
원격 카메라 로봇 제어를 위한 동적 제스처 인식

이주원* · 이병로**

Dynamic Gesture Recognition for the Remote Camera Robot Control

Ju-Won Lee* · Byung-Ro Lee**

요 약

본 연구에서는 원격 카메라 로봇 제어를 위한 새로운 제스처 인식 방법을 제안 하였다. 제스처 인식의 전처리 단계인 동적 제스처의 세그멘테이션이며, 이를 위한 기존의 방법은 인식 대상에 대한 많은 칼라정보를 필요로 하고, 인식단계에서는 각각 제스처에 대한 많은 특징벡터들을 요구하는 단점이 있다. 이러한 단점을 개선하기 위해, 본 연구에서는 동적 제스처의 세그멘테이션을 위한 새로운 Max-Min 탐색법과 제스처 특징 추출을 위한 평균 공간 사상법과 무게중심법, 그리고 인식을 위한 다층 퍼셉트론 신경망의 구조 등을 제안하였다. 실험에서 제안된 기법의 인식율이 90%이상으로 나타났으며, 이 결과는 원격 로봇 제어를 위한 휴먼컴퓨터 인터페이스(HCI : Human Computer Interface) 장치로 사용 가능함을 보였다.

ABSTRACT

This study is proposed the novel gesture recognition method for the remote camera robot control. To recognize the dynamics gesture, the preprocessing step is the image segmentation. The conventional methods for the effectively object segmentation has need a lot of the color information about the object(hand) image. And these methods in the recognition step have need a lot of the features with the each object. To improve the problems of the conventional methods, this study proposed the novel method to recognize the dynamic hand gesture such as the MMS(Max-Min Search) method to segment the object image, MSM(Mean Space Mapping) method and COG(Center Of Gravity) method to extract the features of image, and the structure of recognition MLPNN(Multi Layer Perceptron Neural Network) to recognize the dynamic gestures. In the results of experiment, the recognition rate of the proposed method appeared more than 90[%], and this result is shown that is available by HCI(Human Computer Interface) device for remote robot control.

키워드

HCI(Human Computer Interaction) Remote Robot Control, Gesture Recognition, Image Segmentation, Neural Networks

1. 서 론

제스처(gesture) 인식은 인간과 컴퓨터간의 상호 작용(HCI : Human Computer Interaction)을 위한

한 방법으로 그 응용 분야는 컴퓨터의 키보드나 마우스의 사용이 어려운 장애인 또는 노인들의 의사(command)나 행동을 지원하는 시스템과 원격로봇 제어 그리고 가정에서 인간의 삶에 있어 편리를 제

공하기 위한 지능형 홈 시스템(Smart Home System)등이다. 특히 최근에는 위험한 공간에서 원활한 작업 또는 원격 수술 등을 위한 원격 로봇 제어에 있어 제스처(gesture)인식에 관한 연구가 활발하게 이루어지고 있다. 이러한 원격 로봇제어를 위해 제스처의 영상을 인식하는 방법은 타케시(Takeshi)가 제안한 HMM(Hidden Markov Model) 기법과 야마사키(Yamasaki)가 제안한 시계열 패턴 인식 신경망 기법 등이 있으나 이들의 방법은 알고리즘이 매우 복잡하고 현재 기초 연구단계이다 [1][2]. 그리고 동적 영상(Dynamic Image)을 인식함에 있어 고속 촬영이 가능한 고가의 영상 카메라가 필요하고, 그 해상도 또한 매우 높아야하는 문제점을 가지고 있다. 즉, 저해상도인 초당 30 프레임 이하의 웹 카메라를 이용하여 영상 처리를 할 경우 캡처(capture)된 영상이 흐려지는 블러링(blurring) 현상이 발생하여 많은 영상인식 오류를 발생시킨다. 따라서 일반적으로 많이 사용되는 저속/저해상도의 영상 카메라인 웹(web) 카메라에서도 영상인식 성능이 우수한 시스템에 관한 연구가 필요하다. 그리고 동적 영상 인식의 인식 성능에 있어 가장 중요한 인자는 동적 제스처(dynamic gesture) 영상을 얼마나 효과적으로 세그멘테이션(segmentation) 하는가에 따라 그 인식 성능은 달라진다. 이러한 문제점을 해결하기 위해 노리코(Noriko)는 DPM(Dynamic Program Matching) 기법을 제안하였는데 이 기법은 동적인 제스처 영상을 세그멘테이션을 하는 것으로 반복학습(iteration learning)이 요구되는 단점이 있다[3]. 그리고 찬(Chan)이 제안한

영상 세그멘테이션 기법은 손(hand)의 제스처 영상을 인식하기 위한 기법으로 손에 관한 적색 영상과 녹색 영상의 분산과 평균을 얻기 위해 수백 개 이상의 영상이 필요하다는 단점이 있다[4]. 따라서 본 연구에서는 이러한 문제점들을 해결하기 위해 새로운 제스처 영상 인식 기법을 제안하였다. 제안된 영상 인식기법은 손의 제스처 영상을 인식하는 기법으로서 크게 세단계로 구성되어 있다. 첫 번째 단계에서는 동적인 손의 영상을 최대-최소 탐색법(Max-Min Search Method)을 이용하여 손의 영역을 세그멘테이션하는 단계이고, 두 번째 단계는 세그멘테이션된 손의 제스처 영상을 평균 공간(Mean Space)으로 사상(Mapping)하고 사상된 영상 행렬의 가로축과 세로축 방향으로 공간 투영(projection)하여 각 축을 기점으로 관측된 영상에 너지 벡터의 중심을 무게 중심법(COG : Center Of Gravity)으로 추정하는 특징 추출법이다. 그리고 추출된 특징 벡터를 다층 신경망에 의하여 동적 제스처를 인식하는 단계로 구성되어 있다. 제안된 기법의 성능을 평가하기 위해 원격 카메라 로봇 제어를 위한 5가지 손의 제스처에 대하여 실험하고, 그 인식 성능을 분석하였다.

II. 제스처 인식에 의한 원격 카메라 로봇 제어 시스템

일반적인 제스처 인식을 기반한 원격 로봇제어

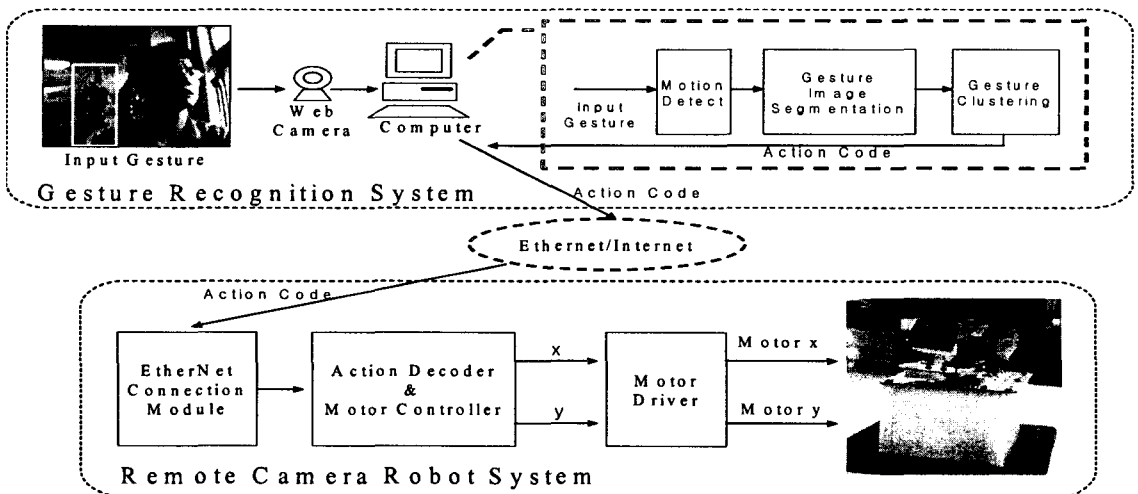


그림 1. 원격 카메라 로봇제어를 위한 시스템의 구조
Fig. 1 The system structure for controlling the remote camera robot

시스템의 구조는 그림 1과 같다. 그림 1은 웹 카메라로부터 제스처 영상을 입력 받아 손의 영역을 세그멘테이션(추출)한 후, 추출된 영상으로부터 손의 자세에 따른 특징을 추출한다. 그리고 그 특징을 기반으로 하여 자세 영상 패턴을 분류하고 분류된 패턴 클러스터에 따라 로봇의 명령 코드를 생성하여 이더넷(ethernet)을 통해 원격의 로봇을 제어하도록 구성되어 있다. 이러한 원격 로봇제어에 있어 가장 중요한 입력 인터페이스는 제스처 영상 인식이며 본 연구의 목표이다. 그리고 본 연구에서는 일반적인 320x240의 8bit 해상도를 가지는 칼라 웹(web) 카메라를 이용하여 원격지의 카메라 로봇의 자세를 제어하기 위한 그림 2와 같이 5가지 제스처 영상을 인식하는 새로운 알고리즘을 제안하고 구현하였다. 제안된 제스처 영상 인식 알고리즘 3절에서 자세히 다루었다.

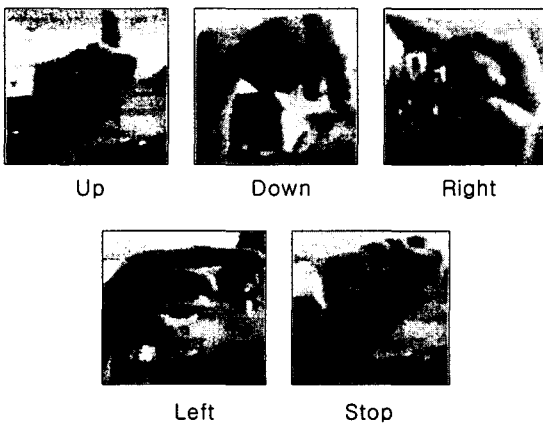


그림 2. 카메라 로봇 제어를 위한 5가지 손의 자세
Fig. 2 Five hand poses for the camera robot control.

III. 제안된 제스처 인식 기법

3.1 움직임 검출 및 세그멘테이션

본 연구에서 제안한 제스처 인식 알고리즘은 다음과 같다. 먼저 영상 카메라로부터 시간 t 간격으로 현재 프레임의 칼라 영상 $I(t, \{r, g, b\})$ 을 캡처(capture)하고 이전에 캡처된 영상 $I(t-1, \{r, g, b\})$ 간의 차(subtraction)와 절대연산(absolute operation) 처리를 하여 움직임이 있는 각 칼라밴드의 영상을 합 연산 처리하여 하나의 움직임 칼라 영상 $I_A(t)$ 을 식(1)과 같이 추출한다.

$$I_A(x, y, t) = \frac{1}{\alpha} \sum_{i=R}^B |I(t, i) - I(t-1, i)| \quad (1)$$

$i = R, B, G, \quad \alpha = \text{scale factor}$

그리고 식(1)에서 매 프레임마다 그 픽셀 값의 진폭과 움직인 영역이 불규칙적이기 때문에 인식 오류를 발생시킨다. 따라서 움직임이 발생하는 영역의 영상을 이진화하고 추출하여야 한다. 본 연구에서는 움직임이 발생한 영상 $I_A(x, y, t)$ 에서 식(2)와 같이 영상의 윤곽선(edge)을 추출하여 이진화 하였다. 여기서 윤곽선 추출에 사용된 마스크 필터는 수평, 수직 방향의 윤곽선 추출 성능이 우수한 프리윗(prewitt) 마스크 $M_p(i, j)$ (식 (3))를 사용하였다.

$$I_M(x, y, t) = I_A(x, y, t) \odot M_p(i, j) \quad (2)$$

$$M_p(i, j) = \left\{ \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \right\} \quad (3)$$

식(2)에서 \odot 는 2차원 컨볼루션(convolution)[5] 연산자이다. 식(2)의 $I_M(x, y, t)$ 영상에서 손의 영역을 세그멘테이션하기 위해 본 연구에서는 $I_M(x, y, t)$ 의 x 축과 y 축으로 투영(projection)하여 각 축방향의 픽셀 평균 벡터 \vec{m}_x, \vec{m}_y 를 식(4, 5)과 같이 구하고 \vec{m}_x, \vec{m}_y 벡터에서 최대값 $x_{max} = f_{max}(\vec{m}_x), y_{max} = f_{max}(\vec{m}_y)$ 의 위치를 손 영상의 중심으로 가정할 수 있기 때문에 식(6)과 같이 그 중심점 h_{mx}, h_{my} 를 추정하였다.

$$m_x(x) = \frac{1}{Y} \sum_{y=1}^Y I_M(x, y, t), x = 1, 2, \dots, X \quad (4)$$

$$m_y(y) = \frac{1}{X} \sum_{x=1}^X I_M(x, y, t), y = 1, 2, \dots, Y \quad (5)$$

$$\begin{aligned} h_{mx} &= \text{index}(f_{\max}(\vec{m}_x)) \\ h_{my} &= \text{index}(f_{\max}(\vec{m}_y)) \end{aligned} \quad (6)$$

식(4, 5)에서 X, Y 는 영상의 x 와 y 의 최대크기이다. 식(6)으로부터 얻은 손 영상의 중심점에서 그림 3(b)와 같이 손의 영역을 세그멘테이션하기 위해 그림 3의 (c)와 (d)의 평균 벡터 \vec{m}_x, \vec{m}_y 에 속해 중심좌표 h_{mx}, h_{my} 에 가장 근접한 $m_x(x)=0, m_y(y)=0$ 인 값을 가지는 영상 좌표를 탐색하는 것

으로 그림 3의 (b, c, d)와 같이 손 영상의 경계점 들인 x_s, x_e, y_s, y_e 를 식(7)로부터 추정할 수 있다.

$$\text{if } \begin{cases} \overline{m_x}(x) = 0, & \text{for } x = h_{mx}, h_{mx} - 1, \dots, 1 \\ \overline{m_x}(x) = 0, & \text{for } x = h_{mx}, h_{mx} + 1, \dots, X \\ \overline{m_y}(y) = 0, & \text{for } y = h_{my}, h_{my} - 1, \dots, 1 \\ \overline{m_y}(y) = 0, & \text{for } y = h_{my}, h_{my} + 1, \dots, Y \end{cases} \\ \text{then } \begin{cases} x_s = x \\ x_e = x \\ y_s = y \\ y_e = y \end{cases} \quad (7)$$

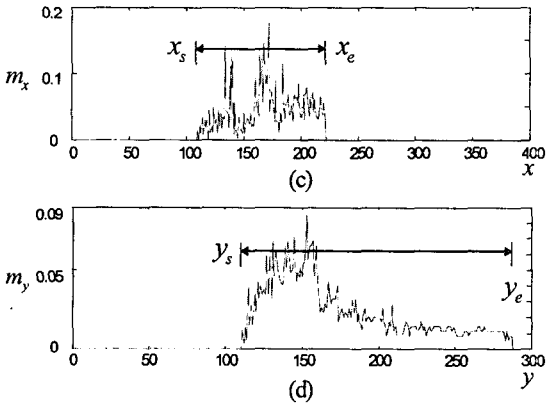
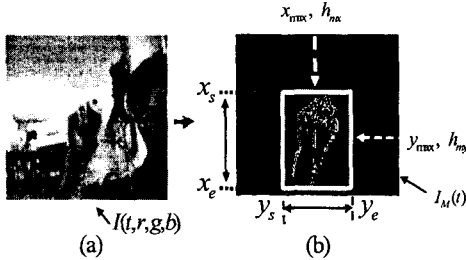


그림 3. 평균벡터 m_x, m_y 에 의한 손 영역 경계점
Fig. 3 Hand area boundaries by average vectors m_x, m_y .

따라서 상기와 같은 알고리즘에 의해 그림 3(b)의 흰색 사각형 안의 제스처 영상 $S(t)$ 만을 이용하여 추출할 수 있다.

$$S(t) = I_M(x_s : x_e, y_s : y_e, t) \quad (8)$$

식(8)은 $I_M(x, y, t)$ 영상 속에 영상좌표 x_s 에서 x_e, y_s 에서 y_e 사이에 있는 부분 영상 $S(n, m, t)$ 을 추출하는 것이다.

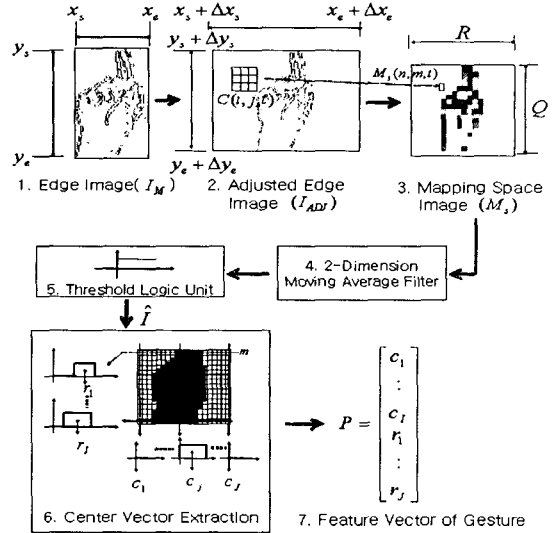


그림 4. 제안된 제스처 특징 추출기법
Fig. 4 The proposed method for the gesture feature extraction.

3.2 제스처 영상의 특징 추출

제안된 세그멘테이션 방법으로 추출된 윤곽(edge) 영상 $S(n, m, t)$ 로부터 제스처 인식을 원활하게 하고 인식 성능을 향상시키기 위해 제스처 영상의 특징 추출이 요구된다. 따라서 본 연구에서는 그림 4의 영상 특징 추출기법을 제안한다. 이 특징 추출기법은 5단계로 구성되어 있으며, 단계 1과 2의 단계는 세그멘테이션한 영상 $S(n, m, t)$ 의 크기가 손의 자세와 카메라 렌즈간의 거리에 따라 일정하지 않기 때문에 일정한 영상크기를 얻기 위한 특징 크기($Q \times R$)의 셀(cell) $C(i, j, t)$ 들로 분할하고 그 셀의 평균 사상 공간(mapping space) $M_s(q, r, t)$ 을 두어 일정한 크기($Q \times R$)의 영상을 만드는 단계이다. 그리고 단계 3과 4는 사상 공간의 영상을 2차원 이동평균필터(moving average filter) 마스크 M_f 처리와 역치 연산하여 윤곽선 추출과정에 고립된 픽셀(point pixel)을 최소화하고, 필링(filling)된 영상 $\hat{I}(q, r, t)$ 을 생성하는 단계이다. 그리고 최종단계인 단계 5에서는 영상 $\hat{I}(q, r, t)$ 에서 손의 방향 성분을 추출하기 행벡터와 열벡터를 추출하여 각 벡터의 무게 중심 좌표 벡터를 각각 추출한다. 이 추출된 벡터는 제스처의 방향을 나타내는 특징 성분이며 이 벡터를 기반으로 하여 제스처를 인식한다. 세부적인 특징추출 절차는 아래와 같다.

Step 1 : 추출된 윤곽 영상으로부터 방향성을 명

확하게 하기위해 식 (9)과(10) 같이 정방행렬 영상으로 재 추출한다.

$$\text{if } \begin{cases} |x_s - x_e| < |y_s - y_e| \\ |x_s - x_e| < |y_s - y_e| \end{cases} \quad (9)$$

$$\text{then } \begin{cases} \Delta x_s = x_s - y_s, \Delta x_e = y_e - x_e, \text{ and } \Delta y_s, y_e = 0 \\ \Delta y_s = x_s - y_s, \Delta y_e = y_e - x_e, \text{ and } \Delta x_s, x_e = 0 \end{cases}$$

$$I_{ADJ} = I_M(x_s + \Delta x_s : x_e + \Delta x_e, y_s + \Delta y_s : y_e + \Delta y_e) \quad (10)$$

Step 2 : I_{ADJ} 을 식(11)과 식(12)의 크기로 셀을 식(13)과 같이 추출하여 식(15)과 같이 평균 사상 공간으로 사상시킨다.

$$a(i) = a(i-1) + \{f_i (\frac{|x_s - x_e|}{Q}) - 1\} \quad (11)$$

$$b(i) = b(i-1) + \{f_i (\frac{|y_s - y_e|}{R}) - 1\} \quad (12)$$

$$P_c = I_{ADJ}(a(i-1) : a(i), b(i-1) : b(i)) \quad (13)$$

여기서 아래의 식(14)의 조건으로 만족하도록 식(11)~(13)을 반복적으로 연산하며, $a(i), b(i)$ 의 초기값은 $a(0)=1, b(0)=1$ 과 사상공간의 크기($Q \times R$)를 $Q=R$ 로 설정 하였다. 그리고 식(15) N 과 M 은 추출된 셀 영상의 크기이다.

$$\begin{cases} i \leq R, i = i + 1 \\ i \geq R, j = j + 1 \text{ and } i = 0 \\ j \geq Q, \text{ Mapping Processing Stop} \\ \text{other, go to step 2} \end{cases} \quad (14)$$

$$M_s(i, j) = \frac{1}{Q} \sum_{n=1}^N \sum_{m=1}^M P_c(n, m) \quad (15)$$

Step 3, 4: 사상 공간의 영상 M_s 의 고립점 픽셀과 필터링 처리를 하기 위해 2-차원 이동평균 필터처리와 역치 연산은 식(16)과 같이 처리하였고, 이동평균필터 마스크는 식(17)과 같다.

$$\hat{I}(r, q) = \begin{cases} (M_s(r, q) \odot F_M) > 0, & 1 \\ \text{other,} & 0 \end{cases} \quad (16)$$

$$r = 1, 2, \dots, R, \quad q = 1, 2, \dots, Q$$

$$F_M = \frac{1}{f_o} \begin{bmatrix} w_{11} & \dots & w_{1j} \\ \vdots & \ddots & \vdots \\ w_{i1} & \dots & w_{ij} \end{bmatrix}, \quad \forall w_{ij} = 1 \quad (17)$$

f_o = 이동평균 필터의 차수

Step 5 : Step 4로부터 얻은 영상 $\hat{I}(r, q)$ 의 각 행과 열벡터 \vec{c}_k, \vec{r}_k 를 추출한다(식(18), 식(19)).

$$\vec{c}_k = \begin{bmatrix} \hat{I}(1, k) \\ \vdots \\ \hat{I}(q, k) \\ \vdots \\ \hat{I}(Q, k) \end{bmatrix}^T, \quad k = 1, 2, \dots, Q \quad (18)$$

$$\vec{r}_k = [\hat{I}(k, 1) \dots \hat{I}(k, r) \dots \hat{I}(k, R)] \quad k = 1, 2, \dots, Q \quad (19)$$

이 벡터는 제스처의 방향 성분으로서 식(20)과 같이 무게 중심법(center of gravity)을 이용하여 제스처 특징 $P(t)$ (식(21))를 추출하였다. 그리고 본 연구에서 다른 5가지 제스처에 관한 특징 추출의 형태는 그림 5에 나타내었다.

$$G_k^x = \frac{\sum_{i=1}^Q c_k(i) i}{\sum_{i=1}^Q c_k(i)}, \quad G_k^y = \frac{\sum_{i=1}^Q r_k(i) i}{\sum_{i=1}^Q r_k(i)} \quad (20)$$

$$P(t) = \begin{bmatrix} G_1^x \\ \vdots \\ G_Q^x \\ G_1^y \\ \vdots \\ G_R^y \end{bmatrix} \quad (21)$$

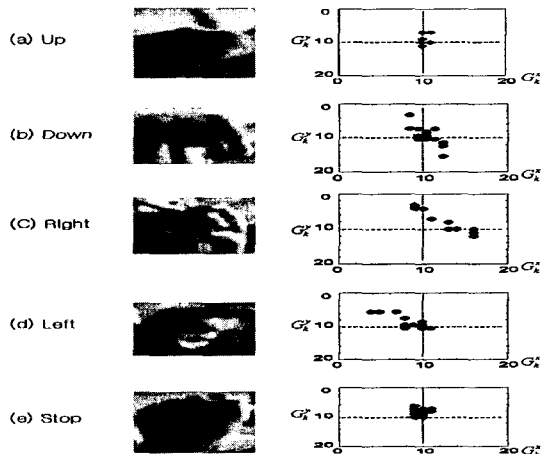


그림 5. 5가지 제스처에 의한 추출된 특징벡터
Fig. 5. The extracted feature vectors with five gestures.

3.3 제스처 인식 신경망

본 연구에서는 제스처 특징 벡터의 함수적인 인식을 위해 함수 근사화에 우수한 성능을 보이는 MLP(Multi-Layer Perceptron Neural Network) 신경망[6]을 사용하였으며, 제스처 인식을 위한 신경망의 구조는 그림 6과 같다. 그림 6의 구조는 특징 벡터의 크기에 따른 Q개의 입력 뉴런과 J개의 은닉 뉴런, K개의 출력 뉴런을 가지며, 신경망의 입력력은 제스처 특징벡터로서 식(22)와 같이 구성하여 입력된다.

$$\vec{P}(t) = [P_1(t), P_2(t), \dots, P_Q(t)]^T \quad (22)$$

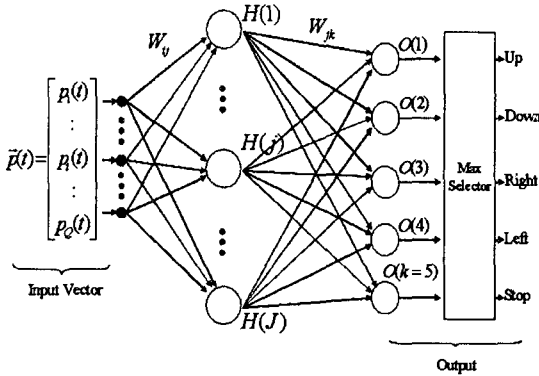


그림 6. 제스처 패턴 인식을 위한 MLP 신경망의 구조
Fig. 6 The structure of MLP neural network for the gesture recognition.

식(23)의 입력 벡터에 의해 신경망의 은닉층 출력은 다음과 같이 구할 수 있다.

$$H(j) = \sum_{i=1}^Q W_{ij} P_i(t) \quad (23)$$

식 (24)에서 신경망의 활성화 함수를 기울기가 1인 선형함수로 설정하여 연산하였다. 그리고 신경망의 최종 출력은 제스처에 따른 로봇의 명령 코드에 대하여 식(25)와 같이 신경망의 최종 출력 벡터 \vec{O} 에서 최대값을 가지는 뉴런의 인덱스를 생성하였다.

$$O(k) = \sum_{j=1}^J W_{jk} H(j) \quad (24)$$

$$R_c(t) = \text{index} \{ \max(\vec{O}) \} \quad (25)$$

그리고 제스처 인식을 위한 학습에 있어 학습 방법은 감독학습을 토대로 신경망의 출력이 원하는 제스처 출력 값과의 오차 E (식(26))를 최소화 되도록 최적의 가중치 W_{ij} , W_{jk} 를 오류역전파 알고리즘[6]을 사용하여 구하였다. 이 학습 알고리즘에 대한 가중치의 변화량 ΔW_{ij} , ΔW_{jk} 는 식(29),(30)과 같이 구할 수 있다.

$$E = \frac{1}{2} \sum_{k=1}^K (T(k) - O(k))^2 \quad (26)$$

$$W_{jk} = W_{jk} + \Delta W_{jk}, \quad W_{ij} = W_{ij} + \Delta W_{ij} \quad (27)$$

$$\Delta W_{jk} = -\eta_1 \frac{\partial E}{\partial W_{jk}} = \eta_1 (T(k) - O(k)) H(j) \quad (29)$$

$$\Delta W_{ij} = -\eta_2 \frac{\partial E}{\partial W_{ij}} = \eta_2 P_i(t) \sum_{k=0}^K W_{jk} (T(k) - O(k)) \quad (30)$$

식(29)~(30)에서 η_1 , η_2 는 학습 상수이다.

IV. 실험 및 결과

연구에서 제안한 제스처 인식 기법의 성능을 평가하기 Mathwork사의 MATLAB을 이용하여 구현하였다. 사용된 영상카메라는 영상크기가 가로세로 320x240이고 칼라 해상도는 8bit인 초당 30프레임을 가지는 VAST WEB 카메라를 사용하였다. MATLAB에서 영상을 캡처하기 위해 Kazuyuki Kobayashi가 개발한 영상 캡처함수인 VCAPG2[7]를 사용하였다. 제안된 기법에 대하여 평가는 제스처 영상 추적율과 제스처 인식률에 대하여 2가지의 항목으로 평가하였다. 제스처 영상을 추적율 평가하기 5가지의 손동작을 100회 취했으며, 이때 제스처 영역을 추적, 세그멘테이션, 윤곽선을 추출한 결과를 그림 7에 나타내었다. 이때 제스처 영상 추적에 대한 오차율은 무게중심법(COG)과 제안된 방법과 비교하여 표1에 나타내었다.

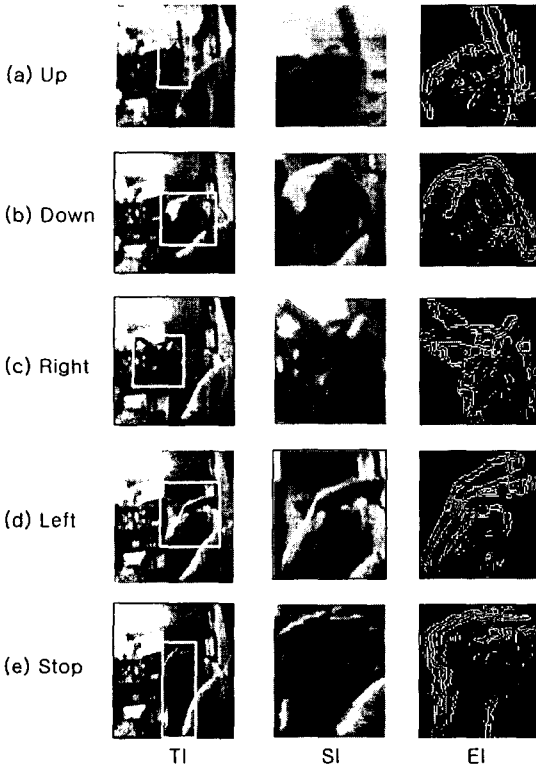


그림 7. 제스처 추적, 세그멘테이션, 윤곽선 추출결과(TI : Tracking Image, SI: Segmentation Image, EI: Edge Image)

Fig. 7 Results of gesture tracking, segmentation, and edge extraction.

표 1. 제스처 영상을 추적한 결과
Table 1. Results of gesture image tracking.

| 기법 | 총 실험 횟수 | 에러수 | 추적수 | 추적율[%] |
|--------|-----------|-----|-----|--------|
| DPM | 100 | 23 | 77 | 77% |
| 제안된 기법 | 100 | 4 | 96 | 96% |
| 성능 개선 | 19% 성능 향상 | | | |

표 1의 결과와 같이 제안된 기법이 COG기법보다 우수한 성능을 보였다. 그리고 그림 7의 제스처에 따른 평균 공간 사상의 영상과 그 특징 벡터는 그림 8과 같이 나타났으며, 이 결과는 제스처의 방향성과 제스처 형태를 육안으로도 판단 가능함을 보였다. 그림 8에 나타난 특징벡터와 맵핑 공간의 크기는 가로, 세로 20x20의 크기로 설정하여 실험

하였다.

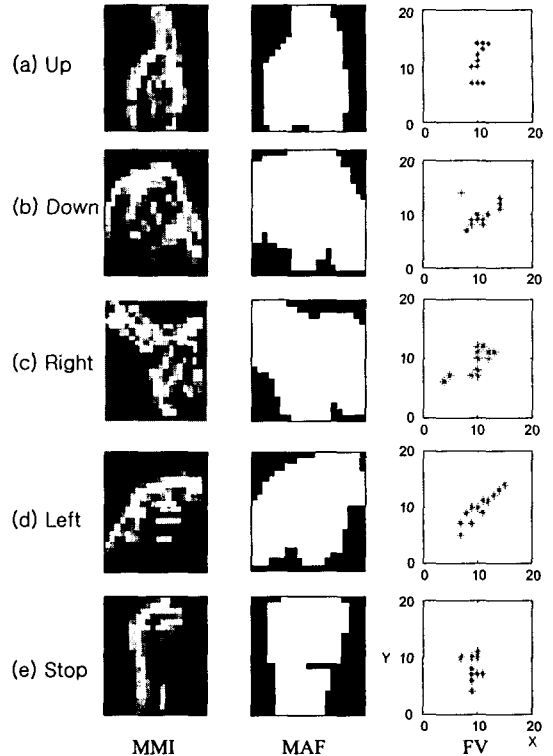


그림 8. 제안된 기법의 특징 추출결과(MMI : Average mapping space image, MAF: results of Moving average filter, FV : Feature vectors)

Fig. 8 Results of the feature vectors extraction by the proposed method.

본 연구에서 제스처 인식을 위해 사용한 MLP 신경망의 구조는 입력 뉴런을 40개, 은닉층 뉴런을 20개, 출력층 뉴런을 10개로 구성한 3층 구조이며, 각 뉴런의 활성화 함수는 선형함수를 사용하였다. 그리고 학습에 있어 100개의 제스처 특징벡터들을 2000회 학습하였고, 학습의 수렴 상수 η_1, η_2 는 각각 0.01로 설정하였다. 학습된 신경망의 가중치 정보들을 이용하여 5가지 제스처를 무작위로 총 100번을 입력하였으며, 인식 결과는 표 2와 같이 나타났다. 표2의 결과에서 "Down" 자세와 "Right" 자세 그리고 "Stop" 자세의 인식결과가 다른 자세보다 많은 에러를 보였는데, 그 이유는 이들 세 가지 자세의 특징벡터들이 중심에 많이 위치하기 때문이라 사료된다.

표 2. 다양한 제스처에 대한 제안된 기법의 인식 결과
Table 2. Recognition results by the proposed method for the different hand poses.

| 자세 | 입력횟수 (100) | 에러 수 | 인식 수 | 인식률 [%] |
|----------|---------------|------|------|------------|
| Up | 24 | 1 | 23 | 95.8 |
| Down | 26 | 4 | 22 | 91.7 |
| Right | 18 | 2 | 16 | 88.9 |
| Left | 17 | 1 | 16 | 94.1 |
| Stop | 15 | 3 | 12 | 80.0 |
| 총 인식률[%] | | | | 90.1 |

V. 결 론

본 연구에서는 원격 로봇제어에 있어 기존의 동적 영상 인식 기법의 문제점들을 개선하기 위한 새로운 영상 세그멘테이션, 특징 추출 및 제스처 인식 기법을 제안하고 그 성능평가를 위해 실험하였다. 그 결과 실험에서 영상추적과 세그멘테이션에 있어 DP기법 보다 19%이상의 성능개선을 보였고, 특징 추출 기법은 기존의 방법보다 적은 특징벡터를 이용하여 90%이상의 인식 성능을 얻었다. 이와 같이 본 연구에서 제안한 제스처 인식 기법은 제스처 인식 시스템의 구현에 있어 많은 손의 영상을 필요로 한 기존의 제스처 추적 및 세그멘테이션 기법보다 구현이 용이할 것으로 사료된다. 그리고 제스처 인식에 있어 상당수의 특징벡터들이 요구된 기존의 방법보다 적은 양의 특징벡터로 인식이 가능하기 때문에 인식 속도향상이 기대된다. 향후 제안된 제스처 인식 기법을 원격 카메라 로봇 제어에 적용하여 그 성능 분석이 요구될 것으로 사료된다.

참고문헌

[1] N. Takeshi, S. Haruyama, T. Kobayashi, HMM-base Human Gesture Recognition, Technical Report of IEICE, pp. 53-59, 1996.
[2] T. Yamasaki, Y. Kataoka, K. Maeyama, K. Nakano, Neural Networkss Memorizing

Sequential Patterns, Technical Report of IEICE, pp. 109-116, 1998.

[3] Noriko Yoshiike, Yoshiyasu Takefuji, Object Segmentation using maximum neural networks for the gesture recognition system, Neurocomputing Vol. 51, pp. 213-224, 2003.
[4] Chan Wah Ng, Surendra Ranganath, Real-Time Gesture Recognition System and Application, Image and Vision Computing, Vol. 20, pp. 993-1007, 2002.
[5] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, Addison Wesley Longman, pp. 192-249, 418-427, 1992.
[6] Chin Teng Lin, George Lee, Neural Fuzzy System, Prentice Hall, 1996.
[7] Kazuyuki Kobayashi, <http://www.ikko.k.hosei.ac.jp/~matlab/matkatuyo/>, Web site.

저자소개

이주원(Ju-Won Lee)



1999년 2월 한국해양대학교 전자통신공학과(공학석사)
2003년 2월 경상대학교 전자공학과(공학박사)
2003년~현재 경상대학교 공학연구원 연구원

※ 관심분야 : HCI, 의용전자, 인공지능, 재활로봇

이병로(Byung-Ro Lee)



1989년 2월 아주대학교 전자공학과(공학사)
1988년~1992년 LG전자
1995년 2월 동아대학교 전자공학과(공학석사)

1998년 2월 경상대학교 전자공학과(공학박사)
※ 관심분야 : 멀티미디어 이동통신, 적응변조, 멀티캐리어 시스템