

모션벡터를 이용한 가상현실 체험 시스템의 구현

서정만*, 정순기**

Implementation of Virtual Reality Immersion System using Motion Vectors

Jeong-Man Seo *, Soon-Key Jung **

요약

본 논문에서는 인간의 오감 중에서 시각을 이용하여 가상현실을 실제로 체험할 수 있는 가상현실 체험 시스템을 구현 하였다. 본 논문에서는 3 단계 검색(TSS : Three Step Search)방법을 이용, 현재 프레임에 대응되는 프레임을 인접블록에서 검색하여 두 프레임으로부터 모션벡터를 추출하였다. 구현된 시스템의 성능 평가를 위해 센서를 사용하여 측정된 시뮬레이터 축들의 가속도 값과 영상으로부터 추출한 모션벡터 값을 비교, 평가하였다. 본 논문에서 제시한 가상현실 체험 시스템이 영상의 움직임에 보다 가깝게 시뮬레이터를 동작시킬 수 있는 것으로 입증되었다.

Abstract

The purpose of this research is to develop a virtual reality system which enables to actually experience the virtual reality through the visual sense of human. TSS was applied in tracing the movement of moving picture in this research. By applying TSS, it was possible to calculate multiple motion vectors from moving picture, and then camera's motion parameters were obtained by utilizing the relationship between the motion vectors. For the purpose of experiencing the virtual reality by synchronizing the camera's accelerated velocity and the simulator's movements, the relationship between the value of camera's accelerated velocity and the simulator's movements was analyzed and its result was applied to the neural network training. It has been proved that the proposed virtual reality immersion system in this dissertation can dynamically control the movements of moving picture and can also operate the simulator quite similarly to the real movements of moving picture.

▶ Keywords : 모션벡터(Motion Vector), 가상현실(virtual reality), 블록정합(Block Match)

* 한국재활복지대학 컴퓨터게임개발과 교수

** 충북대학교 전기전자컴퓨터공학부 교수

1. 서론

가상현실은 여러 학문 분야의 지식을 광범위하게 활용하는 학제간의 연구 성격을 가지고 있으며(1), 인간과 컴퓨터간의 상호작용(HCI: Human Computer Interaction)을 실현하는 하나의 도구로서 인식되기도 한다. 국내의 각 연구기관과 대학에서도 수년 전부터 가상현실 기술에 대한 많은 연구가 활발히 진행되고 있으며, 과학기술부에서는 감성공학을 선도기술산업으로 지정하여 그간 많은 연구 성과를 도출하였다. 제조업, 건설업, 의료업 및 게임 소프트웨어 산업을 중심으로 산업계에서도 3차원 입체영상의 개발과 몰입형 자동차 시뮬레이터 개발 및 항공기, 탱크의 조종 시뮬레이터 개발 등을 통해 그간 축적된 가상현실 기술을 활발히 응용하고 있다.

본 논문에서는 인간의 시각을 통해 포착할 수 있는 동영상의 움직임에서 모션벡터를 추출하여 움직임의 종류를 분석하고, 이를 기반으로 영상의 움직임을 제어할 수 있는 명령어를 생성하여 시뮬레이터를 구동시킴으로써 영상의 움직임을 실제 체험할 수 있는 가상현실 체험 시스템을 제안하고자 한다. 기존 가상현실 체험 시스템에서는 동영상의 움직임 제어에 레더 다이어그램(LD: Ladder Diagram) 프로그램을 사용하므로 어플리케이션에 따라 매번 제어 프로그램을 갱신해야 한다. 이러한 단점을 보완하기 위하여 동영상으로부터 모션벡터를 추출하여 카메라 모션 파라미터의 추출에 이용하고, 카메라 모션 프레임을 이용하여 가상현실 체험 시뮬레이터를 구동시킴으로써 동영상의 움직임을 실제 상황에서와 동일하게 체험할 수 있도록 한다.

II. 모션벡터 검출 방법

3차원 공간에서 물체의 운동계는 시간에 따라 변하는 영상의 밝기 분포도만을 이용하여 계산할 수 없다. 특정 물체

가 물체의 배경영상과 다른 속도로 움직일 때 2차원 평면상에 투영된 영상의 밝기패턴의 이동속도를 분석하여 물체의 움직임을 계산한 결과를 모션벡터라고 부른다(2). 모션벡터를 이용하여 3차원상에서 물체의 이동속도를 구할 수 있다.

블록정합(block matching) 알고리즘(3)[4][5]은 동영상의 압축에도 일반적으로 사용되고 있으며, 동영상 압축기술은 HDTV (High Definition TV), 주문형 비디오, 영상의 등 다양한 응용분야에서 사용되고 있다. 최근에는 초저속(low bit rate) 동영상 전송을 위해 모션보정 비디오압축(motion compensated video compression) 기법에 대한 연구가 진행되고 있다. 특히, 영상의 움직임 추정 및 보정을 위해 연속된 프레임들 간에 시간적인 중복성을 제거시켜 영상 압축율을 높이고, 초저속 동영상 전송이 가능하도록 하고자 한다. 동영상의 움직임 추정 및 보정 기법으로는 수행시간이 비교적 적게 소요되고, 하드웨어의 구현이 용이한 블록정합 알고리즘(6)이 가장 많이 사용되고 있다.

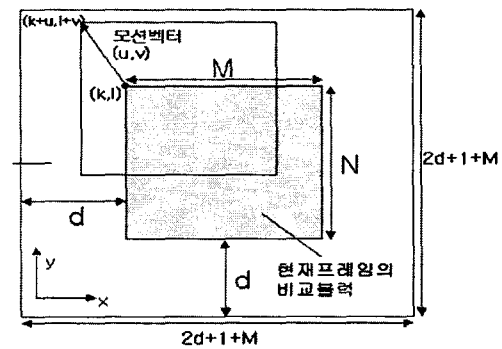


그림 1. 블록정합원리
Fig 1. Theory of Block Match

(그림 1)는 블록정합의 기본 원리를 나타낸다. 참조 프레임에 대해 일정한 크기($M \times N$)의 블록을 지정하고, 후속 프레임으로부터 참조 프레임의 블록에 대응되는 블록의 좌표($k+u, l+v$)를 구하여 두 프레임 간의 모션벡터(u, v)를 계산한다. 모션벡터의 최대 변위는 x 축과 y 축 모두에 대해 $\pm d$ 라고 가정한다면, 모션벡터(u, v)를 구하기 위해 $(2d+1+M) \times (2d+1+M)$ 크기의 참조 프레임 내에서 참조 프레임의 블록에 대응되는 후속 프레임의 블록 좌표 계산에 식(2-4)와 같이 두 블록내의 픽셀들에 대한 발광도(luminance) 값의 오차 $D(u, v)$ 를 이용한다.

III. 가상현실 체험 시스템 구현

1. 시스템 구조

기존 가상현실 시스템의 구성도를 살펴보면 영사기나 비디오에서 동영상을 재생하면 그것이 동영상 화면 스크린에 비추게 되고, 동영상에 맞게 PLC용 레더다이아그램을 동기화시켜 시뮬레이터를 구동한다. 동영상의 내용과 레더 프로그램이 동기화가 이루어지지 않으면 가상현실 체험이 효과적으로 수행되지 못한다. 또한 동영상의 내용이 바뀔 때 마다 레더 프로그램을 수정해야 하며, 그것에 따르는 고비용이 발생하는 단점이 있다. 그러므로 기존의 시스템에서는 20분 내외의 짧은 시간의 동영상만을 적용하고 있는 실정이다. 이와 같은 단점을 보완하기 위하여 소프트웨어 기반의 가상현실 체험 시스템을 제안한다.

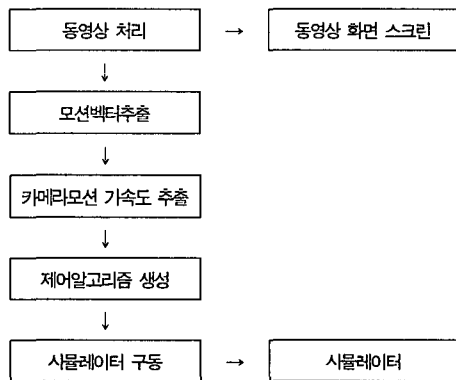


그림 2. 시스템 구성도
Fig 2. system configuration

〈그림 2〉은 본 논문에서 제안한 시스템의 구성도이다. 시스템에서 동영상은 디지털 형식으로 변환하여 컴퓨터에 저장되어 있다. 동영상을 컴퓨터 스크린에 재생하고 이를 프로젝터를 통하여 대형화면으로 투사한다. 동영상 처리에서는 디지털 형식으로 저장된 동영상으로부터 각 프레임별로 일련의 정지 영상을 추출하고 모션벡터 추출을 위한 영역설정 등의 초기화 작업을 수행한다. 모션벡터 추출에서는 TSS방법을 이용하여 모션벡터를 추출한다.

카메라 모션 가속도 추출에서는 4절에서 제안한 방법으로 동영상에 함축된 카메라의 모션 파라미터를 이용하여 가속도를 추출한다.

제어 알고리즘 생성에서는 추출된 카메라의 가속도를 이용하여 시뮬레이터 구동을 위한 제어 알고리즘을 생성한다. 시뮬레이터 구동은 제어 알고리즘의 명령어를 RS-232 통신으로 받아 가상현실 시뮬레이터를 구동한다. 본 논문에서 제안한 방법은 기존의 동영상 변경에 따른 레더 프로그램을 수정할 필요가 없고, 그에 따른 시간과 비용이 발생하지 않는 장점이 있다.

2. 카메라 모션의 가속도 추출

3개의 카메라 모션 파라미터 값을 계산하는 과정과 추출된 모션 파라미터 값과 영상의 모션벡터를 이용하여 카메라 모션의 가속도를 구하는 과정을 설명하면 다음과 같다.

〈그림 3〉에서와 같이 카메라의 볼록렌즈에 의해 촬영된 물체는 렌즈의 뒤쪽에 도립형태의 영상으로 나타난다. 렌즈로부터 물체까지의 거리가 a 이고, 렌즈의 초점 거리가 f 이며, 도립된 영상상이 형성되는 거리를 b 라 할 경우 변수 a, f, b 간의 관계는 식 (1)과 같이 표현되며, 렌즈의 배율은 식 (2)와 같이 표현된다.

$$\frac{1}{a} + \frac{1}{b} = \frac{1}{f} \dots\dots\dots (1)$$

$$m = \frac{a}{b} \dots\dots\dots (2)$$

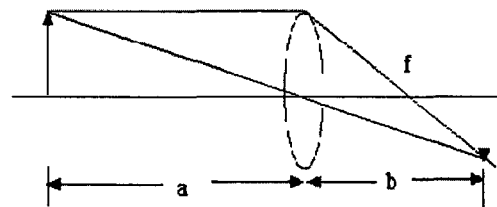


그림 3. 투영원리
Fig 3. Theory of projection

3차원 공간상의 물체를 촬영시 영상은 카메라에 2차원으로 투영된다. 〈그림 4〉은 COP(Center of Projection)를 기준으로 대상 물체의 특정한 지점에 해당하는 좌표 $r(x, y, z)$ 이 카메라 렌즈의 평면 좌표 $p(u, v)$ 에 투영되는 형태를 벡터로 표현한 것이다.

COP를 중심으로 좌표 $r(x, y, z)$ 까지의 벡터를 \vec{r} 이라 하고, 이에 대응되는 수직 단위 벡터를 i, j, k 로 표현하면 \vec{r} 은 식 (3)과 같다.

$$\vec{r} = xi + yj + zk \dots\dots\dots (3)$$

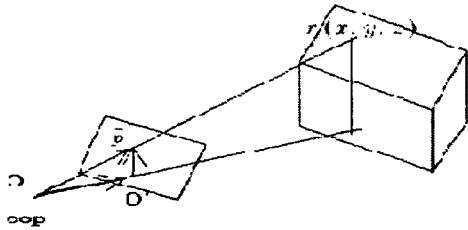


그림 4. 투영형태
Fig 4. Type of Projection

좌표 $r(x, y, z)$ 이 평면에 투영되는 좌표 $p(u, v)$ 를 이미지 평면상의 중심점 o' 에서 벡터로 표현하면 식 (4)와 같다.

$$\vec{o'p} = ui + vj \dots\dots\dots (4)$$

조건으로, $\vec{oo'}$ = bk 인 경우 벡터 \vec{op} 와 \vec{r} 의 관계를 표현하면 식 (5)와 같다.

$$\vec{op} = \frac{\vec{r}}{z} | \vec{oo'} | = b \frac{\vec{r}}{z} = \vec{oo'} + \vec{o'p} \dots\dots\dots (5)$$

그러므로 이미지 평면상의 벡터 $\vec{o'p}$ 는 벡터 \vec{op} 와 벡터 $\vec{oo'}$ 의 차이 값에 해당되므로 식 (6)이 성립된다.

$$\vec{o'p} = \vec{op} - \vec{oo'} = b \frac{\vec{r}}{z} - bk \dots\dots\dots (6)$$

식 (6)에서 비례식을 이용하여 식 (7)을 구하고,

$$b : z = | \vec{op} | : | \vec{r} |$$

$$| \vec{op} | = \frac{b}{z} | \vec{r} | \dots\dots\dots (7)$$

식 (3)의 양변을 z 로 나누고, b 를 곱하면 식 (8)을 구할 수 있다.

$$\frac{b}{z} \vec{r} = \frac{b}{z} xi + \frac{b}{z} yj + \frac{b}{z} zk$$

$$= \frac{b}{z} xi + \frac{b}{z} yj + bk \dots\dots\dots (8)$$

식 (8)을 식 (6)에 대입하면 식 (9)과 같은 관계식을 구할 수 있다.

$$\vec{o'p} = \frac{b}{z} xi + \frac{b}{z} yj \dots\dots\dots (9)$$

한편, $\vec{o'p} = ui + vj$ 이므로 식 (9)으로부터 이미지 평면의 좌표 $p(u, v)$ 와 공간 좌표 $r(x, y, z)$ 의 관계식 식 (10)과 같이 얻을 수 있다

$$u = \frac{b}{z} x, v = \frac{b}{z} y \dots\dots\dots (10)$$

시뮬레이터를 구동시키기 위하여 동영상에 포함된 카메라의 이동방향을 역추적 한다. 따라서 이미지 평면에서 모션벡터의 변화로부터 카메라의 움직임을 계산해야 한다. 이를 위해 공간상의 속도벡터 $\frac{d\vec{r}}{dt}$ 와 이미지 평면상의 모션벡터 변화를 $\frac{du}{dt}, \frac{dv}{dt}$ 와의 관계식을 유도해야 한다. 벡터 \vec{r} 의 변화율 $\dot{\vec{r}}$ 와 벡터 $\vec{o'p}$ 의 변화율($\dot{\vec{o'p}}$)은 각각 식 (11), (12)과 같다.

$$\frac{d\vec{r}}{dt} = \dot{\vec{r}} = \dot{x}i + \dot{y}j + \dot{z}k \dots\dots\dots (11)$$

$$\frac{d}{dt} (\vec{o'p}) = \frac{d}{dt} (\frac{b}{z} \dot{\vec{r}} - bk)$$

$$\dot{\vec{o'p}} = \frac{d}{dt} (\frac{b}{z} xi + \frac{b}{z} yj)$$

$$\dot{\vec{o'p}} = (b \frac{\dot{x}}{z} - \frac{bx\dot{z}}{z^2})i + (\frac{b\dot{y}}{z} - \frac{by\dot{z}}{z^2})j$$

$$\dot{\vec{o'p}} = \dot{u}i + \dot{v}j \dots\dots\dots (12)$$

따라서 식 (12)으로부터 다음과 같은 식 (13), (14)를 유도할 수 있다.

식에서 $\frac{b}{z}\dot{x}$ 는 수평으로 이동하는 성분이며, $\frac{bxz}{z^2}$ 는 이동속도의 성분을 나타낸다.

$$\frac{du}{dt} = \dot{u} = b\frac{\dot{x}}{z} - \frac{bxz}{z^2} \dots\dots\dots (13)$$

$$\frac{dv}{dt} = \dot{v} = \frac{by}{z} - \frac{byz}{z^2} \dots\dots\dots (14)$$

식 (13), (14)에서 $\dot{x} = 0, \dot{y} = 0$ 인 경우 식 (15), (16)를 유도할 수 있다.

$$\dot{u} = -\frac{bxz}{z^2} \dots\dots\dots (15)$$

$$\dot{v} = -\frac{byz}{z^2} \dots\dots\dots (16)$$

따라서 $\dot{z} > 0$ 인 경우는 물체가 카메라로부터 멀어지거나, 카메라가 물체로부터 멀어지는 경우로 모션벡터는 식 (15), (16)으로부터 구할 수 있다.

만일 $\dot{z} = 0, \dot{y} = 0, \dot{x} > 0$ 인 경우 식 (13), (14)에서 $\dot{u} = b\frac{\dot{x}}{z}, \dot{v} = 0$ 이 된다.

즉, 카메라가 x 축으로 이동하는 경우 모션벡터의 방향은 <그림 5>와 같다. 이미지 평면을 일정 블록으로 나누어 각 구역마다 모션벡터를 구하고, 이들의 평균 값 \bar{u}, \bar{v} 를 계산한다. 식 (13), (14)을 이용하여 평균 모션벡터의 성분을 계산하면 식 (17), (18)과 같다.

$$\bar{u} = \sum_{i=1}^N u_i = \frac{1}{N} \sum_{i=1}^N \left(\frac{b}{z_i} \dot{x}_i - \frac{bx_i z_i}{z_i^2} \right) \dots\dots\dots (17)$$

$$\bar{v} = \sum_{i=1}^N v_i = \frac{1}{N} \sum_{i=1}^N \left(\frac{b}{z_i} \dot{y}_i - \frac{by_i z_i}{z_i^2} \right) \dots\dots\dots (18)$$

식 (17), (18)에서와 같이 모션벡터의 평균속도는 카메라의 x 축 또는 y 축에 대한 이동성분과 z 축에 대한 이동성분의 조합으로 이루어진다.

이미지 평면상의 \bar{u} 와 \bar{v} 는 이미지 평면상의 여러 블록으로부터 각각의 모션벡터를 추출하여 이들의 평균값을 계산하여 구할 수 있다. 이미지 평면상의 블록 중심점을 기준으로 대칭시켜 대칭되는 블록들 중에서 작은 블록을 선택하면 복수개의 블록이 이미지 평면상에서 동일한 피사체의 영상이 되므로 $z_i = z_0 = const$ 가 성립된다.

식 (17), (18)에서 작은 블록을 선택한다고 가정을 하였으므로,

$$\left. \begin{matrix} \dot{x}_i = \dot{x} \\ \dot{y}_i = \dot{y} \\ \dot{z}_i = \dot{z} \end{matrix} \right\} \dots\dots\dots (19)$$

따라서 (13)으로부터 식 (20)을 구할 수 있다.

$$\left. \begin{matrix} \dot{u}_1 = \frac{b}{z_0} \dot{x} - \frac{b}{z_0^2} x_0 \dot{z} & \dot{u}_2 = \frac{b}{z_0} \dot{x} \\ \dot{u}_3 = \frac{b}{z_0} \dot{x} + \frac{b}{z_0^2} x_0 \dot{z} \\ \dot{u}_4 = \frac{b}{z_0} \dot{x} + \frac{b}{z_0^2} x_0 \dot{z} \\ \dot{u}_5 = \frac{b}{z_0} \dot{x} + \frac{b}{z_0^2} x_0 \dot{z} \\ \dot{u}_6 = \frac{b}{z_0} \dot{x} & \dot{u}_7 = \frac{b}{z_0} \dot{x} - \frac{b}{z_0^2} x_0 \dot{z} \\ \dot{u}_8 = \frac{b}{z_0} \dot{x} - \frac{b}{z_0^2} x_0 \dot{z} & \dot{u}_9 = \frac{b}{z_0} \dot{x} \end{matrix} \right\} \dots\dots\dots (20)$$

식 (20)을 이용하여 식 (21), (22)과 같은 카메라 모션 파라미터를 추출한다.

$$\frac{\dot{u}_1 + \dot{u}_2 + \dots + \dot{u}_9}{9} \frac{b}{z_0} \dot{x} = \frac{1}{m} \dot{x} \dots\dots\dots (21)$$

$$\left. \begin{aligned} & \frac{(u_3 + u_4 + u_5) - (u_1 + u_7 + u_8)}{6} \\ & \frac{b}{z_0^2} x_0 \dot{z} = \frac{1}{m} \frac{x_0}{z_0} \dot{z} \\ & = \frac{1}{m} \tan\left(\frac{FOV_x}{2}\right) \dot{z} \end{aligned} \right\} (22)$$

즉, 이미지 블록의 모션벡터 $\{(u_1, v_1) \dots (u_9, v_9)\}$ 를 구하고 식 (21), (22)를 이용하여 카메라의 모션 파라미터 x, y, z 를 구한다(7).

카메라 모션 파라미터 x, y, z 를 이용, 카메라 모션의 가속도를 구하기 위해 프레임에 따라서 시간대 별로 속도의 변화율을 계산한다. 실험에서는 시뮬레이터에 부착된 센서를 통해 측정된 가속도 값과 카메라 모션의 가속도 값을 비교하여 카메라 모션 파라미터의 값이 정확히 추출되었는지를 확인한다.

IV. 실험 및 결과 분석

실험환경으로는 운영체제 윈도우 2000, 메모리 383 메가바이트, 사용언어는 비주얼 C++, CPU는 펜티엄 4, 컴퓨터와 시뮬레이터와의 통신 방법은 RS232, 시뮬레이터는 Cobra Chair로 실험을 하였다. 가속도 센서는 3개의 축을 측정할 수 있는 "3-AXIS ACCELEROMETER"인 LabPro를 사용하였다. 실험을 위하여 시뮬레이터(6)의 3개 축에 LabPro 가속도 센서를 부착하였고, 운동량을 분석하여 가속도 벡터를 추출한다. 합성이미지를 사용하여 모션 벡터를 추출하고, 가속도 값을 측정한다. 이 실험에서는 x, y, z축의 값을 측정하여 나타내었다.

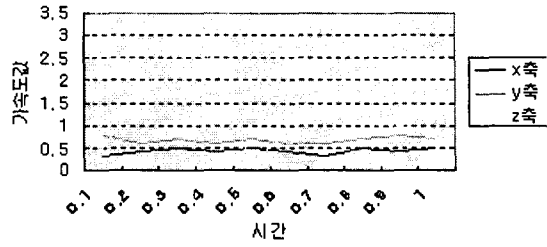


그림 5. 센서 그래프
Fig 5. Graph of sensor

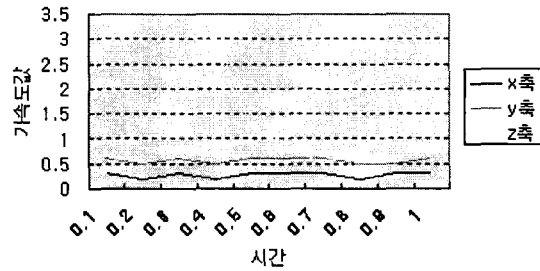


그림 6. 실험 그래프
Fig 6. Graph of Simulation

실제적으로 실험을 위해 촬영한 실제(real image) 동영상 상을 가지고 모션벡터를 추출한 후, 가속도 값을 측정한다. 화면의 동영상 파일의 가속도 값 중에서 앞뒤이동에 대한 가속도 값을 측정하였다. 먼저 동영상 촬영 시 카메라에 가속도 센서를 부착하여 얻은 가속도 값과 프로그램을 이용하여 계산된 동영상으로부터 측정된 가속도 값을 상호 비교 분석하여 시스템의 성능평가를 하였다.

위의 그래프를 통하여 알 수 있듯이 앞뒤이동에 대한 그래프에서 센서로 측정된 값과 프로그램 실험에 의한 값 차이가 미세함을 알 수 있다.

V. 결론

본 논문에서는 동영상으로부터 모션벡터를 추출하여 실제 영상의 움직임과 동일하게 시뮬레이터를 구동시킬 수 있는 가상현실 체험 시스템을 제안하였다. 동영상으로부터 모션벡터의 추출에 이용되고 있는 여러 종류의 알고리즘들을

조사, 분석하였으며, 그 중에서도 TSS 알고리즘을 선택하여 동영상에서 모션벡터의 추출에 이용하였다. 동영상으로부터 복수개의 모션벡터를 추출하여 영상 움직임의 종류를 분석하였으며, 이들 간의 관계를 분석하여 카메라 모션 파라미터를 도출하였다. 카메라 모션 파라미터 값의 정확성 검증을 위해 카메라에 부착된 센서를 통해 측정된 카메라 모션 파라미터의 가속도 값과 동영상으로부터 추출한 모션 벡터 값의 패턴을 비교, 분석하였다. 카메라 모션의 가속도 값과 시뮬레이터에 장착된 의자의 현재 위치 값을 뉴럴 네트워크(BPN)에 입력하여 의자의 다음 위치정보를 계산하여 시뮬레이터의 동작을 제어토록 하였다.

측정된 가속도 값들을 각각 비교한 결과 시간적인 차이를 나타내고 있지만, 측정된 값들의 패턴은 동일한 것으로 분석되었으므로 시뮬레이터가 영상의 실제 움직임에 가깝게 작동하는 것으로 입증되었다.

본 논문에서 제안된 시스템에 대한 향후 연구과제로는, 시스템의 효율적인 운영을 위해서 디지털 신호를 실시간으로 처리할 수 있는 하드웨어 구조와 영상 제어 알고리즘 구조에 대한 연구가 필요하다.

[7] 서정만, "모션벡터를 이용한 가상현실 체험 시스템의 구현", 충북대학교 대학원, 박사학위논문, pp.35-40, 2003.

참고문헌

[1] 나필화, "가상현실 환경 구현을 위한 데이터베이스를 이용한 통합 저작도구 개발", 충남대학교 대학원, 석사학위논문, pp.1-2, 2001.

[2] A. Singh, "Optical Flow Computation", IEEE Computer Society Press, 1991.

[3] 안교주, "비디오 압축을 위한 고속 블록 정합 알고리즘에 관한 연구", 세종대학교 대학원, 석사학위논문, pp.13, 2000.

[4] 정신영, "다중물체가 움직이는 동화상에서 자동적인 특징점 추출 및 추적", 성균관대학교 대학원, 박사학위논문, pp.8-13, 1996.

[5] 성윤주, "H.263에서 움직임 정합 블록을 찾기 위한 개선된 3단계 탐색 알고리즘", 경상대학교 대학원, 석사학위논문, pp.13, 2001.

[6] 가상현실용 3차원 시뮬레이터, 특허, 출원번호 20-1998-0006935, 1999.

저자소개



서 정 만
 2003. 2 충북대학교
 컴퓨터공학과 공학박사
 2002 ~ 현재
 한국재활복지 대학 컴퓨터
 게임개발과 교수



정 순 기
 1994
 Rijksuniversiteit Groningen,
 Computing Science Ph.D
 1986 ~ 현재 충북대학교 컴퓨터
 공학과 교수