

論文99-36C-6-7

손 제스처 인식을 이용한 실시간 아바타 자세 제어

(On-line Motion Control of Avatar Using Hand Gesture Recognition)

金鍾成*, 金亭培**, 宋慶俊*, 閔丙義*, 卞增男**

(Jong-Sung Kim, Jung-Bae Kim, Kyung-Joon Song, Byungeui Min, and Zeungnam Bien)

요 약

본 논문에서는 가상 환경에서 움직이는 인체 Avatar의 움직임을 인간의 가장 자연스러운 동작의 하나인 손 제스처를 이용하여 실시간으로 제어하는 인식 시스템의 구현에 관하여 상술한다. 동적 손 제스처는 컴퓨터와 제스처를 사용하는 사람과의 상호 연결 수단이다. 가상공간 상에서의 자연스러운 움직임을 표현하기 위해 32개의 자유도(DOF)를 가진 인체 아바타를 구성하였으며, '정지', '전후좌우로 한 걸음 이동', '건기', '달리기', '좌우로 회전', '뒤로 돌기', '물건 잡기'의 동작 모드를 정의하여 가상공간 상의 인체 아바타는 미리 설정된 손 제스처에 따라 실시간으로 3차원공간상에서 움직일 수 있다. 실시간의 인체 아바타 이동에는 역 기구학과 기구학을 혼용하여 적용하였으며, 사이버 터치를 착용한 사용자의 손 제스처 인식에는 인공 신경망 이론과 퍼지 이론을 도입하여 실시간 인식이 가능하였다.

Abstract

This paper presents a system which recognizes dynamic hand gestures on-line for controlling motion of human avatar in virtual environment(VE). A dynamic hand gesture is a method of communication between a computer and a human being who uses gestures, especially both hands and fingers. A human avatar consists of 32 degree of freedom(DOF) for natural motion in VE and navigates by 8 pre-defined dynamic hand gestures. Inverse kinematics and dynamic kinematics are applied for real-time motion control of human avatar. In this paper, we apply a fuzzy min-max neural network and feature analysis method using fuzzy logic for on-line dynamic hand gesture recognition.

I. 서 론

인간의 손은 주변의 환경과 상호 교류하기 위한 가장 유용하고 다양한 수단의 하나로써 특히, 종래에 키

보드 또는 마우스를 이용하던 사람과 컴퓨터사이의 상호연결과정에 사람이 두 손으로 표현하는 자유로운 몸짓을 직접 이용하려는 시도가 추고되고 있다. 손 운동 자세를 인식하는 분야의 연구는 크레인 조종을 위한 수신호 인식, 지휘자의 지휘동작 인식, 수화(sign language) 인식 등을 대상으로 미국, 일본 등지에서 활발히 추진되는 중이다^[1-3].

특히, 가상 현실에 대한 관심이 높아지면서 가상 공간에서 사용자를 대신하는 아바타의 연구가 활발히 진행되고 있다. 초기에는 단순화한 물체나 손이 아바타

* 正會員, ETRI 멀티미디어 연구부 立體情報研究팀
(MIT Team, Multimedia Dept., ETRI)

** 正會員, 韓國科學技術院 電氣 및 電子工學科
(Department of Electrical Engineering, KAIST)
接受日字:1999年2月11日, 수정완료일:1999年5月8日

가 되었으나 요즘은 인체 모델이 아바타로서 등장하고 있다. 특히 오락이나 훈련과 같은 가상 현실의 응용에서는 자연스러운 아바타 구현을 위해 많은 연구가 진행 중이다. 자연스런 인간 아바타를 구현하기 위해서는 다양한 가상 공간 환경내에서 실제 인간과 같이 움직일 수 있는 이동 능력이 필요하다. 가상현실(virtual reality) 연구분야에서 사용하는 원격 조작 시스템에서 기존의 3차원 제어장치는 상당한 자유도를 가진 장비를 제어하기 위해 많은 전문성을 요구한다. 이와 같은 일들을 효과적으로 하기 위해 사용자와 기계가 3차원 정보를 교환하기 위한 새로운 입력장비가 필요하게 되었고 결국 사람의 손 제스처에 상당한 관심이 집중되었다^[4]. 즉, 인간의 손은 실제 세계(real world)와 교감하는 유용한 툴(tool)로써 뿐만 아니라 가상환경(virtual environment)에 직접 손의 사용을 가능하게 하였다.

본 논문에서는 사용자가 가상환경에서 특정의 손 제스처 명령을 수행하면 그 결과를 실시간으로 인식하여 아바타가 반응하는, 장갑장치에 기반을 둔 아바타 동작 제어 시스템을 구현하고자 한다. 전체 시스템의 실시간성을 고려하여 본 논문에서는 손의 움직임을 감지하는 장비로 Virtual Technologies사의 Cyber-Touch™ 시스템^[5]을 사용하였다.

동적 손 제스처의 인식에 관한 연구는 미국 수화(American Sign Language)를 대상으로 몇 가지 연구가 있었다. Fels^[2]과 Waldron^[3]은 미국수화에서 선택한 몇 가지 손 제스처를 인식하기 위해 신경망을 이용하였는데, 동작의 시작과 끝의 명확한 구분이 필요하며, 특히 많은 학습시간을 필요로 하는 시스템이다. 본 논문에서는 이와 같은 어려움을 극복하기 위하여 퍼지 이론을 이용한 손의 운동 방향 분류기와 P. Simpson^[6]이 제안한 퍼지 최대-최소 신경망을 이용하여 손 모양 분류기를 개발하였다. 이 패턴 분류기는 새로운 집단(class)을 학습할 수 있고 기존 집단의 정보를 없애지 않으면서 온라인으로 빠르게 집단을 분류할 수 있었다.

아바타의 움직임에 대한 연구로써 1993년에 Bruderlin과 Calvert는 보행 동작에 기구학(kinematics) 파라미터법을 적용하여 다양한 대화식의 실시간 인간 이동 패턴을 구현하였다^[7]. Ko와 Badler(1993)은 곡선 경로를 따라 이동하는 인간에 대한 경험적인 보행 동작 데이터의 동역학적인 일반화

를 시도하였다^[8]. 최근에 Witkin과 Popovic(1995)는 애니메이션 경계를 대처하기 위해서 적응적인 보행에 대한 motion warping 기술을 소개하였다^[9]. 반면에 Brudelin과 Williams(1995)는 기존의 모션 데이터를 변형하고 결합하기 위해서 신호처리 기법을 사용하였다^[10]. Unuma와 Anjyo와 Takeuchi(1995)는 원형 데이터로부터 모션 특성을 얻는 방법과 이로부터 새로운 모션을 발생시키는 방법을 기술하였다^[11]. Hodgins, Wooten, Brogan, OBrien(1995)는 달리기, 깡충깡충 뛰기, 도약하기를 포함하는 다리를 움직이는 이동에 대한 연구를 하였고^[12], Girard와 Maciejewski(1985)는 다각의 공동 작용에 대한 연구를 하였다^[13].

로보틱스에서는 Vukobratovic(1990)가 다리가 평탄한 길에서의 걷기, 계단 오르기 동안 일어나는 로봇의 다리를 제외한 다른 인체 운동을 시뮬레이션하였다^[14]. Furusho와 Masubuchi(1986)은 양다리를 이용한 보행 로봇을 만들었지만, 보행 동작에서 안정성은 쉽게 이루지 못하였다^[15].

이동 가능한 가상 인간이 나오는 가상 환경에 대한 연구로서 Pratt 등(1994)은 보병이 출현하는 가상 전쟁을 제공하기 위해서 Industrial Soldier Mobility Simulator(ISMS)를 개발하였다^[16]. 걷기와 달리기의 연속 장면은 움직이는 사람의 얼굴 방향과 속도에 따라서 발생된다. 한편, Granieri와 Badler(1994)는 Tean Tactical Engagement Simulator(TTES)를 개발하였는데, 여기서 산업용 군인의 이동과 자세 변화 방법이 개발되었다. TTES는 이미 저장된 걷기, 달리기, 뒤로 걷기에 대한 장면을 주어진 속도에 맞게 다시 보여줌으로써 이동 동작을 구현하였다.

ISMS와 TTES는 두 시스템 모두 속도 입력에 따른 이동 동작을 조절하는 시스템인데 이들은 정확한 위치 제어나 네비게이션이 필요한 응용 작업에는 적당하지 않다. ISMS는 스텝의 빠르기라는 제어 입력에 대한 응답은 하지만, 위치 에러를 제어하지는 못한다. ISMS는 스텝의 끝에서 최근의 속도라는 제어 데이터에 따라 다음 스텝을 결정한다. 다시 말해서 스텝의 중간에서는 방향이나 속도를 바꿀 수가 없다. TTES는 이전에 저장된 운동의 연속 장면을 재생하는 방법을 사용하는데, 회전 운동과 직선 운동에서 실감이 떨어지는 미끄러짐이 생기기 쉽다. 예를 들어, 입력된 속도 방향이 저장된 속도 방향과 다를 때에는, 인체 전

체가 하나의 고체 덩어리처럼 미끄러져 회전하는 것처럼 보인다.

서울대학교 고희석 교수 연구실에서는 1993년에 VRLOCO라는 가상현실에서 사용하는 이동 동작을 구현하는 엔진을 개발하였다^[17]. VRLOCO는 위에서 언급한 ISMS, TTES와 유사하지만 몇 가지 다른점을 가지고 있다. VRLOCO는 스텝의 빠르기를 입력으로 하기 때문에 하는 스텝의 중간에서 방향이나 속도를 바꿀 수 없는 ISMS와는 달리 스텝의 빠르기보다는 상대적으로 높은 주파수로 처리되는 위치 입력에 따라 직접 제어한다. 그래서 이 경우에는 위치 에러는 축적되지 않아서 좀더 부드럽고, 정확하고, 더 반응에 민감한 이동 제어를 할 수가 있다. 또한 저장된 스텝을 재생하는 TTES와는 달리 VRLOCO는 원형의 이동 데이터를 일반화하는 과정에서 제어 입력을 고려한다. 그 결과 VRLOCO는 다양한 이동 모드가 있지만 회전 운동이나 직선 운동에서 미끄러짐이 생기지 않는다.

II. 아바타 자세 제어를 위한 손 제스처 인식

본 연구는 손 제스처에 의해 이루어지는 모든 명령들을 실시간으로 인식하여 인체 아바타를 실시간으로 이동시킬 수 있는 시스템을 구현하는 것을 목표로 하고 있다.

인체 아바타를 가상 현실에서 사용할 경우에 아바타의 자연스러운 이동은 사용자로 하여금 몰입 효과를 느끼게 할 수가 있다. 본 연구에서는 인체 아바타의 동작 모드를 그림 1처럼 정지, 전·후·좌·우로 한 걸음 이동, 건기, 달리기, 좌·우로 회전, 뒤로 돌기, 물건 잡기로 정의하였다.

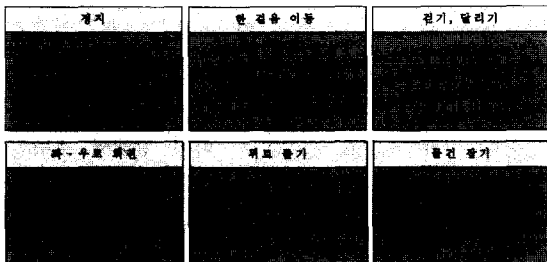


그림 1. 인체 아바타의 이동 제어
Fig. 1. Motion Control of Human Avatar.

본 논문에서는 손 제스처의 데이터를 감지하는 장치로 최근에 가상현실 분야에서 주요한 입력 장치로 사용되고 있는 CyberTouchTM^[5] 시스템을 이용하였는데, 이는 각 손가락 관절의 굽힘 정도를 실시간으로 측정할 수 있다. 또한 장갑의 손등에는 손의 3차원 공간상의 자세(x, y, z, yaw, pitch, roll)를 측정하는 Polhemus센서가 부착되어 있으며 각 손가락에는 진동 바이브레이터가 장착되어 있다. 모두 13개의 손에 대한 정보(손가락 굽힘, 손의 위치)가 1/15초마다 측정되어지고 실시간 그래픽과 함께 인식된 결과가 컴퓨터에 표시된다.

본 논문에서는 가상 현실용 손 제스처로 정의된 11가지 종류의 명령에 대한 동작들을 정의하기 위해 한글 표준사전^[18]을 참고하였다. 그림 2는 본 논문에서 인식하고자 하는 손 제스처에 대한 11가지 명령을 보인 것이다. 본 논문에서는 사용자가 명령을 내릴 때의 편의를 위해 손의 회전은 고려하지 않고 있으며 손 제스처는 다음과 같이 표현된다.

$$G_k = (x_k, z_k, t_k, F_{ki}),$$

$$1 \leq k \leq K : \text{time step}[1/15\text{sec.}]$$

$$F_{ki} : \text{flex angles of right hand, } i=1, \dots, 10.$$

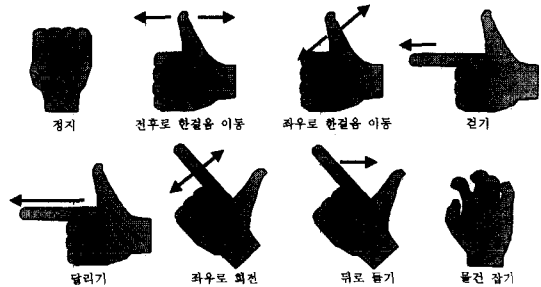


그림 2. 11가지 손 제스처 명령
Fig. 2. 11 Hand Gesture Commands.

본 연구에서는 동적 손 제스처를 두 개의 기본적인 요소(basic elements)로 분류하였는데, 즉, 손의 운동과 손의 모양에 의해 손 제스처를 구성할 수 있다. 본 논문에서는 복잡한 손 제스처를 손 운동 분류기(classifier)와 손 모양 분류기로 구분하여 두 개의 인식 결과를 결합하여 동적 손 제스처 인식을 구성하고자 한다. 이와 같은 방법으로 전체 인식 시스템을 구성할 경우 새로운 손 제스처를 인식기에 추가할 경우 인식 동작의 확장이 쉽게 가능하다. 그림 2를 인식

기 측면에서 분석한 후 기본적인 손 운동 방향으로 그림 3과 같은 5가지의 손 운동 집단을 얻었고 손의 모양은 그림 4와 같이 모두 4종류의 집단이 존재한다. 앞에서 설명한 바대로 손의 모양은 본 시스템의 특성상 회전에 관계없이 같은 집단으로 분류하게 된다.

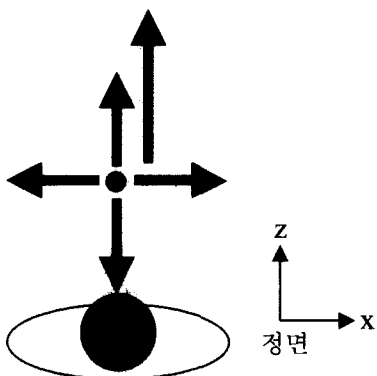


그림 3. 5가지 기본 손운동
Fig. 3. 5 Hand Motion-directions.

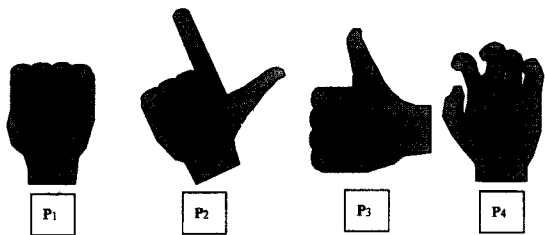


그림 4. 4가지 기본 손모양
Fig. 4. 4 Hand Postures.

손 운동은 3차원 공간에서 이루어지기 때문에 사람마다 같은 동작이라도 변화가 다양하며 같은 동작을 반복해서 할 때에도 다른 데이터를 얻게되는 등 시간에 따른 변화도 함께 존재한다. 또한, 손가락의 외형 및 물리적 구조가 사람마다 다르기 때문에 다른 두 사람에 의해 만들어진 같은 손 모양도 같은 센싱장비에 의해 측정될 때 다른 측정값을 나타낸다. 다양한 데이터의 특성을 나타내고 있는 동적 손 제스처의 인식을 위해 본 논문에서 4가지의 손 모양은 P. K. Simpson이 제안한 퍼지 최대·최소 신경망 분류기^[6]를 사용하였다.

1. 속력 변화를 이용한 제스처 분리 및 손 운동의 분류
제스처의 의미를 알기 위해서는 먼저 연속적으로

이루어지는 제스처들을 단일한 제스처로 분리하는 작업이 이루어져야 한다. 제스처가 시작해서 진행을 하고 끝이 나기까지는 속력의 변화가 있기 마련이다. 제스처의 시작은 정지로부터 속력이 커지는 특징을 가지고 있으며, 제스처의 진행은 속력이 계속 크며, 제스처의 끝은 속력이 큰 상태에서 정지로 변해간다. 이러한 특징을 통해서 제스처의 처음과 끝을 알 수 있다.
이것은 다음과 같이 표현 할 수 있다.

```

Max_Speed = Max((x(k) - x(k-1))^2, (z(k) - z(k-1))^2)
if((start == false) && Max_Speed >= Thr){
    Origin.x = x(k);
    Origin.z = z(k);
    start = true;
}
else if((start == true) && Max_Speed >= Thr){
    Current.x = x(k);
    Current.z = z(k);
}
else if((start == true) && Max_Speed < Thr){
    Final.x = x(k);
    Final.z = z(k);
    Decision(Origin, Final);
    start = false;
}
    
```

여기서는 손위치의 x, z 좌표인 $x(k)$ 와 $z(k)$ 만을 고려했는데, 정의된 제스처에는 상하 운동이 없기 때문이다. 위 표현은 현재 제스처가 이미 시작 했는지와 현재 손이 움직이고 있는지를 판단하여 제스처의 시작인 $Origin.x$, $Origin.z$ 와 제스처의 끝인 $Final.x$, $Final.z$ 을 알아내고 있다. 제스처의 끝부분에서는 $Decision$ 이라는 함수를 통해서 손운동의 종류를 알아낸다.

손운동을 분류하는 모듈에서는 제스처의 시작과 끝의 좌표값을 보고 손이 움직인 방향과 크기를 판단하게 된다.

2. 퍼지 최대·최소 신경망 분류기에 의한 손 모양의 분류

앞에서 설명한 것처럼 사람의 손은 손가락의 굽힘에서 다양한 변화가 존재하고 사람들마다 다양한 크기,

모양의 손가락 구조를 가진다.

본 논문에서는 그림 4에 보여진 4가지의 기본 손 모양을 P. K. Simpson의 퍼지 최대·최소 신경망 분류기^[6]를 이용하여 실시간으로 분류하고자 한다. 사용된 소속함수는 다음과 같이 표시할 수 있다.

$$b_j(F, V_j, W_j) = \frac{1}{10} \sum_{i=1}^{10} [1 - f(F_j - w_{ji}, \gamma) - f(v_{ji} - F_j, \gamma)]$$

$$f(x, \gamma) = \begin{cases} 1, & xy > 1 \\ xy, & 0 \leq xy \leq 1 \\ 0, & xy < 0 \end{cases}$$

여기서, $F_j, j=1,3,\dots,9$ 는 손가락의 첫 번째 관절의 굽힘 정도이고 $F_j, j=2,4,\dots,10$ 은 두 번째 관절의 굽힘 정도를 나타내고 있다.

III. 가상 환경 내의 아바타 자세 제어

1. 인체 아바타

인체 아바타는 그림 5처럼 많은 수의 자유도(DOF)를 가지고 있으며, 이것을 기본 골격으로 기구학(kinematics)을 적용하여 인체 아바타의 중심 위치와 각 관절의 각도를 계산함으로써 인체 아바타의 자세 제어가 가능하다.

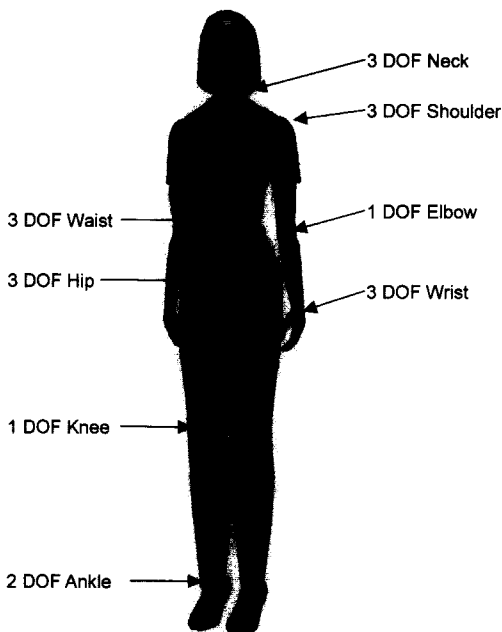


그림 5. 인체 아바타의 자유도
Fig. 5. Degree Of Freedom of Avatar.

1) 인체 아바타 모델

인체 아바타 모델은 인체를 여러 개의 부분으로 구분하여 이것들을 합한 것이다. 각 부분들은 머리, 가슴, 배, 양팔, 양다리로써 양팔은 각각 상박, 하박, 손으로 구성되고 양다리는 각각 허벅지, 종아리, 발로 구성되어 총 15개의 부분들로 나뉜다. 본 논문의 아바타 모델은 목 3자유도, 어깨 3자유도, 팔꿈치 2자유도, 팔목 2자유도, 허리 3자유도, 엉덩이 3자유도, 무릎 1자유도, 발목 2자유도 등 총 32개의 자유도를 갖고 있다. 또한 각 관절은 지역 좌표계를 가지고 있고 각 자유도에 해당하는 운동을 할 수 있다.

2) 인체 아바타 편집기(editor)

본 논문에서는 인체 아바타 모델의 동작을 편집하고 그 결과를 수치와 그림으로 보여주기 위해서 인체 아바타 편집기를 제작하였다. 인체 아바타 편집기는 여러 개의 부분 구성체들로 구성된 아바타를 각 구성체별로 움직임을 조절할 수 있도록 되어있다. 이 에디터는 가상 공간에서 아바타를 움직이도록 하기 전에 먼저 각 동작에 대해서 어색한 부분이나 자연스럽지 않게 구성된 신체 부위 등을 미리 찾고 재구성하는 작업과 함께 실제 동작을 미리 편집하는 용도로 사용될 수 있다.

2. 아바타 자세제어 알고리즘

사람의 골격을 간략화하여 생성된 관절체는 기구학 사슬(kinematics chain)로 나누어 표현할 수 있다. 이러한 기구학 사슬에서 생기는 문제는 두가지로 분류할 수 있는데, 하나는 각각의 관절의 회전각이 주어진 경우, 각 관절의 위치 정보와 end effector의 위치 정보를 찾는 기구학 문제이며, 반대의 경우인 다른 하나는 end effector의 위치 정보로부터 기구학 사슬의 관절의 회전각을 찾는 역기구학 문제이다.

역기구학에서 사용자가 원하는 해는 아래의 식을 사용 한다.

$$\Delta \vec{\theta} = J^+ \Delta \vec{X} + (I - J^+ J) \vec{Z}$$

여기서 J^+ 는 $(I - J^+ J)$ 는 투사 연산자(projection operator)이고 \vec{Z} 는 6-space에 존재하는 임의의 벡터로서 최적화 하고자 하는 함수를 나타낸다.

이 방법은 회전각의 변화가 충분히 작은 경우를 가정하고 있으므로, 한 번의 계산이 끝난 후에는 다음의 반복을 위해서 다시 Jacobian 행렬을 구해야 한다. 일

반적으로 유사 역행렬을 구하기 위해서는 $O(n^2)$ 연산이 필요하다.

본 연구에서 인체 아바타의 이동 동작을 구현할 때에, 주변에 물체가 주어지지 않을 경우에는 주어진 각도만큼 관절을 구부려서 걷는 동작을 구현하였고, 여기에서 기구학이 적용된다. 하지만 매 프레임마다 기구학을 적용하기에는 계산량이 많기 때문에 키프레임에 대해서만 기구학을 적용하고 나머지 프레임은 보간법을 사용한다.

그러나 주변에 부딪히는 물체가 있다거나 물건을 잡는 경우에는 특정한 위치로 end effector를 이동해야 하고 이를 위해서는 각 관절의 각도를 계산해야 하기 때문에 여기서는 역기구학이 적용된다.

3. 기본 동작 모드를 위한 유한 오토마타

인체 아바타의 이동은 한 동작 모드에서 다른 동작 모드로 전환을 할 때에도 자연스러운 처리가 되어야 한다. 본 연구에서는 정지에서 걷기, 걷기에서 정지, 정지에서 달리기, 달리기에서 정지, 걷기에서 달리기, 달리기에서 정지로 동작 모드가 바뀔 때 자연스런 처리가 되도록 과도 상태를 두어서 이 상태를 거쳐서 다음 단계로 가도록 하였다. 정지 상태에서 걷기 명령이 들어오면, 정지와 걷기의 과도 상태를 거쳐서 걷기 상태로 가며 달리기 입력이나 정지 입력이 들어오거나 장애물을 만나기 전까지는 걷기를 유지하게 된다. 달리기도 동일한 과정을 거쳐서 걷기나 정지 입력이 들어오거나 장애물을 만나기 전까지는 걷기를 유지한다.

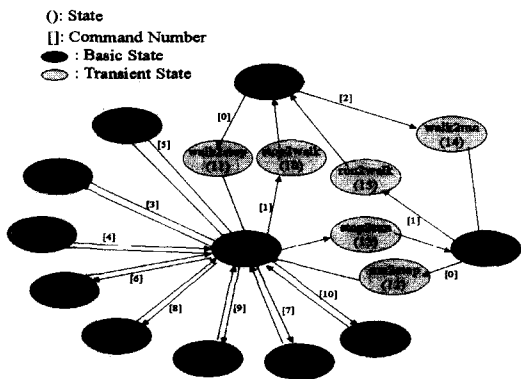


그림 6. 동작 모드에 대한 유한 오토마타
Fig. 6. Finite Automata for the motion mode.

이것을 유한 오토마타로 구성을 하면 그림 6과 같다. 여기서 상태는 과도 상태 6개와 이동 모드 상태

11개를 포함해서 17개이다. 입력은 각 이동 모드의 수와 같은 11개이다. 입력으로 0번은 정지, 1번은 걷기, 2번은 달리기, 3번은 좌회전, 4번은 뒤로 돌기, 5번은 우회전, 6번은 좌로 한걸음 이동, 7번은 앞으로 한걸음 이동, 8번은 뒤로 한걸음 이동, 9번은 우로 한걸음 이동, 10번은 물건 잡기이다.

4. 경로 찾기

인체 아바타의 이동은 아이콘으로 된 11가지의 명령을 선택함으로써, 그리고 제스처 입력으로도 이동이 가능하지만 복잡한 경로를 연속해서 매번 명령하는데는 불편함이 있다. 뿐만 아니라 사용자의 몰입감을 위해서 아바타가 이동하는 대로 카메라가 따라서 같이 이동하기 때문에 전체 가상환경 중에 현재 아바타의 위치를 조감하기에는 어려움이 있다. 이를 위해서 전체 가상환경이 간단하게 묘사된 그림 7과 같은 비트맵 윈도우를 구성하였는데, 이 윈도우에서 움직이는 아바타를 보면 현재의 위치와 방향을 알 수가 있다.

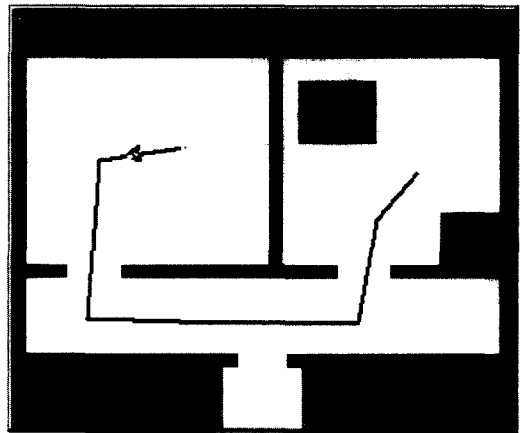


그림 7. 비트맵 윈도우
Fig. 7. Bitmap Window.

또한 이 윈도우에서 아바타의 이동경로를 선택할 수가 있는데, 이때 얻어지는 것은 목표위치들의 열이다. 목표 위치는 N 개의 x 와 z 좌표의 쌍으로 구성되어 있었는데, 아바타는 장애물에 부딪히지 않는 한 N 개의 목표 위치로 차례로 이동하게 된다. 이동의 과정은 먼저는 다음 목표 위치로 방향을 바꾸는 회전을 먼저 수행하고, 그 후에 목표 위치를 향해 걷는다. 그 목표에 도착을 하면 다음 목표 위치로 다시 방향을 바꾸고 그 목표 위치를 향해 걷는다. 이런 과정을 N 번째 목표 위치에 도착할 때까지 반복을 한다.

$$\theta = \text{atan2}(\text{target}.x - \text{present}.x, \text{target}.z - \text{present}.z)$$

$$\text{direction} = \theta - \phi$$

여기서 $\text{target}.x$ 와 $\text{target}.z$ 는 목표 위치의 x 와 z 좌표이고, $\text{present}.x$ 와 $\text{present}.z$ 는 현재 위치의 x 와 z 좌표이다. 이로부터 계산된 θ 는 현재 위치로부터 다음 목표 위치까지를 잇는 벡터의 각도이고, ϕ 는 현재 인체 아바타가 바라보고 있는 방향각이다. 이 각을 뺀 각도가 인체 아바타가 현재 회전해야 할 각도이다.

5. 물건 잡기

본 논문은 인체 아바타의 이동 모드뿐만 아니라 탁자 위의 물건을 잡는 모드도 수행할 수 있도록 되어 있다. 물건 잡기 명령이 들어 왔을 때, 인체 아바타는 물건과 멀리 떨어져 있다 하더라도 물건을 향해서 걸어가서 물건을 잡아서 들어올리는 일련의 행동을 한다. 일단 물건을 들었을 때는 다른 이동 명령, 즉 걷기, 달리기, 회전, 한 걸음 이동을 해도 물건을 계속 들고 다닐 수 있다. 물건 놓기 명령이 들어오면, 인체 아바타는 물건을 탁자 위에 올려놓는다. 이런 과정을 블록도로 구성하면 그림 8과 같다.

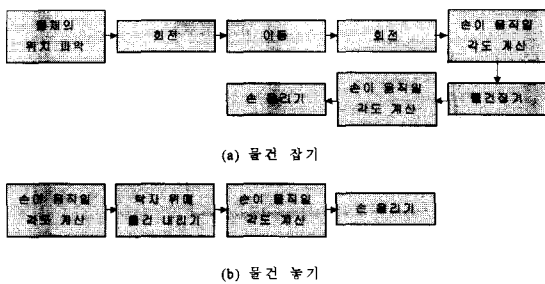


그림 8. 물건 잡기와 물건 놓기의 과정
Fig. 8. Processes of Grabbing and Setting down the Object.

물건 잡기의 과정은 다음과 같다. 여기서 물체의 위치는 미리 알고 있어야 한다. 아바타가 물체와 멀리 떨어져 있을 때 아바타는 경로 찾기에서 사용된 방법으로 목표 위치 방향으로 회전하고, 그 위치로 이동을 하고 다시 물체 방향으로 회전을 하게 된다. 이동이 모두 끝났으면 역기구학을 이용하여 손의 관절이 움직일 각도를 계산하게 된다. 계산된 각도에 따라 물건을 잡고 다시 손의 관절이 움직일 각도를 역기구학을 이용해 구해서 물건을 들어올린다.

한편 물건 놓기의 과정은 다음과 같다. 여기서 인체

아바타는 물건을 놓을 적당한 위치에 있어야 한다. 탁자 앞에서 역기구학을 이용해서 손의 관절이 움직일 각도를 계산하여 들고 있던 물체를 탁자 위에 내려 놓는다. 다시 역 기구학을 이용해서 초기 자세의 손위치로 손을 이동한다.

6. 충돌 검사

인체 아바타는 가상 환경 내에서 이동 및 물건 잡기를 수행하므로, 벽이나 테이블과 같은 가상 환경에서의 장애물에 대해서 적절한 동작을 취할 수 있어야 한다. 본 논문에서는 인체 아바타가 이동 중에 벽이나 탁자를 만나면 동작을 멈추고 정지 모드로 들어가도록 하였다. 인체 아바타의 절대 좌표와 가상 환경 내의 장애물에 대한 절대 좌표 값을 비교하여 물체들이 서로 겹치지 않도록 하였다. 충돌시에는 사용자의 몰입감을 위해서 벽에 부딪히는 소리를 발생하도록 하였다.

7. 가상현실의 실감을 위한 효과

실감 있는 가상현실을 구현하기 위해서 본 논문에서는 아바타가 이동하는 대로 따라다니며 아바타의 시선대로 환경을 바라보는 카메라를 사용하였다. 그러나 때로는 전체 환경을 바라보는 것이 더 편리할 때에는 카메라의 위치를 알맞은 장소로 이동하도록 하였다. 뿐만 아니라 가상 환경 내에 다른 방으로 이동을 할 때에는 음향 효과가 있는 자동문을 사용하여 보다 현실감을 느끼도록 하였다.

IV. 실험 결과

1. 손 제스처 명령의 인식 및 아바타 자세 제어

인체 아바타의 이동 제어는 4가지 작업으로 분류될 수가 있다. 첫째, 인체 아바타가 이동하는 가상 현실내의 가상 환경이다. 여기서는 실내로 작업공간을 만들었다. 둘째, 손 제스처를 인식 하는 부분이다. 셋째, 이동 제어 계산 부분이다. 넷째, 손 동작과 이동 제어를 이어주는 전체 운용 알고리즘 부분이다.

가상 환경은 인체 아바타가 걸어가고, 달리고, 앉고, 물건을 잡을 수 있는 실내로 구성이 되어 있다. 손 제스처 인식 부분은 손 모양과 손 운동 분류기를 통해서 나온 결과로 11가지 동작을 인식하고 있다. 이동 제어 계산 부분은 기구학과 역기구학을 이용하였다. 전체 운용 알고리즘은 손제스처의 입력을 받아서 손 동작

인식 부분에 보내고, 여기서 판단된 이동 동작에 따라서 기구학과 역 기구학에 입력될 데이터를 결정한다.

사용자(signer)가 손 제스처 명령을 시작함에 따라 CyberTouch™ 로부터 많은 데이터가 시스템에 입력된다. 모든 동작의 시작은 3차원 공간상의 임의의 위치에서 이루어 질 수 있으며 특징 추출과정에서 미소 한(1인치 이하) 움직임은 무시되기 때문에 가상 현실 공간에서 작업할 때 생기는 사용자의 약간의 손 떨림, 작은 손의 움직임 등은 무시할 수 있게 된다. 일단 미소 범위 이상의 동작이 시작되면 속력 변화를 이용한 제스처 분리 과정이 진행된다. 이렇게 분리된 제스처는 손 움직임에 따라 5가지 손 운동 중에 하나로 결정이 된다. 일단 현재의 동작에서 손 운동 방향이 정해지면 이때 손의 모양에 대한 정보가 앞에서 설명한 퍼지 최대-최소 신경망 분류기에 입력되어 4가지 손 모양 중의 하나로 분류가 된다. 이 두개의 결과를 결합함으로써 온라인으로 동적 손 제스처 동작을 인식하게 된다.

이동 제어는 키프레임 기법을 적용하여 중요 프레임에 대해서만 기구학 및 역 기구학을 적용하고, 나머지 프레임에 대해서는 보간법(interpolation)을 적용하여 계산 속도를 향상시켰다. 또한, 이 부분에서는 걷기, 달리기, 한걸음 옮기기, 돌기, 멈추기의 각 이동 모드에 따라서 유한 오토마타를 구성해서 센싱 구역 내에 물체가 있을 때와 새로운 외부 입력에 대한 대처 방법을 만든다.

손제스처 인식을 이용한 실시간 아바타 자세 제어에 대한 전체 시스템 블록도를 보면 그림 9와 같다. 먼저 사이버터치에서 제스처 입력이 들어오면 속력 변화를 이용해서 제스처를 분리한다. 이렇게 분리된 제스처의 손 운동과 손 모양을 판단하여 인체 아바타의 동작 모드가 인식된다. 같은 손 운동과 손 모양을 가진 걷기와 달리는 손 운동의 속도에 따라 구분이 된다.

다음 단계에서는 현재의 모드와 위치를 통해서 다음 동작의 방향과 속도를 정한다. 그 다음 단계에서는 이동 모드와 속도에 따라서 한 개 단위의 동작이 완성되는 스텝의 갯수와 동작 끝의 위치 좌표를 정하게 된다. 그 다음 단계에서는 기구학을 통해 동작 도중에서의 몸 중심의 좌표와 몸 관절의 각도를 구한다. 그 다음 단계에서는 벽이나 탁자와 같은 장애물에 대해 충돌을 검사하게 된다. 충돌이 없으면 인체 아바타의 움

직임은 3D 그래픽으로 자연스럽게 나타나게 된다. 충돌이 생기면 정지 동작을 취하고 다음 제스처 입력을 기다리게 된다.

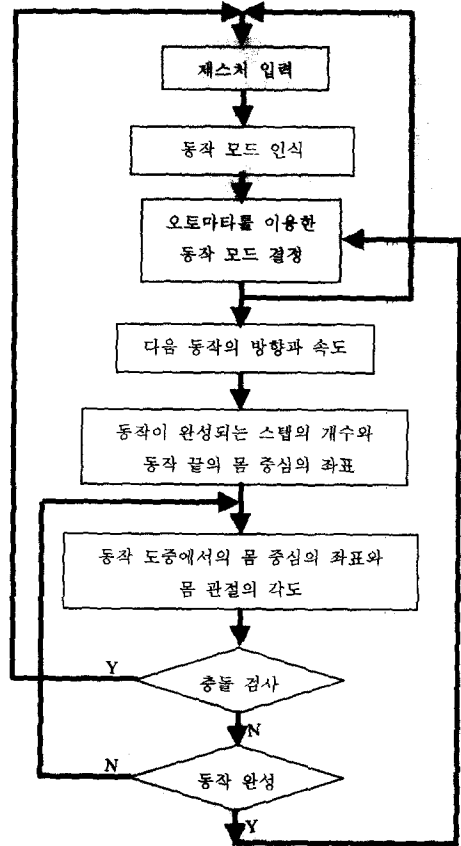


그림 9. 전체 시스템의 블록도
Fig. 9. Block diagram of system.

‘걷기’나 ‘좌로 회전’과 같은 동작이 완성 되었는지의 여부를 파악해서 동작이 완성되지 않았으면 다음 동작을 계속 진행한다. 동작이 완성되었으면 위에서 언급한 유한 오토마타를 통해서 다른 동작 모드를 결정한다. 유한 오토마타의 입력은 현재의 동작 모드 상태와 현재의 제스처 입력에 따른 새로운 동작 모드이다. 이 입력에 따라서 유한 오토마타는 다음 동작 모드를 결정하거나 동작을 마치게 된다.

2. 실험 결과

아바타 자세 제어를 실현하기 위한 전체 가상 환경은 그림 10과 같다.

가상 환경은 현관과 두 개의 방과 이들을 잇는 복도로 구성되어 있다. 복도에는 세 개의 자동문이

있으며 자동문을 통해서 현관과 각 방으로 이동할 수가 있다. 한 방은 물체 잡기 및 놓기를 할 수 있도록 물체의 탁자가 있으며 다른 방은 걷기, 뛰기를 할 수 있도록 넓은 방을 구성하였다.

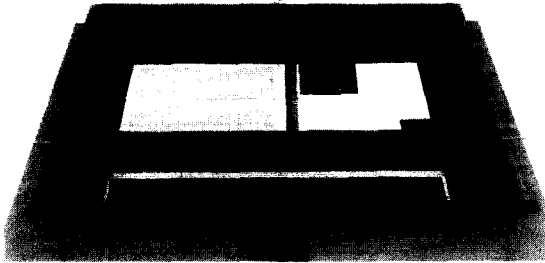


그림 10. 가상 환경
Fig. 10. Virtual Environment.

그림 11은 본 논문의 실험을 위한 가상 현실용 손 제스처 인식 시스템의 외관을 나타낸 것이다. 전체 시스템은 주(main) 컴퓨터로 InterGraph 워크스테이션(workstation)을 사용하였고 CyberTouch™는 38,400 baud rates로 RS232 통신을 통해 주 컴퓨터에 13종류의 데이터(10개의 손 굽힘, 3개의 위치)를 보낸다.

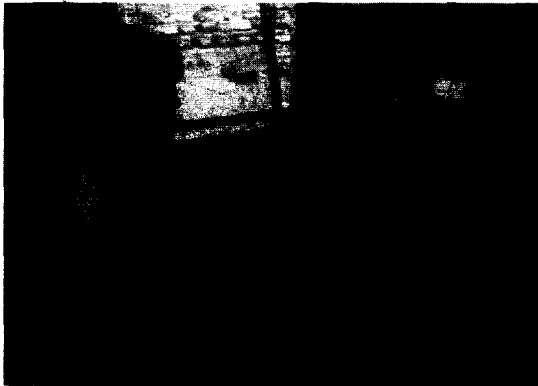
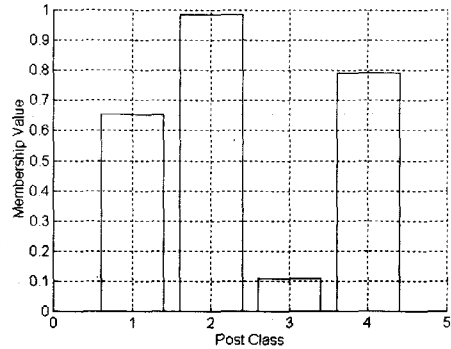


그림 11. 손 제스처 인식 시스템
Fig. 11. Hand gesture recognition system.

그림 12는 엄지와 검지손가락을 펴고 손을 앞으로 내미는 '걷기'라는 동작(그림 2 참조)에 대한 인식 결과를 보인 것이다. 그림 12(a)는 손의 모양에 대한 인식 결과이다. 손가락 정보가 FMMNN에 입력되어 그림과 같은 결과를 얻는다. 그림 12(a)에서 수평축에 있는 숫자는 그림 4에 있는 상응하는 손 모양의 숫자와 관계된다. FMMNN에서 사용된 변수는 $\theta = 0.2$, $\gamma_1 = 4.0$, $\gamma_2 = 8.0$ 이고 $\delta = 0.90$ 이다. 그림 12(a)

에서 보는 바와 같이 2번째 집단(P_2)이 역치(δ)를 초과하는 최대값을 가지고 따라서 두 결과를 이용하여 그림 12(b)에서의 같이 아바타가 앞으로 움직임을 볼 수 있다.



(a)



(b)

그림 12. 실험결과 '걷기'
Fig. 12. Experimental result 'Walk'.

10가지의 손 제스처 명령에 대하여 5명의 다른 실험자를 대상으로 10회의 반복 실험을 수행하였고 98.0%의 평균 인식율을 얻었다. 인식 오류는 주로 이용자의 손 제스처 동작 숙지의 미숙, 동작 중에 발생한 동작 미숙, 그리고 비주기적인 센서의 오동작 등에서도 기인하고 있다.

V. 결론 및 추후과제

본 논문은 컴퓨터와 인간의 상호 접속에서 가장 중요한 매개체가 되고 있는 손 제스처의 인식을 위해 복잡한 사람의 손 제스처 중에서 가장 체계적인 한글 수화에서 선택한 8가지 손 제스처를 대상으로 인식 대상의 확장이 용이한 기본 요소(손의 방향, 손의 모양)

분류기에 의한 인식 알고리즘 및 이를 이용한 아바타 자세 제어 알고리즘을 개발하였다.

본 연구는 서론에서 언급한 일반적인 손 제스처의 인식 시스템(수화 인식, 수신호 등) 구현에도 쉽게 응용 가능하며, 앞으로 인식가능 동작의 확대를 위한 기본 요소의 확장 및 정적인 동작과의 결합을 통해 연속적으로 인식할 수 있는 시스템 및 양손에 의한 손 제스처의 확장 시스템을 구현하고자 한다.

참 고 문 헌

- [1] David J. Sturman, *Whole-Hand Input*, Doctoral Dissertation of MIT, Feb. 1992.
- [2] S. Sidney Fels and G. E. Hinton, "Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesizer," *IEEE Trans. Neural Networks*, vol. 4, no. 1, pp. 2-8, Jan. 1993.
- [3] M. B. Waldron and S. Kim, "Isolated American sign language recognition system for deaf persons," *IEEE Trans. on Rehabilitation Engineering*, vol. 3, pp. 261 - 271, Sept. 1995.
- [4] D. Kessler, L. Hodges, et. al, "Evaluation of the cybergloveTM as a whole hand input device," *ACM Trans. on CHI*, vol. 2, pp. 263-283, Dec. 1995.
- [5] Virtual Technologies, Inc., *Cyber-TouchTM User's Manual*, CA, 1997.
- [6] P. Simpson, "Fuzzy Min-Max Neural Networks-Part 1 : Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 776-786, Sep. 1992.
- [7] Bruderlin, Calvert, "Interactive animation of personalized human locomotion", *Proceedings of Graphics Interface 93*, pp.17-23, Toronto, Canada, 1993.
- [8] Ko, H., & Badler, N. I., "Intermittent non-rhythmic human stepping and locomotion", *Proceedings of the First Pacific Conference on Computer Graphics and Applications, Pacific Graphics 93*, pp.283-300, Seoul:World Scientific, 1993.
- [9] Witkin, A., & Popovic, Z., "Motion warping", *Computer Graphics, (SIGGRAPH 95 Proceedings)*, pp.105-108, 1995.
- [10] Bruderlin, A., and Williams, L., "Motion Signal Processing", *Proceedings of SIGGRAPH 95*, pp.97-104, 1993.
- [11] Unuma, M. and et el, "Fourier principles for emotion-based human figure animation," *Computer Graphics (SIGGRAPH 95 Proceedings)*, pp.91-96, 1995.
- [12] Hodgins, J. K., W. L., Brogan, D. C., & OBrien, J. F., "Animating human athletics", *Computer Graphics (SIGGRAPH 95 Proceedings)*, pp.71-78, 1995.
- [13] Girard, M., Maciejewski, and A. A., "Computational modeling for the computer animation of legged figures", *Computer Graphics*, 19(3), pp.263-270, 1985.
- [14] Vukobratovic, M., "Biped locomotion", *Scientific Fundamentals of Robotics 7, Communications and Control Engineering Series*. Springer-Verlag, 1990.
- [15] Furusho, J., & Masubuchi, M., "Control of a dynamic biped locomotion system for steady walking", *Journal of Dynamic Systems, Measurement, and Control*, pp.108-118, 1986.
- [16] Pratt, D. and at el, "Insertion of an articulated human into a networked virtual environment", *Proceedings of the 1994 Conference on AI, Simulation, and Planning in High Autonomy Systems*, pp.84-90, 1994.
- [17] Ko, H., Cremer, J., "VRLOCO: Real-Time Human Locomotion from Positional Input Streams", *Presence*, vol.5, No.4, pp.367-380, 1996.
- [18] 김 승국, 표준 수화사전, 오성 출판사, 1993

저 자 소 개

金 鍾 成(正會員) 第 34卷 C編 第 2號 參照

1997년 2월 ~ 현재 한국 전자통신연구원 컴퓨터-소프트웨어 기술연구소 입체정보 연구팀 선임 연구원. 주관심분야는 재활공학(Rehabilitation Engineering 패턴 인식, Machine Vision, Intelligent System(Fuzzy, 인공 신경망), 가상현실 등임



閔 丙 義(正會員)

1982년 한양대학교 졸업(학사). 1984년 한국과학기술원 전기 및 전자공학과 졸업(석사). 1992년 한국과학기술원 전기 및 전자공학과 졸업(공학박사). 1984년 ~ 1987년 대림산업기술 연구소 1987년 ~ 현재

한국전자통신연구원 입체정보연구팀 팀장, 책임연구원. 주관심분야는 멀티미디어시스템, 에이전트, 가상현실 등 임



金 亨 培(正會員)

1971년 1월 21일생. 1996년 2월 연세대학교 전자공학과 학사. 1998년 2월 한국과학기술원 전기 및 전자공학과 석사. 1998년 ~ 현재 한국과학기술원 전기 및 전자공학과 박사 과정. 주관심분야는 Gesture

Recognition, Virtual Reality, Robotics, Image Processing 등임

卞 埜 男(正會員) 第 30卷 B編 第 10號 參照

현재 한국과학기술원 전기 및 전자공학과 교수

宋 慶 俊(正會員) 第 34卷 C編 第 9號 參照

1985년 1월 ~ 현재 한국 전자통신연구원 입체정보연구팀 선임연구원. 주관심분야는 가상현실, HCI, 멀티미디어 시스템 등임