따라서 적당한 임계값(threshold value)을 이용해 배경에 대해서 상대적인 움직임을 갖는 손 영역을 얻을 수 있다. 이때 배경 영상은 초기 수 십 프레임 동안 얻은 영상들의 평균으로 얻을 수 있다. 그런데 일반적으로 카메라로부터 얻어지는 영상은 조명의 변화나 반사, 그리고 카메라의 잡음에 민감하므로 고정된 카메라로부터 얻은 영상이라 할지라도 영상을 획득하는 시간에 따라 배경의 화소에 변화가 생긴다. 따라서 일반적으로 두 영상간의 차이를 구하는데 사용되는 각 화소들의 R, G, B 성분 또는 밝기값의 절대차(absolute difference)는 손 영역이 아닌 배경의 화소들에 대해서도 큰 값을 가질 수 있다. 본 논문에서는 식 (5)와 같이 조명의 변화에 안정적인 정규화된 r, g 값을 이용해 두 영상간의 차이를 구한다.

$$D(x, y, t) = |r(x, y, t) - r(x, y, 0)| + |g(x, y, t) - g(x, y, 0)| \quad (5)$$

여기에서 $(x, y, 0)$는 배경 영상의 각 화소를 나타내며, (x, y, t)는 현재 프레임의 영상의 각 화소를 나타낸다. 또한 제안하는 방법은 해당 화소의 차를 구할 때, 주위 화소들의 변화도 함께 고려함으로써 잡음 등에 의한 오차를 최소화한다. 이때 주위의 화소들을 고려한 해당 화소의 움직임은 식 (5)의 절대차를 그대로 누적하는 대신 식 (6)과 같이 적절한 임계값을 넘는 화소들의 개수를 더하는 방법으로 구한다. 이로 인해 해당 화소의 움직임이 고려되는 화소들 중에서 변화가 가장 큰 화소의 절대차에 종속되는 것을 방지한다.

$$UPC(x, y, t) = \sum_{i=x-N/2}^{x+N/2} \sum_{j=y-N/2}^{y+N/2} a(i, j, t) \quad (6)$$

$$a(i, j, t) = \begin{cases} 1, & \text{if } D(i, j, t) > T(i, j, t) \\ 0, & otherwise \end{cases} \quad (7)$$

$$T(i, j, t) = \frac{255}{1 + \exp\left\{\frac{Y'(i, j, t) - 255/2}{Q}\right\}} \quad (8)$$

여기서 주위 화소의 변화를 고려하기 위해 사용하는 창의 크기는 (N+1)×(N+1)이며 $Y'(i, j, t)$는 식 (4)의 색상 변환 결과를 [0, 255] 의 범위로 비례 확대(scaling)한 것이다. $T(i, j, t)$는 시그모이드(sigmoid) 함수로서 현재 프레임의 영상에서 위치 (i, j)에 있는 화소에 대해 색상 변환한 결과에 따라 절대차에 대한 적응적인 임계값을 제공한다. 이 때, Q 값은 시그모이드 함수의 기울기를 결정한다.

그림 3에서와 같이 시그모이드 함수는 기본적으로 색상 변환된 값이 큰 화소에 대해서는 작은 임계값을 제공한다. 따라서 배경에 대한 손 영역의 움직임이 작더라도 이를 검출해 낼 수 있다. 보통 배경에 피부색에 가까운 색상이 많으면 Q 값을 작게 설정하고, 그렇지 않은 경우에는 Q 값을 크게 설정한다.

색상 변환한 결과에 대해 식 (6)에서 얻은 적응적인

움직임을 가중치를 곱함으로써 피부색을 가지면서 배경에 대한 움직임을 갖는 손 영역을 구할 수 있다.

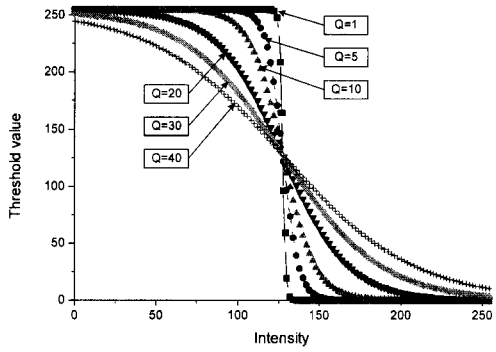


그림 3. 적응적인 임계값을 위한 시그모이드 함수
Fig. 3. Sigmoid function for adaptive thresholding.

$$Z(x, y, t) = Y(x, y, t) \otimes UPC(x, y, t) \quad (9)$$

그림 4는 제안하는 방법을 통해 입력 영상에서 손 영역을 추출하는 과정의 예를 보여주고 있다. (a)와 (b)는 각각 배경 영상 및 입력 영상이고 (c)는 색상 변환한 결과 영상이다. (d)는 색상 변환의 결과에 적응적인 움직임을 곱하여 얻은 손 영역을 보여주고 있다.

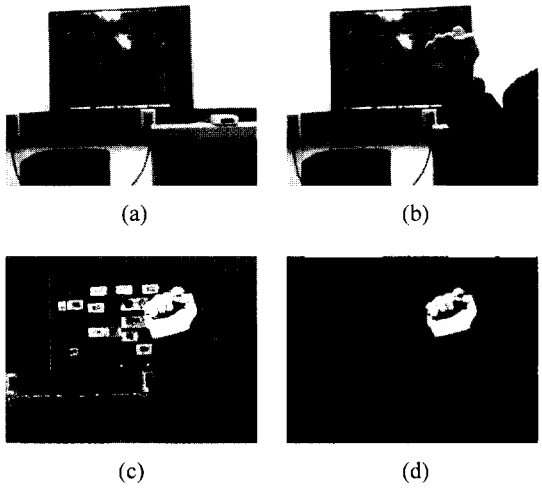


그림 4. 손 영역 추출 과정 (a) 배경 영상 (b) 입력 영상 (c) 색상 변환된 영상 (d) 움직임을 고려하여 추출한 손 영역
Fig. 4. The hand region extraction procedure (a) Background image (b) Input image (c) Result of color transform (d) Hand region extracted using relative motion.

Ⅲ. RBF 신경망을 이용한 제스처 인식

특정한 패턴들에 대해 선별적으로 반응하는 뉴런들로 구성되는 하나의 중간층을 갖는 RBF(Radial Basis Function) 신경망은 학습을 통하여 임의의 함수를 근사(approximation)하거나 패턴을 분류할 수 있다[11][12][13][14]. RBF 신경망은 교사 학습법(supervised learning)이 적용되는 가중치 계층이 중간층과 출력층 사이에 하나만 존재하고 이로 인해 학습 자체가 선형적이므로 다층 퍼셉트론(multi-layer perceptron)보다 학습 속도가 빠르며 잘 수렴한다. 보통 비교사 학습(unsupervised learning)을 통해서 결정되는 중간층의 각 뉴런은 특정한 입력 패턴들에 대해서만 높은 반응을 나타내며, 입력 패턴들이 존재하는 모든 공간에 대해서 반응하도록 서로 중첩되게 설계된다. 기존의 연구 결과들로부터 RBF 신경망의 근사 능력은 영상과 같이 높은 차원의 불연속 데이터를 다루는데 적합하므로 본 논문에서는 추출된 손 영상으로부터 제스처를 인식해내는 방법으로 RBF 신경망을 사용한다[15][16][17]. 본 논문에서는 두 종류의 정적 제스처를 사용하는데 이들은 구문론적 방법에 의해 인식되는 시점이 다르므로 각각 다른 구조의 RBF 신경망을 적용한다. 입력 벡터의 차원으로 결정되는 입력층 뉴런의 개수와 중간층 및 가중치를 결정하는 학습 방법은 두 신경망 모두 같으며, 구분해낼 제스처의 클래스에 해당하는 출력층의 개수와 학습을 통해서 결정되는 중간층 뉴런의 개수 및 변수들은 달라지게 된다. 이와 같이 모든 제스처를 하나의 RBF 신경망으로 인식하지 않고 두 개로 나눔으로써 각각의 구조는 매우 간단해진다. 따라서 학습할 때 빠른 속도로 수렴할 수 있으며 인식할 때에도 처리 속도가 빨라 실시간 인식 방법으로 적합하다. 또한 다른 제스처들을 추가하는데도 충분한 여유를 가지게 된다.

그런데 Ⅱ장에서 설명한 방법으로 얻은 사각형의 손 영상은 다양한 크기를 갖게 되므로 RBF 신경망을 이용해 인식하기 위해서는 입력층의 N개의 뉴런에 모든 화소가 대응되도록 손 영상의 크기를 정규화 하여야 한다. 본 논문에서는 추출된 손 영상을 $\sqrt{N}\times\sqrt{N}$의 정사각형 영상으로 정규화 한다. 이를 위해 먼저 손 영역 내부에서 적절한 임계값 이상의 밝기값을 갖는 화소들의 중심점(centroid)에 해당하는 좌표를 구한다. 그런 다음 이 좌표에서 가장 먼 화소까지의 거

리를 $\sqrt{N}/2$로 두고 나머지 화소들의 좌표를 비례 축소하여 추출된 영상의 중심점을 새로운 중심으로 하는 $\sqrt{N}\times\sqrt{N}$ 영상으로 변환한다.

1. RBF 신경망의 구조

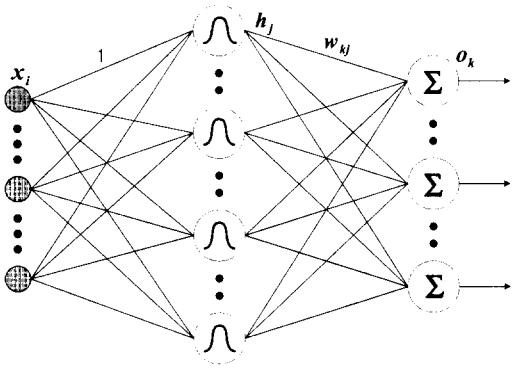


그림 5. RBF 신경망의 구조
Fig. 5. The structure of a RBF neural network.

RBF 신경망은 그림 5와 같이 하나의 입력층과 하나의 중간층(또는 은닉층, hidden layer), 그리고 하나의 출력층으로 구성되는 전방향 구조의 혼합 학습 신경망이다. 입력층은 N개의 뉴런으로 구성되며 모든 뉴런은 입력 벡터 $\boldsymbol{x}$의 각 성분과 일대일로 대응된다. 중간층은 활성 함수(activation function)를 갖는 M개의 뉴런으로 구성되며 보통 활성 함수로는 가우시안 함수가 사용된다. 중간층의 출력값은 다음과 같다.

$$h_j = \exp\left\{-\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{\sigma_j^2}\right\} = \exp\left\{-\frac{\sum_{i=1}^{N}(x_i-c_{ji})^2}{\sigma_j^2}\right\} \tag{10}$$

여기서 $\boldsymbol{c}_j$와 σ_j는 뉴런의 특성을 결정하는 변수로서 각각 '중심(center)'과 '너비(width)'라고 한다. 입력 패턴이 중심에 가까울수록 뉴런의 출력값은 크며, 너비는 각 뉴런의 반응 범위를 결정한다. 너비가 작으면 뉴런은 중심과 유사한 입력 패턴에 대해서만 반응하지만, 너비가 크면 좀 더 넓은 범위의 입력 패턴에 대해서도 반응하게 된다.

출력층에서 각 뉴런의 출력값은 다음과 같이 중간층 출력의 선형 가중합으로 표현된다.

$$o_k = \sum_{j=1}^{m} w_{kj}h_j \tag{11}$$

여기에서 m은 중간층 뉴런의 개수이고 h_j는 중간층의 j번째 뉴런의 출력값을 나타낸다. w_{kj}는 중간층의 j번째 뉴런과 출력층의 k번째 뉴런 사이의 가중치로 교사 학습법을 통해서 구할 수 있다. 중간층 및 가중치의 자세한 학습 방법에 대해서는 다음 절에서 설명하기로 한다.

2. RBF 신경망의 학습

본 절에서는 손 제스처 인식을 위해 사용하는 RBF 신경망을 학습시키는 방법에 대해서 설명한다. RBF 신경망의 학습은 중간층의 중심과 너비를 결정하고, 중간층과 출력층 사이의 가중치를 구하는 세 단계로 나눌 수 있다. 본 논문에서는 클러스터링 방법인 K-평균(K-means)을 효율적으로 사용해 중심을 구하고 이들로부터 너비를 결정한다. 그리고 LMS(least mean square) 방법을 사용해 중간층과 출력층 사이의 가중치를 학습시킨다.

중심을 구하는 방법으로는 학습 패턴에서 임의로 n개의 표본을 추출해서 사용하는 방법, K-평균 또는 SOFM(self organizing feature map)과 같은 클러스터링 방법 및 기타 여러 방법이 있다[18][19]. 가장 단순하고 일반적인 방법은 모든 학습 패턴들을 각 뉴런의 중심으로 할당하는 것이나, 대부분의 응용이 많은 학습 패턴을 사용하므로 실용적이지 못하다. K-평균을 사용하는 방법은 학습 패턴을 K 개의 그룹으로 나누고 각 그룹의 평균 패턴을 구하여 중간층의 각 뉴런에 할당한다. 그런데 학습 패턴들이 각 클래스마다 입력 공간에 균일하게 분포되어 있지 않는 경우 K-평균을 적용하게 되면 학습 패턴이 조밀하면서 서로 근접하여 분포하는 클래스들에는 적은 수의 뉴런들이 할당되고 이들과 멀리 떨어져 있으면서 넓게 분포되어 있는 클래스에는 상대적으로 많은 수의 뉴런들이 할당된다. 이에 따라 클러스터링이 변화가 많은 학습 패턴의 특정 클래스에 치우쳐 다른 클래스의 입력 패턴의 공간을 충분히 표현하지 못하게 되는 오류가 발생하게 된다. RBF 신경망에서는 학습 패턴이 어느 클래스에 속하는지를 미리 알고 있다. 따라서 본 논문에서는 각 클래스별로 따로 K-평균 방법을 적용하여 모든 클래스에 대해 중심이 적절한 수로 분배되도록 한다.

중간층의 너비는 각 뉴런의 반응 범위가 어느 정도 중첩되도록 실험적으로 찾아내는 것이 보통이다[12][18]. 뉴런들의 반응이 중첩되도록 너비를 결정하는 이유는 실제로 입력되는 패턴들은 학습 패턴의 분포보

다 훨씬 더 다양하므로 이들이 분포하게될 공간에 대해서도 중간층의 뉴런들이 적절한 반응을 나타낼 수 있게 하기 위함이다. 달리 말하면 어떠한 중간층의 뉴런에 대해서도 출력값을 갖지 못하는 입력 패턴은 RBF 신경망을 통해서 구분될 수 없다. 식 (12)와 같이 각 뉴런의 중심과 다른 모든 뉴런의 중심과의 평균 유클리디안 거리를 사용하면 좀 더 효율적으로 너비를 구할 수 있다[20].

$$\sigma_\alpha = \frac{1}{\sqrt{2}} \langle \| c_\alpha - c_\beta \| \rangle \qquad (12)$$

여기에서, $\|\cdot\|$은 유클리디안 거리를 나타내며, $\langle \cdot \rangle$은 모든 유클리디안 거리 쌍에 대한 평균을 나타낸다. 위의 식으로부터 나머지에 비해 중심이 멀리 떨어져 있는 뉴런의 너비는 큰 값을 갖게되고, 근접해 있는 뉴런들은 작은 값의 너비를 갖게 된다.

중간층과 출력층 사이의 가중치는 식 (13)으로 표현되는 신경망의 오차를 최소화하도록 학습된다.

$$E_p = \frac{1}{2} \sum_k (d_{pk} - o_{pk})^2 \qquad (13)$$

여기에서 d_{pk}는 출력층의 k번째 뉴런에서 p번째 입력 패턴의 요구되는 출력값(desired output)을 나타낸다. 식 (11)과 식 (13)에서 보듯이 E_p는 결국 중간층과 출력층 사이의 가중치의 함수이므로 반복적인 학습을 통해 E_p의 값이 최소가 되도록 가중치를 조정할 수 있다. LMS 법칙으로부터 n번째 반복에서 가중치의 조정값은 아래와 같이 구할 수 있다.

$$\Delta w_{kj}(n) = \eta(d_{pk} - o_{pk}) h_{pj} + \alpha \Delta w_{kj}(n-1) \qquad (14)$$

여기에서 η와 α는 변화, 즉 수렴의 속도를 결정하는 변수로서 각각 학습률(learning rate)과 관성항(momentum)이라고 한다.

Ⅳ. 제스처와 구문론적 규칙

본 논문에서 사용하기 위해 정의하는 제스처는 가상 환경의 입력 장치로서의 손 제스처 인식이 얼마나 편리하고 효율적인 기능을 갖는지를 직접적으로 평가하는 기준이 된다. 가상 환경에서의 상호작용은 크게 내비게이션과 물체의 직접 조작으로 나눌 수 있다. 내비게이션은 사용자가 가상 환경 내부를 이동하는 것을 말하며, 물체의 직접 조작은 가상의 세계에 존재하는 물체를 이동시키거나 형태를 바꾸는 것 등을 말한다. 어떠한 장치가 가상 환경의 입력 수단으로 적합하다고 판단하는 기준으로는 얼마나 많은 상호작용을 제공하느냐와 하나의 동작이 인식을 거쳐 가상 환경에서 실행되는 과정이 간단하고 빠르게 진행되느냐로 볼 수 있다. 이러한 관점에서 가장 좋은 손 제스처 인식 시스템은 가능한 많은 제스처를 정의해 놓고 각 제스처마다 각각 하나의 동작을 할당하며, 동적 제스처보다는 표현 시간이 짧은 정적 제스처를 사용하는 것이다. 그런데 가상 환경에서 가능한 모든 동작들에 대해 각각 하나의 제스처를 할당하는 것은 현실적으로 불가능하다. 즉, '어떠한 방향으로 진행한다'라든지 '어느 위치에 있는 물체를 삭제한다'라고 할 때, '어떠한 방향' 또는 '어느 위치'에 해당하는 상황들은 하나의 제스처로 할당하기에는 그 수가 너무도 많다.

본 논문에서는 두 종류의 기본적인 정적 제스처들과 구문론적 방법을 통해서 가상 환경에서의 내비게이션 및 물체의 직접 조작을 효율적으로 제공한다. 정적 제스처는 명령 제스처와 지시 제스처 나눈다. '진행한다' 또는 '삭제한다'와 같이 넓은 의미의 동작을 나타내는 제스처들을 '명령 제스처'라 하며, 명령 제스처로 7개의 동작 즉, [진행], [감속/후진], [정지], [시선 변화], [물체 선택], [물체 삭제], 그리고 [물체 이동]을 정의하였다.

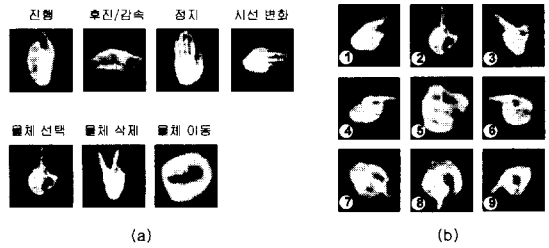


그림 6. (a) 명령 제스처 (b) 지시 제스처
Fig. 6. (a) Command gestures (b) Pointing gestures.

그림 6의 (a)는 이러한 명령 제스처들을 표현하는 손의 자세를 보여주고 있다. 또한 '어떤 방향'과 '어느 위치'에 해당하는 제스처들을 '지시 제스처'라 하며, 그림 6의 (b)와 같은 9개의 제스 처들로 정의하였다. 가상 환경에서 사용되는 동작들은 명령 제스처로 구분되며 지시 제스처는 위치와 방향을 나타내는 보조 역할을 한다. [진행]과 같은 내비게이션에 해당하는 명령 제스처와 결합하는 경우 지시 제스처는 특정한 방향을

표현하며, [물체 선택]과 같은 물체의 직접 조작에 해당하는 명령 제스처와 결합하는 경우 지시 제스처는 특정한 위치를 가리키게 된다. 제안하는 방법은 간단한 구문론적 규칙을 통해 이러한 명령 제스처와 지시 제스처를 순차적으로 조합함으로써 다양한 상호작용을 가능하게 한다.

구문론적 규칙에 의해 가상 환경에서의 모든 동작들은 다음의 두 가지 형태로 구성한다. 첫 번째 형태는 동작이 명령 제스처만으로 이루어진 경우이다. 여기에 해당하는 명령 제스처로는 [정지], [감속/후진] 그리고 [물체 삭제]가 있다. [물체 삭제]의 경우에는 삭제의 대상이 되는 물체의 위치를 알아야 하는데, 이에 대한 설명은 나중에 하기로 한다. 두 번째는 형태는 동작이 명령 제스처와 지시 제스처가 결합되어 이루어진 경우이다. 이 경우에는 명령 제스처가 인식된 후 곧바로 지시 제스처가 인식되면서 동작이 실행된다. 여기에 해당하는 명령 제스처로는 [진행], [시선 변화], [물체 선택], 그리고 [물체 이동] 등이 있다. 단, 명령 제스처 중 [진행]은 앞으로 설명될 상태의 종류의 따라 두 형태를 모두 가질 수 있다.

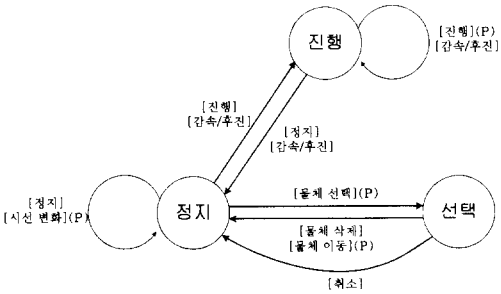


그림 7. 명령 제스처의 상태 천이도
Fig. 7. State transition diagram of command gestures.

그리고 구문론적 규칙은 정지, 진행, 그리고 선택의 세 가지 상태를 통해 모든 동작들이 체계적으로 실행되도록 한다. 그림 7은 동작들의 실행에 따른 가상 환경의 상태의 변화를 보여주고 있다. 정지 상태에서 가상 환경은 움직이지 않고 멈춰있으며[시선 변화], [물체 선택], [진행], 그리고 [감속/후진] 등의 동작이 가능하다. [시선 변화]는 멈춰있는 상태에서 지시 제스처로 인식된 방향으로 카메라의 뷰(view)만을 이동하는 것이다. 정지 상태에서 [진행]이 인식되면 가상 환경은 초기 속도로 앞으로 진행하게 되며 상태도 변한다. 여기서 다시 [진행]이 인식되는 경우에는 지시 제스처의 인식을 통해 상, 하, 좌, 우로 단위량만큼 회전하거나 속도가 증가한다. [감속/후진]도 진행 상태에서 가능한 동작으로서 지시 제스처 없이 속도를 감소시킨다. 진행 상태에서는 [정지] 또는 [감속/후진]으로 속도가 0이 되었을 때 정지 상태로 넘어갈 수 있으며 선택 상태로는 바로 이동할 수 없다. 정지 상태에서 [감속/후진]이 인식되면 가상 환경은 일정한 속도로 후진한다. 정지 상태에서 [물체 선택]이 인식된 후 지시 제스처로 특정 위치의 물체를 지정하면 선택 상태로 이동한다. 선택 상태에서는 [물체 삭제] 및 [물체 이동]이 가능한데, [물체 삭제]는 지시 제스처의 인식만으로 [물체 선택]에서 지정된 물체가 삭제되지만 [물체 이동]은 물체가 이동할 위치를 추가로 지정해야 하므로 지시 제스처의 인식이 필요하다. 이 두 동작이 실행되면 자동으로 정지 상태로 되돌아 간다. 또한 선택 상태에서 현재 프레임에 손 영역이 아예 존재하지 않는 '취소'가 발생하는 경우에는 선택된 물체에 대한 아무런 조작 없이 정지 상태로 복귀한다. 표 1은 가능한 모든 동작들에 대한 구문론적 규칙을 자세히 설명하고 있다.

표 1. 각 동작에 대한 구문론적 규칙
Table 1. Description of syntactics rules of all interactions.

동작	현재 상태	다음 상태	지시 제스처	세부 구문
진행	정지	진행	×	초기 속도로 전진 : v = a
	진행	진행	⑤	속도 증가 : v = v + a
			All (⑤는 제외)	지시한 방향으로 b° 회전
감속/후진	정지	진행	×	초기 속도로 후진 : v = -a
	진행	진행, 정지	×	속도 감소 : v = v - a
정지	진행	정지	×	정지 : v = 0
시선 변화	정지	정지	All	지시한 방향으로 뷰(view)만 b° 회전
물체 선택	정지	선택	All	지시한 위치의 물체를 선택
물체 삭제	선택	정지	×	선택된 물체를 삭제
물체 이동	선택	정지	All	선택된 물체를 지시한 위치로 이동

지시 제스처는 [진행], [시선 변화], [물체 선택], [물체 이동] 등의 명령 제스처와 같이 사용되는데 그림 7의 "(P)"는 지시 제스처가 결합된다는 의미이다. 지시 제스처가 [진행]이나 [시선 변화]와 결합될 때는 그림 6의 (b)와 같이 직관적으로 9개의 방향성을 제공한다. 따라서 사용자는 이를 이용해 8방향으로 회전하거나 속도 조절이 가능하므로 가상환경을 충분히 내비게이션할 수 있다. 물체의 직접 조작에 있어서 9개의

지시 제스처는 전방의 공간이 투영된 평면을 9개로 나누었을 때, 그 각각에 일대일 대응하여 해당 영역의 위치를 표현한다. 그런데 9개의 영역으로는 물체의 자세한 위치를 나타낼 수 없으므로 실질적으로 사용하기 힘들다. 자세한 위치의 지시가 요구되는 [물체 선택]이나 [물체 이동] 등과 같은 동작에서는 다단계 지시 방법(multi-level pointing)을 사용하여 이러한 문제를 해결할 수 있다.

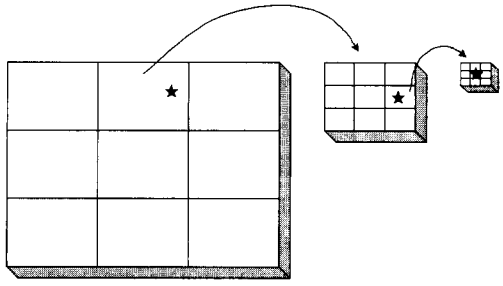


그림 8. 다단계 지시 기법
Fig. 8. Multi-level pointing.

그림 8과 같이 다단계 지시 방법은 우선 전체의 9개 영역 중 하나를 선택한 다음 다시 지시 제스처를 이용해 그 영역에 대해서 작은 9개의 영역을 나눈다. 이러한 과정을 반복함으로써 자세한 위치까지도 지정할 수 있다.

V. 실험 방법 및 결과

1. 실험 환경 및 실험 방법

본 논문에서 제안한 제스처 인식 시스템은 펜티엄 PC 상에서 C/C++ 언어를 이용하여 구현하였다. 영상의 입력 장치로는 PULNiX사의 TMC-74 CCD 컬러 카메라를 사용하였다. Matrox사의 Genesis 보드를 이용해 640x480 해상도의 영상을 초당 30 프레임의 속도로 얻었으며 이를 320x240의 해상도로 샘플링해서 사용하였다. 가상 환경에서의 자연스러운 인터페이스를 제공하기 위해서 가장 필수적인 것은 빠른 속도로 손 제스처를 인식해 내는 것이다. 따라서 본 논문에서는 Genesis 보드에서 지원하는 Texas Instrument사의 TMS320C80 DSP 칩과 펜티엄 CPU에 작업을 적절히 분배하고 병렬처리 함으로써 제안하는 방법을 빠른 속도로 구현하였다.

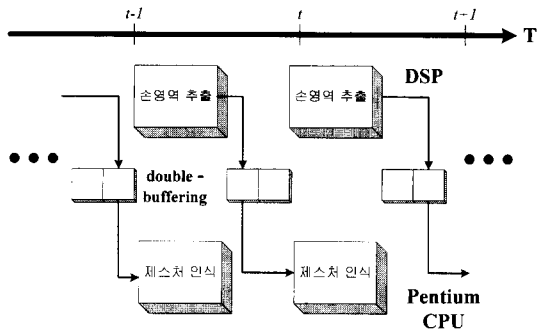


그림 9. 손 제스처 인식 방법의 병렬 처리
Fig. 9. Parallel processing of the hand gesture recognition algorithm.

그림 9에서와 같이 카메라로부터 영상을 받아서 색상 및 움직임을 이용하여 손 영역을 분리해 내는 작업은 DSP 칩에서 수행되며 펜티엄 CPU에서는 추출된 손 영역을 크기에 대해 정규화한 다음 RBF 신경망을 통해 제스처를 인식하는 작업을 수행하였다. 가상 환경의 인터페이스로서 제안한 제스처 인식 시스템의 성능을 평가하기 위해서 간단한 3차원의 가상 환경을 구성하였다. 가상 환경은 Silicon Graphics사의 Octane에서 Open Inventor를 이용해 제작하였다. 제작된 가상 환경은 3차원의 실험실 내부를 보여주고 있으며 가로, 세로, 그리고 깊이가 각각 15000, 10000, 20000 화소로 내비게이션이 충분히 가능하였다.

2. 손 영역 추출 결과

가상 환경의 인터페이스로서 제안한 손 제스처 인식 시스템이 잘 동작하기 위해서는 무엇보다도 입력되는 영상에서 정확하게 손 영역을 추출해야 한다. 실험에서 색상변환에 사용하는 GSCD의 평균값과 분산값은 r, g 각각에 대해 101과 79 그리고 12^2와 10^2을 사용하였다. 위의 값들은 카메라의 모델에 따라 약간씩 달라질 수 있다. 보통은 분산값만을 적절히 조절하여 사용할 수 있으나, 카메라의 특성이 아주 다른 경우에는 그에 맞는 피부색의 평균값을 다시 구해야 한다. 배경 영상으로는 초기 100 프레임에 대한 평균 영상을 사용하였다. 움직임에 대한 적응적인 임계값을 제공하는 시그모이드 함수의 Q값은 35로 설정하였다. 대표적인 손 영역 추출의 결과는 그림 10에 나타나 있다.

그림 10의 (a)와 (b), 그리고 (c)는 올바른 손 영역 추출의 결과이다. (a)에는 배경의 좌측 중간쯤에 피부색의 인형 얼굴이 있는데도 손 영역을 제대로 추출하

였으며 (b)와 같이 우측 상단에 피부색을 갖는 몸의 다른 일부분이 어느 정도 포함된 경우에도 손 영역을 추출할 수 있다. (c)에서는 현재 입력되는 영상이 배경 영상과 다른 조명 상태를 가질 때의 손 영역 추출 결과로서 조명 변화에도 안정적으로 손 영역을 추출할 수 있음을 보여준다.

그림 10의 (d)와 (e)는 손 영역이 제대로 추출되지 못한 경우를 보여준다. (d)는 손뿐만 아니라 얼굴 전체가 움직임이 있는 피부색 영역으로 검출된 경우이다. 제안한 방법에서는 일반적으로 움직임이 있는 피부색 영역은 손뿐이라고 가정하고 있으며, 그렇지 않은 경우 가장 큰 영역을 손으로 추출한다.

따라서 이 경우에는 b)와는 달리 얼굴이 훨씬 더 큰 영역이므로 손이 제대로 추출되지 못하였다. (e)는 피부색과 유사한 색의 무늬를 갖는 옷이 손 영역과 겹쳐서 하나의 덩어리를 형성함으로 인해 손 영역이 잘못 추출된 경우이다.

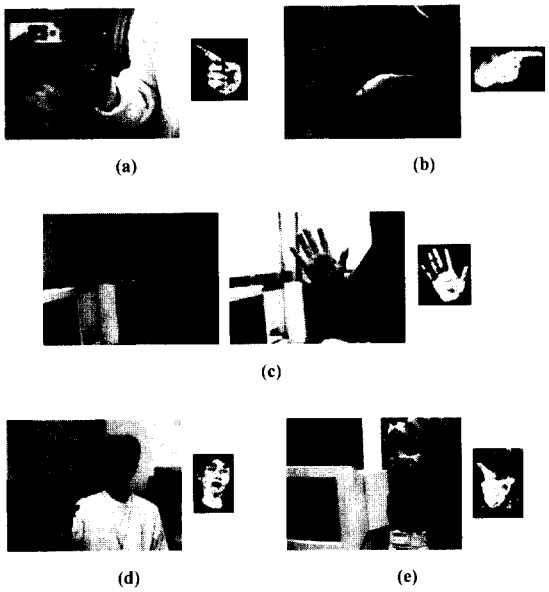


그림 10. 손 영역 추출 결과 (a) 배경에 살색이 있는 경우 (b) 얼굴의 일부가 포함된 경우 (c) 배경과 조명이 다른 경우 (d) 손의 크기가 작고 얼굴 전체가 포함된 경우 (e) 옷이 피부색과 유사한 경우

Fig. 10. Results of extracting hand region (a) With skin color in background (b) Including a part of face (c) Different illumination condition with background (d) With a small hand and a whole face (e) With clothes of skin-like color.

3. 제스처 인식 결과

추출된 손 영역으로부터 제스처를 인식하기 위해 명령 제스처들과 지시 제스처들에 대해 각각 하나의 RBF 신경망을 사용하였다. 6명에 대해서 한 사람마다 70장의 명령 제스처 영상과 90장의 지시 제스처 영상을 얻어 결국 명령 제스처를 인식하는 RBF 신경망에는 총 420장의 학습 패턴들을 사용하였으며 명령 제스처에 대해서는 총 540장을 사용하였다. 각 학습 패턴은 49x49의 크기로 정규화하여 사용하였으므로 RBF 신경망은 2401개의 입력 뉴런을 갖게 되었다. K-평균 방법으로 중간층 뉴런은 모든 제스처들에 대해서 4개를 할당했는데 즉, 명령 제스처의 RBF 신경망에는 28개, 지시 제스처의 RBF에는 36개의 중간층 뉴런을 사용하였다.

표 2. RBF 신경망의 손 제스처 인식률
Table 2. Recognition rate of hand gestures by using RBF neural networks.

실험자	인식률(%)		
	명령 제스처	지시 제스처	전체
가	67/70(95.7%)	87/90(96.7%)	154/160(96.3%)
나	67/70(95.7%)	86/90(95.6%)	153/160(95.6%)
다	68/70(97.1%)	84/90(93.3%)	152/160(95.0%)
전체	202/210(96.2%)	257/270(95.2%)	459/480(95.6%)

표 2는 RBF 신경망의 인식 능력을 측정하기 위한 실험의 결과를 보여주고 있다. 이 실험에서는 학습에 사용되지 않은 사람들의 손 영상을 대상으로 하였다. 같은 제스처라 하더라도 사람마다 표현하는 방법이 약간씩 차이가 있으므로 개인마다 인식률이 조금씩 다르다. 표의 결과로부터 명령 제스처보다는 지시 제스처가 조금 더 표현하기 어렵다는 것을 알 수 있다.

4. 가상 환경에서의 실험 결과

위의 각 부분별 실험을 바탕으로 제안한 손 제스처 인식 시스템을 실제로 3차원의 가상 환경과 연결해 보았다. 제작된 가상 환경은 내비게이션이 가능하도록 제작되었으며 속도의 조절 및 좌, 우, 위, 아래로의 진행 방향의 변경이 가능하도록 하였다. 손 제스처 인식 시스템으로부터 가상 환경 시스템으로의 데이터 전달 방법으로는 TCP/IP를 사용하였다. 이 때 전달되는 데이터는 손 제스처 인식의 결과가 구문론적 규칙에 의해 가상 환경 시스템이 요구하는 형태로 변환된 것이

다. 표 1에서 a로 표시한 속도의 증가 및 감소 단위량은 초당 500 pixel로 사용하였으며, 진행 방향의 변경을 위한 회전의 단위량 b는 5 °로 설정하였다.

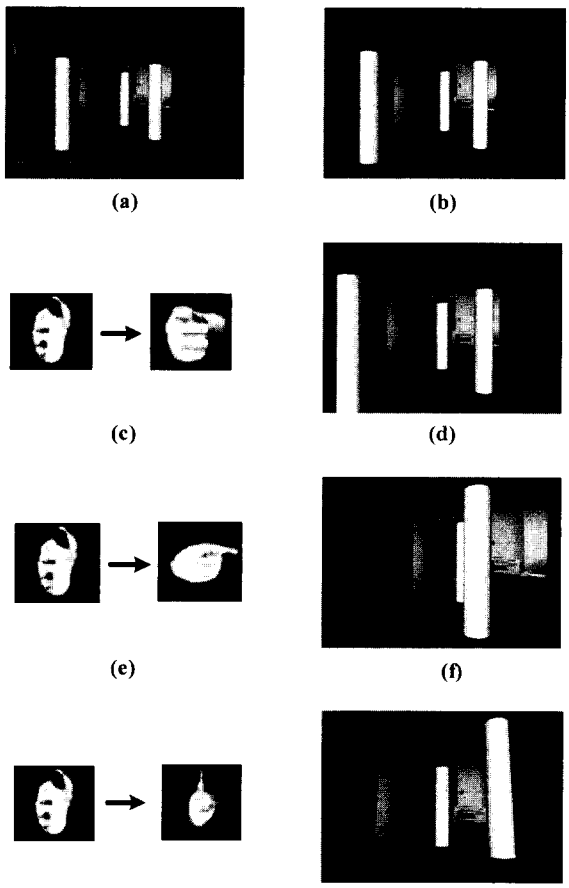


그림 11. 가상 환경에서의 내비게이션 실험 (a) 초기 환경 (b) 1단계 속도로 5초간 진행 후 (c) 속도 변화를 위한 손 제스처 (d) 5단계 속도로 5초간 진행 후 (e) 진행 방향을 왼쪽으로 바꾸기 위한 손 제스처 (f) 진행 방향이 왼쪽으로 회전된 모습 (g) 진행 방향을 위로 바꾸기 위한 손 제스처 (h) 진행 방향이 위로 회전된 모습

Fig. 11. The navigation test in a virtual environment (a) Initial scene (b) The scene after 5 second navigation at 1 level of speed (c) The sequence of hand gesture for changing speed (d) The scene after 5 second navigation at 5 level of speed (e) The sequence of hand gesture for navigation direction change to left (f) The scene turned left (g) The sequence of hand gesture for navigation direction change to upward (h) The scene rotated upward.

그림 11의 (a)는 가상 환경의 초기 모습이며 (b)는 1 단계의 속도로 가상 환경을 5 초간 진행했을 때의 모습을 보여주고 있다. (c)는 가상 환경에서의 속도 변경을 위해 필요한 일련의 제스처를 보여주고 있다. 우선 명령 제스처로 [진행]이 입력되어야 하고, 그런 다음 가속을 나타내도록 정면을 가리키는 지시 제스처 ⑤가 입력되어야 한다. (d)는 이와 같은 과정을 통해 5 단계의 속도로 가상 환경을 5초간 진행한 후의 가상 환경의 모습이다. 그림 11의 (e)와 (g)는 진행하는 도중 방향을 왼쪽과 위쪽으로 바꾸기 위해 입력되어야 할 제스처를 보여주고 있으며 (f)와 (h)는 이에 따라 진행 방향이 바뀌어진 가상 환경의 모습을 보여주고 있다.

제안한 손 제스처 인식 시스템에서 한 프레임을 처리하는데 걸리는 시간은 약 85ms 정도이다. 그런데 사용자가 원하는 제스처를 정확히 표현하는데는 어느 정도 시간이 걸리며, 제스처와 제스처 사이에 불필요한 동작들이 잘못 인식될 경우가 있다. 따라서 실험에서는 동일한 제스처가 10회 반복되어 인식되었을 때, 그 제스처가 실제로 사용자가 원하는 것으로 간주하고 해당하는 동작을 실행하였다. 따라서 하나의 명령이 손 제스처 인식을 거쳐 실제의 3차원 가상 환경으로 실행되는데 걸리는 시간은 각 명령마다 약간의 차이가 있는데, [정지]와 같이 명령 제스처만을 필요로 하는 것은 약 1초가 소요되고, 지시 제스처를 필요로 하는 [진행]은 약 2초에서 2.5초가 소요된다.

VI. 결 론

본 논문에서는 카메라로부터 실시간으로 입력되는 영상으로부터 손 제스처를 인식하여 가상 환경의 인터페이스를 제공하는 시스템을 제안하였다. 가상 환경에서의 내비게이션 및 물체의 직접 조작을 위해 7개의 명령 제스처를 정의하였으며, 이 명령 제스처는 구문론적 규칙에 의해 9개의 지시 제스처와 결합되어서 효율적인 상호작용을 가능하게 하였다. 또한 명령 제스처와 간단한 구문론적 규칙만을 추가하여 더 많은 상호작용을 제공할 수 있다. 손 제스처 인식 시스템은 카메라에서 입력되는 영상의 특징을 기반으로 다른 장치의 조작 없이 자연스러운 가상환경 및 컴퓨터와의 상호작용을 가능하게 하였다. 그리고 기존의 영상의 특징을 기반으로 하는 방법들과는 달리 배경의 제약이

나 표식을 사용하지 않고 복잡한 배경에서 손 영역을 추출해 내었다. 또한 주로 얼굴 인식 등과 같은 신호처리의 응용에 사용되었던 RBF 신경망을 이용하여 각 제스처를 비교적 정확하게 인식해 내었다. 그리고 명령 제스처와 지시 제스처에 대해 각각 다른 RBF 신경망을 사용함으로써 그 구조가 작고 간단하므로 빠른 인식을 수행하였다. 따라서 계산 속도 면에서도 다른 제스처를 추가할 수 있는 충분한 확장성을 가지고 있다. 제안하는 시스템은 펜티엄 PC상에서 DSP 칩과 펜티엄 CPU와의 병렬처리를 통해 초당 12 프레임의 속도로 손 제스처를 인식해냄으로써 가상 환경에서의 상호작용을 위한 빠른 인터페이스를 제공하였다. 손 영역 추출 및 인식 능력과 빠른 연산의 수행은 가상 환경의 인터페이스뿐만 아니라 손 제스처 인식을 기반으로 하는 다른 응용에서도 충분히 사용될 수 있을 것이다.

본 논문에서는 손이 아닌 다른 움직이는 피부색 영역이 있는 경우 이를 제거하는 방법을 보완하는 것이 향후의 과제로 남아있다. 제안한 시스템에서 사용한 제스처들은 일반적으로 허리와 어깨 사이의 범위에서 표현이 가능하므로 카메라의 FOV(field of view)를 이에 맞추면 대부분의 경우 손 영역을 제대로 추출할 수 있지만 손 영역 추출 실험에서 언급하였듯이 보다 보편적인 경우에는 얼굴이 포함될 수 있고, 다른 피부색 영역과 손이 서로 겹칠 수도 있다. 또한 사용자 이외에 다른 사람이 FOV 내에서 움직이고 있는 경우도 발생할 수 있다. 이런 경우 일단 여러 개의 움직이는 살색 영역을 손의 후보 영역으로 검출한 다음 학습을 통해 알고 있는 손 제스처의 패턴을 이용해 원하는 손 영역을 판별해 냄으로써 보다 실제적인 상황에서도 안정적으로 사용자가 원하는 손 제스처를 추출해 낼 수 있을 것이다.

참 고 문 헌

[1] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction : A Review," *IEEE Trans. PAMI.*, vol. 19, no. 7, 1997.

[2] T. S. Huang and V. I. Pavlovic, "Hand Gesture Modeling, Analysis, and Synthesis," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, pp. 73-79, 1995.

[3] F. K. H. Quek, "Toward a vision-based hand gesture interface," *Proc. of the Virtual Reality System Technology Conf.*, pp. 17-29, 1994.

[4] F. K. H. Quek, T. Mysliwiec, and M. Zhao, "Finger Mouse: A Free Hand Pointing Interface," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, pp. 372-377, 1995.

[5] W. T. Freeman and C. D. Weissman, "Television control by hand gestures," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, pp. 179-183, 1995.

[6] R. Kjeldsen and J. Kender, "Visual Hand Gesture Recognition for Window System Control," *Proc. Int'l Workshop on Automatic Face and Gesture Recognition*, pp. 184-188, 1995.

[7] J. A. Adam, "Virtual Reality," *IEEE Spectrum*, vol. 30, no. 10, pp. 22-29, 1993.

[8] D. L. Quam, "Gesture Recognition With a DataGlove," *Proc. IEEE National Aerospace and Electronics Conf.*, vol. 2, 1990.

[9] D. J. Sturman and D. Zeltzer, "A Survey of Glove-Based Input," *IEEE Computer Graphics and Applications*, vol. 14, pp. 30-39, 1994.

[10] W. Skarbek and A. Koschan, "Colour Image Segmentation - A Survey," Technisher Bericht 94-32, Technical University of Berlin, 1994.

[11] J. Moody and C. J. Darken, "Learning with localized receptive fields," *Proc. of the 1988 Connectionist Models Summer School*, Morgan Kaufmann, pp. 133-143, 1988.

[12] J. Moody and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, vol. 1, pp. 281-294, 1989.

[13] J. Park and I. W. Sandberg, "Universal Approximation Using Radial-Basis-

Function Networks," *Neural Computation*, vol. 3, pp. 246-257, 1991.

[13] J. Park and I. W. Sandberg, "Approximation and Radial-Basis-Function Networks," *Neural Computation*, vol. 5, pp. 305-316, 1993.

[14] T. Poggio and F. Girosi, "Networks for Approximation and Learning," *Proc. of the IEEE*. vol. 78, pp. 1481-1497, 1990.

[15] F. Girosi, "Some extensions of radial basis functions and their applications in artificial intelligence," *Computers Math. Applic.*, vol. 24, pp. 61-80, 1992.

[16] J. Howell and H. Buxton, "Invariance in Radial Basis Function Neural Networks in Human Face Classification," *Neural Processing Letters*, vol. 2, pp. 26-30, 1995.

[17] M. Rosenblum, Y. Yacoob, and L. S. Davis, "Human Expression Recognition from Motion Using a Radial Basis Function Network Architecture," *IEEE Trans. Neural Networks*, vol. 7, no. 5, 1996.

[18] Y. S. Hwang and S. Y. Bang, "An Efficient Method to Construct a Radial Basis Function Neural Network Classifier," *Neural Networks*, vol. 10, no. 8, pp. 1495-1503, 1997.

[19] S. Chen, C. F. Cowan, and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Trans. Neural Networks*, vol. 2, no. 2, pp. 302-309, 1991.

[20] K. Stokbro, D. K. Umberger, and J. A. Hertz, "Exploiting neurons with localized receptive fields to learn chaos," *Complex Systems*, vol. 5, pp. 603-622, 1990.

저 자 소 개

曺 浩 永(正會員)

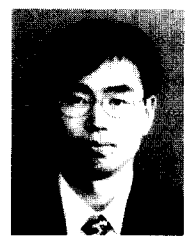
1997년 2월 고려대학교 전자공학과 졸업(학사). 1999년 2월 고려대학교 전자공학과 졸업(공학석사), 주관심분야는 컴퓨터비젼, 인공 지능, 얼굴 및 제스쳐 인식 등임

金 炯 坤(正會員) 第 36卷 S編 第 1號 參照

현재 한국과학기술연구원 영상미디어 연구센터 책임 연구원

高 聖 濟(正會員)

1980년 2월 고려대학교 전자공학과 졸업(학사). 1986년 5월 State Univ. of New York at Buffalo, 전기 및 컴퓨터 공학과(공학석사). 1988년 8월 State Univ. of New York at Buffalo, 전기 및 컴퓨터 공학과(공학박사). 1981년 8월 ~1983년 12월 대한 전선중앙연구소 연구원. 1988년 8월 ~ 1992년 5월 The University of Michigan-Dearborn 전기 및 컴퓨터공하과 조교수. 1996년 11월 IEEE APCCAS best paper award. 1997년 12월 대한 전자공학회 해동 논문상 수상. 1997년 ~ 현재 IEEE Senior member. 현재 고려대학교 전기전자전파공학부 교수. 주 관심분야는 신호 및 영상 처리, 영상압축 및 통신, 멀티미디어 통신 등임.



安 相 喆(正會員)

1988년 서울대 제어 계측 공학과 졸업(학사). 1990년 서울대 제어 계측 공학과 졸업(공학석사). 1996년 서울대 제어 계측 공학과 졸업(공학박사). 1996 ~ 1997년 Univ. of Southern California 초빙 연구원. 현재 한국과학기술연구원 영상미디어연구센터 선임 연구원. 주관심분야는 인공지능, 컴퓨터비젼, 영상처리, 얼굴인식 등임.