

힘 반향을 이용한 속도타원 가상환경 네비게이션 알고리즘 개발

윤인복*, 채영호**

Development of a Velocity Ellipse Navigation Algorithm in Virtual Environments Using Force Feedback

Yoon, I.B.* and Chai, Y.H.**

ABSTRACT

In this paper, a 2 DOF haptic yawing joystick for use as the navigation input device in virtual environments is introduced. The haptic yawing joystick has 360° range for yawing motion and ±100° for pitching motion. The device can support weights of up to 26N for x axis and 10N for y axis with 10KHz of sampling rate. The size of the haptic yawing joystick is so small that it can be assembled on armrest of an arm chair and has relatively larger work space than other conventional 2 DOF joysticks. For the haptic yawing joystick, an ellipse navigation algorithm using the user's velocity in the virtual navigation is proposed. The ellipse represents the velocity of the user. According to the velocity of the navigator, the ellipse size is supposed to be changed. Since the path width of navigation environments is limited, the ellipse size is also limited. The ellipse navigation algorithm is tested in 2 dimensional virtual environments. The test results show that the average velocity of the navigation with the algorithm is faster than the average navigation velocity without the algorithm.

Key words : Haptic Joystick, Navigation, Ellipse Algorithm, Virtual Environments

1. 서 론

햅틱 인터페이스(haptic interface)는 촉감(tactile feedback)이나 역감(force feedback)의 재현을 통하여 가상환경 속에서 자연스럽게 물체를 조작할 수 있게 하는 장치로써 네비게이션(navigation)용 조이스틱의 형태로 구현되었을 경우 시각환경 재현 알고리즘과 이상적으로 결합이 가능하다.

손으로 사용하고 조작하는 햅틱 장치(haptic device)는 손 기반(hand-based)장치와 손가락 기반(finger-based) 또는 펜 기반(pen-based) 장치로 구분된다^[1]. 손가락 기반 또는 펜 기반 장치는 PHANTOMTM^[2]이 대표적이고, 6자유도로서 원활한 움직임을 주지만 고가이다.

손 기반 장치는 동력원을 기준으로 자기부양

(magnetic levitation)이나 모터를 이용한 형태로 구분할 수 있고, 모터 구동형태는 구조상 직교(cartesian)형과 스투워드(stewart)형으로 구분이 가능하다^[3]. 자기부양 형태^[4]는 전체 크기는 작지만, 작업영역이 제한되기에 큰 움직임이 필요한 네비게이션에는 적용이 어렵다. 모터 구동형태^[5-7]는 구조 변경이 용이하고 작업영역이 상대적으로 커서 대부분의 네비게이션용 햅틱 장치에 사용된다.

또한 가상환경에서 충돌 검사는 객체를 감싸는 경계구(bounding sphere) 또는 경계상자(bounding box)를 이용하고 있으나 움직임에 대한 속도감을 적용하지 못한다^[8]. 본 논문에서는 Fig. 1과 같이 네비게이션 입력장치로서 의자에 장착 가능한 모터구동형의 햅틱 조이스틱을 개발하고, 가상환경을 탐험하는 사용자의 속도에 반응하는 속도 타원 네비게이션 알고리즘을 제안한다.

개발된 장치와 알고리즘은 향후 다양한 인터랙티브 라이터의 역감 재현 인터페이스 개발의 기술적기반으로 사용할 수 있다.

*중앙대학교 첨단영상대학원 영상공학과
**중신회원, 중앙대학교 첨단영상대학원 영상공학과
- 논문투고일: 2003. 08. 11
- 심사완료일: 2004. 05. 14

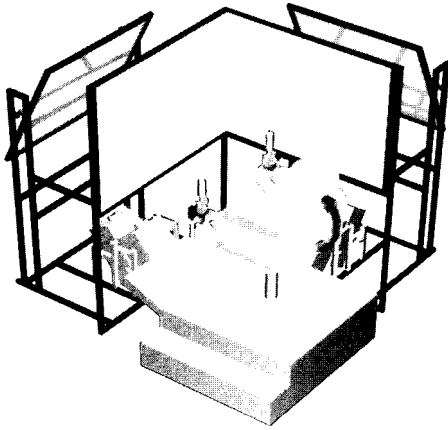


Fig. 1. The overall view of the system.

2. 2자유도 햅틱 yawing 조이스틱

본 논문에서는 시각환경에 따라 제어되는 모션라이더(motion rider) 또는 모션 의자(motion chair)와 함께 연동하여 사용자에게 높은 몰입감을 주면서도 전체 시스템의 공간 이용효율을 높이기 위해 Fig. 1에서 보는 바와 같이 의자의 팔걸이 부분에 장착 가능한 Fig. 2의 2자유도 햅틱 yawing 조이스틱을 소개한다.

이 장치는 손잡이 내부에 모터가 장착되어 요잉운동(yawing motion)으로 공간에서의 좌우운동을 대신하였고 피칭운동(pitching motion)으로 공간의 상하운동을 연결할 수 있도록 설계하였으며, 전체크기는 6*5*20[cm]이면서 360°의 yawing 모션과 ±100°의 pitching모션이 가능한 작업영역을 갖는다.

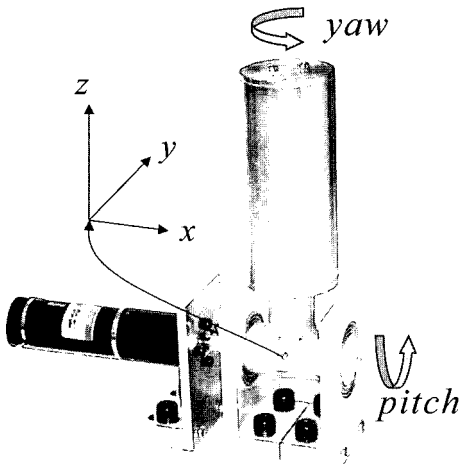


Fig. 2. The 2 D.O.F. haptic yawing joystick.

모터는 각각 Maxon motor사의 DC 모터를 사용하였으며 토크 상수는 x 축, z 축에 각각 43.3, 20 [mNm/A]으로 토크 증가를 위해 81:1과 67:1의 기어를 사용하였다. 따라서 모터부분에서의 최종 토크는 x 축과 z 축 각각 3.51[Nm/A]과 1.34[Nm/A]이다. 또한 네비게이션시 사용자의 움직임을 추적하기 위해 각각 0.36°/count와 1.41°/count의 회전 분해능을 가지는 인코더를 사용하였다. 이것은 기어와 연결되어 최종 분해능이 각각 0.044°/count와 0.021°/count가 된다. 이 분해능은 JND(Just Noticeable Difference)의 성능 값¹³⁾인 사람의 손가락 끝의 분해능 약 2.5°, 사람 팔꿈치나 손목의 분해능 2.0°를 충분히 추적할 수 있는 값이다.

전체 시스템 구성도는 Fig. 3과 같다. Fig. 3에서 사용자(Human operator)는 햅틱 조이스틱을 조작하여 가상환경을 네비게이션 한다. 이 움직임에 대한 위치 정보가 햅틱 조이스틱의 모터와 일체형인 인코더로부터 인코더 보드를 통해 실시간으로 제어기에 전달되고 제어기는 가상환경의 위치정보와 비교 및 분석을 통하여 사용자의 힘 반향(force reflection)을 위해 실시간으로 햅틱 조이스틱에 힘신호를 주도록 구성되어 있다.

시스템의 성능을 시험하기 위해 가상의 벽(virtual wall) 실험을 하였다. Fig. 4는 실험을 하기 위한 가상환경을 나타낸다. 사용자가 손잡이를 회전하여 $-x$ 방향으로 움직여서 햅틱 조이스틱의 왼편에 위치한 가상의 벽에 부딪히게 되고 손잡이를 아래로 당김으로서 $-y$ 방향으로 움직이게 되어 햅틱 조이스틱의 아래편에 위치한 가상의 벽에 접촉하게 된다. 벽과 접촉시 반력은 식 (1) F_x 와 식 (2) F_y 에 의해 사용자에게 전달된다.

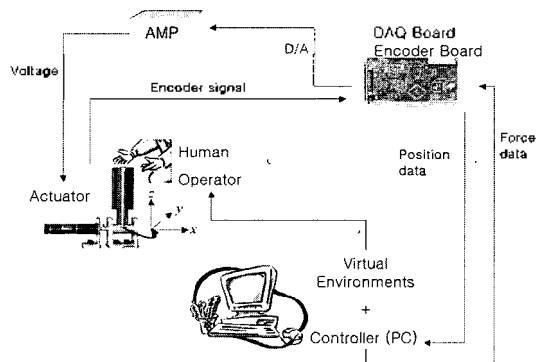


Fig. 3. Schematic diagram of the system.

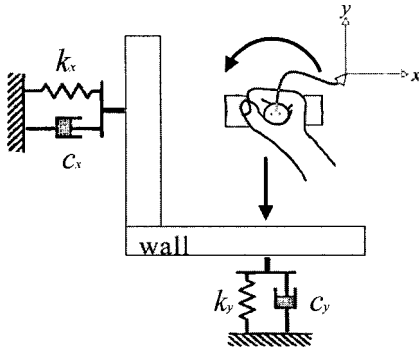


Fig. 4. Virtual wall test.

$$F_x = c_x(\Delta\dot{x}) + k_x\Delta x \quad (1)$$

$$F_y = c_y(\Delta\dot{y}) + k_y\Delta y \quad (2)$$

개발된 2자유도 햅틱 yawing 조이스틱은 비교적 넓은 작업영역을 가지고, 높은 분해능으로 사용자의 움직임에 정밀하게 반응할 수 있어 가상환경 네비게이션의 입력장치로 사용될 경우 보다 현실감 넘치는 상호작용을 가능하게 한다.

3. 속도타원 가상 네비게이션 알고리즘

3.1 알고리즘의 정의

인터랙티브 콘텐츠의 진경생성을 위한 시점변환이나 애니메이션 제작 과정에서 경로생성은 모두 3차원 가상공간의 탐색 과정을 필요로 한다. 이를 위한 공간상의 물체와 충돌 테스트 및 부드러운 움직임을 위한 지능적 네비게이션 알고리즘의 개발은 가상환경의 현실감 증대에 중요한 요소이며 일정한 규칙을 발견할 경우 모든 가상현실 시스템에 핵심이 되는 필수 요소로 적용될 수 있다. 본 장에서는 공간상의 물체위치를 검색하여 속도를 조절하고 자연스러운 주행이 될 수 있도록 하여 몰입감을 증가시키는 네비게이션 알고리즘을 제안한다.

3.1.1 기존의 힘 반향 방식

가상환경의 네비게이션에서 사용자가 이동 중 장애물들의 위치를 좀더 빨리 알 수 있으면 쉽게 장애물과의 충돌을 피하면서 속도감을 향상시킬 수 있게 될 것이다. 이러한 속도감 향상은 사용자의 몰입감과 직결된 요소로 작용하므로 햅틱요소를 이용하여 빠르고 간결한 네비게이션을 생각해 볼 수 있다.

일반적으로 많이 이루어지는 힘 피드백(force feedback) 방식은 Mark¹⁶⁾의 면과 탐침(plane and probe) 방식과 같이 충돌 시 힘의 크기는 같고 방향이 반대인 형식이였다. 그러나 이러한 힘 반향 방식이 네비게이션에 적용될 경우, 힘의 작용방향이 진행방향과 반대인 이유로 진행을 방해하게 된다. 이것은 네비게이션의 어려움으로 작용하는 경우가 많아 오히려 속도감을 떨어뜨리게 된다.

Fig. 5에서 가상환경의 장애물 객체인 VO(virtual object)를 감싸는 경계원과 HP(haptic point)를 감싸는 경계 원이 네비게이션 중 충돌했을 시 충돌에 대한 힘 반향의 방향은 각 객체 경계원의 접선 벡터 T 에 대한 법선 벡터 N 의 방향과 같다. 면과 탐침¹⁶⁾의 방식도 역시 탐침(probe)이 부딪힌 면(plane) 방향의 수직인 방향으로 힘이 작용한다. 여기서 탐침의 속도는 고려되지 않았다.

실제 네비게이션에서 사용자는 Fig. 6에서 보는 바와 같이 시각적으로 장애물임을 확인하는 순간 피해 가려는 일련의 동작들을 취하게 된다.

이러한 과정에서 장애물을 인지한 사용자는 속도를 줄이게 되고 장애물의 수가 많을수록 속도 저하가 빈번하게 발생하여 자연스러운 네비게이션이 불가능하다. 따라서 이러한 속도 저하를 힘반향을 통해 개선할 수 있는 능동형 속도 타원 알고리즘이 필요하다.

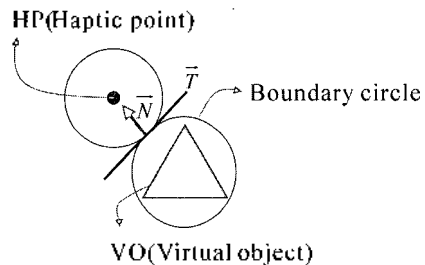


Fig. 5. Conventional force reflection¹⁶⁾.

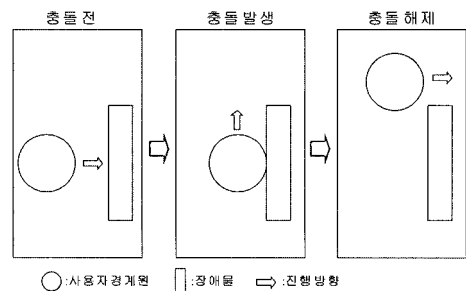
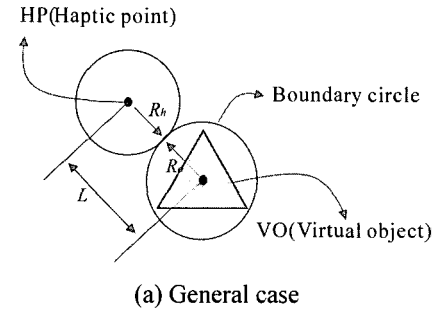
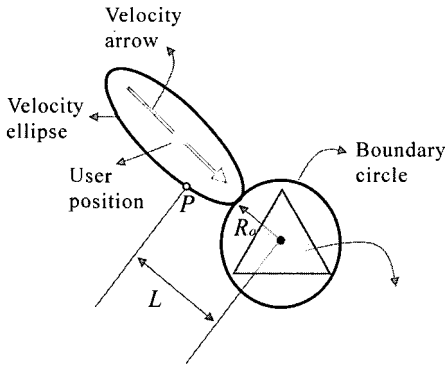


Fig. 6. Collision process.



(a) General case



(b) Velocity ellipse case

Fig. 7. Collision detection.

3.1.2 능동형 속도 타원 알고리즘

본 연구에서 제안하는 능동형 속도 타원 알고리즘은 다음과 같은 세 가지 절차에 의해 실행된다.

① 충돌 검사

먼저 사용자 객체가 장애물과 접촉되었는지 여부를 알아야 한다. Fig. 7에서 일반적인 경우의 충돌 검사와 타원의 경우의 충돌 검사 방법을 보이고 있다. 일반적인 경우는 Fig. 7(a)와 같이 햅틱 포인터인 사용자의 경계원 반지름 R_h 와 장애물 객체 반지름 R_o 와의 거리를 비교해서 $(R_h + R_o) < L$ 이면 충돌하지 않은 것이고 $(R_h + R_o) \geq L$ 이면 충돌을 일으킨 것으로 판단한다.

그러나 충돌을 감지하기 위한 경계를 타원으로 하면 경계원처럼 반지름이 일정하지 않으므로 Fig. 7(b)에서와 같이 경계타원 위의 점 P로부터 장애물 객체 중심까지의 거리 L 과 장애물의 반지름 R_o 간의 거리를 직접 비교하여야 한다.

② 사용자 움직임의 실시간 판단

일반적인 비메케이션에서 충돌시 반향 되는 힘의 방향은 사용자 객체의 경계원이 장애물과 충돌할 때

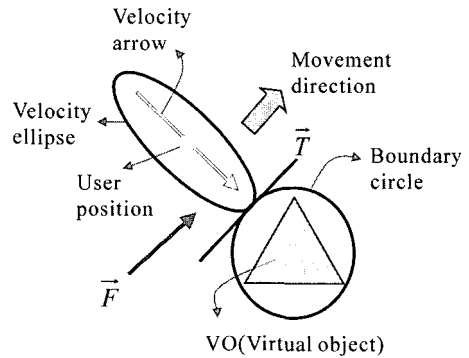


Fig. 8. Velocity ellipse algorithm.

법선 방향으로 힘을 전달하였기 때문에, 장애물을 피하기 위한 사용자의 회전움직임을 판별할 필요가 없었다.

그러나 Fig. 8에서 보는 것처럼 사용자의 속도에 반응하는 경계타원이 장애물과 접촉시, 사용자가 원하는 방향으로 이동하기 쉽도록, 반향되는 힘의 작용 방향을 사용자가 이동하는 방향과 같은 방향으로 준다면 회전시 발생하는 속도저하를 보상할 수 있을 것이다. 따라서 사용자의 회전 방향을 능동적으로 판단하는 것이 필요하게 된다.

사용자의 움직임 방향의 능동적 판별은 사용자의 움직임과 동시에 발생하는 인코더의 출력 값으로부터 아래와 같은 간단한 변화율 식에 의해 가능하다.

$$\vec{P}_{change} = \frac{\vec{P}_{front} - \vec{P}_{back}}{P_{back}} \quad (3)$$

식 (3)에서 벡터 P_{change} 는 사용자의 이동 변화율이고 벡터 P_{front} 는 시간 t 에서 사용자의 위치벡터이며 벡터 P_{back} 는 시간 $t-1$ 에서 사용자의 위치벡터이다. 식 (3)의 이동 변화율 벡터 P_{change} 로부터 Fig. 9에서 보는 바와 같이 사용자의 경계타원이 장애물과 접촉하게 되었을 때 $P_{change} > 0$ 이면 우측으로 회전이고 $P_{change} < 0$ 이면 좌측으로 회전하는 것이며 $P_{change} = 0$ 이면 회전하지 않고 움직임이 없다는 것으로 실시간 판별이 가능하다.

③ 반향되는 힘의 방향

Fig. 8에서 보는 바와 같이 경계타원이 장애물과 접촉하면서 회전하게 될 때 그 회전 방향을 실시간으로 판단하여 회전 방향으로 힘 반향을 주게 되면, 회전시 장애물과 충돌을 피하기 위해 나타나는 속도저하를

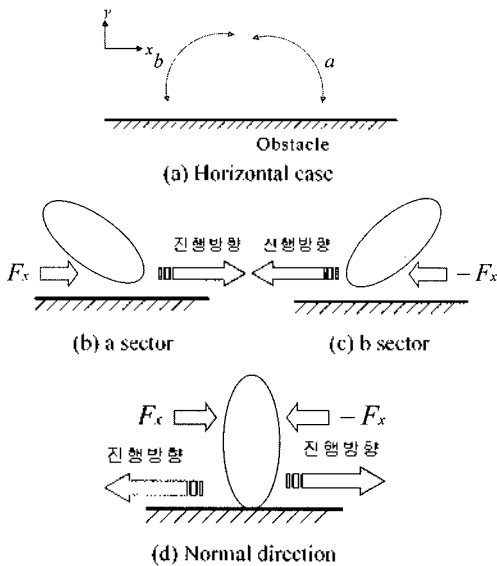


Fig. 9. Force reflection for horizontal obstacles.

보상할 수 있게 된다. 그러나 충돌 검출을 위한 경계가 타원이기 때문에 타원의 특성상 힘 반향의 작용 방향을 좀더 구분하여 줄 필요가 있다.

제안된 속도타원 알고리즘에서 중요한 것은 사용자의 경계타원과 장애물의 경계가 접촉하였을 경우 제어기가 사용자의 진행방향을 판단하는 것이다. 그렇게 판단된 진행방향에 따라 힘 반향을 준다.

Fig. 9, 10, 11에서 반향 되는 힘의 작용 방향을 구하기 위한 장애물의 기울어진 형태와 사용자의 진행 방향에 따른 구분을 보이고 있다. 경계 타원이 장애물의 경계와 접촉하게 되면 두 모터에 의해 두 방향으로 힘반향을 생성하기 때문에 각 그림에서 장애물의 법선 방향 사용자의 진행방향을 기준으로 각각 두 개씩의 구간으로 구분하였다. 그러나 장애물의 법선 방향으로 접촉하는 사용자의 움직임 화살표는 식 (3)로 회전방향을 판단하여 작용되는 힘의 방향을 정해주었다.

3.2 네비게이션 실험

실험은 자유공간에서의 자연스러운 움직임을 위해 저대역 통과 필터(low-pass filter)를 사용한 PD 제어를 이용하여 위치제어를 하였다. 제어 시스템 블록 다이어그램은 Fig. 12과 같다. K_p 는 비례이득 값이고 K_d 는 미분이득 값이다. 가상환경내 임의 장애물의 위치정보 P_g (position of geometry)와 햅틱 조이스틱의 움직임으로 구할 수 있는 사용자의 위치정보 P_h (position of human operator)의 관계로부터 가상환경

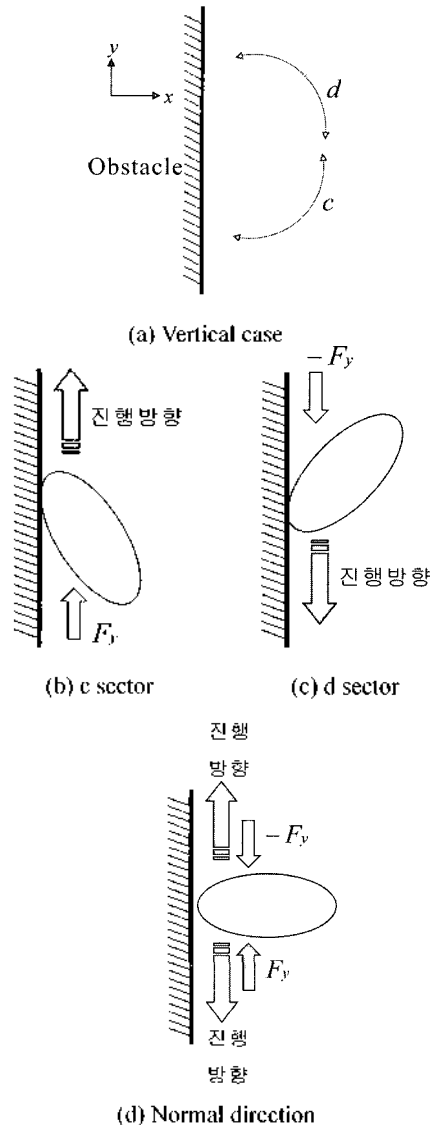


Fig. 10. Force reflection for vertical obstacles.

내 힘반향 정보를 식 (4), (5)을 이용해 구할 수 있다.

$$\vec{P}_{des} = \vec{P}_g - \vec{P}_h \tag{4}$$

$$\vec{F}_{hf} = k_p \vec{P}_{des} + k_d \cdot (L.P Filter) \cdot \vec{P}_{des} \tag{5}$$

식 (4)에서 벡터 P_{des} 는 벽과 충돌할 때의 사용자의 위치정보를 나타내고 식 (5)에서 벡터 F_{hf} 는 햅틱 제원의 힘(haptic feedback force)으로서 장애물과 충돌 시 적용된다.

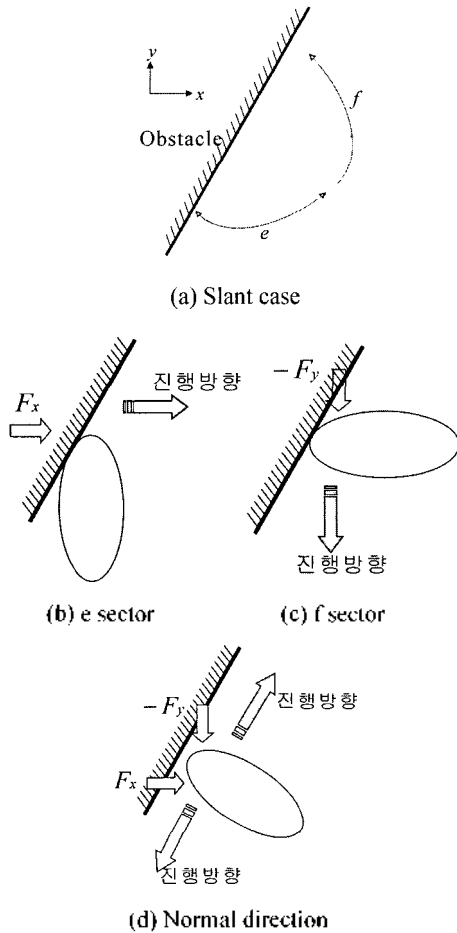


Fig. 11. Force reflection for slant obstacles.

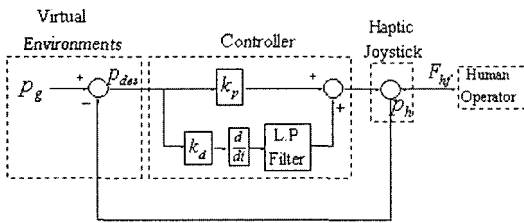


Fig. 12. Control block diagram.

3.2.1 경계 원과 경계 타원의 비교

Fig. 13에서 장애물을 검색하는데 걸리는 시간을 측정하기 위한 실험방법을 보인다. 실제 네비게이션에서 사용자가 장애물과 부딪히는 것은 화살표방향이기 때문에 실험은 정측방향으로 하였다. 또한 출발선을 동일하게 한 다음 경계원이 속도에 따라 장애물을 검색하는 시간과 경계 타원이 속도에 따라 장애물을 검색하는 시간을 측정하였다. 경계 타원은 속도에 따라

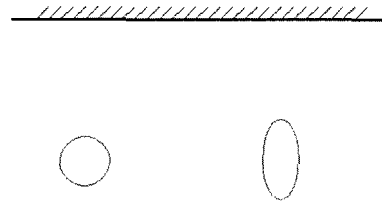


Fig. 13. Test of detecting an obstacle.

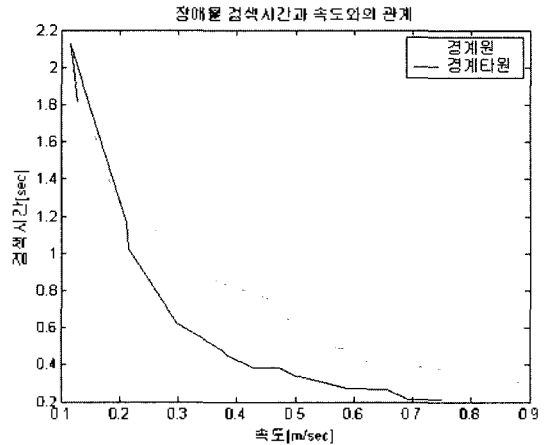


Fig. 14. Result of the test.

비례하여 커지므로 속도가 빠르면 빠를수록 경계 타원의 장애물 검색 시간이 경계원보다 짧을 것이다. 그 결과를 Fig. 14에서 보인다.

Fig. 14에서 보는 바와 같이 저속(약 0.2 m/sec)에서는 경계 타원과 경계 원의 장애물검색 시간이 큰 차이가 없으나 속도가 점점 빨라질수록 경계 타원이 경계 원보다 장애물을 먼저 검색하는 것을 알 수 있다. 따라서 속도에 따라 유동적으로 반응하는 경계 타원을 이용함으로써 장애물의 검색시간이 단축되고 빠른 장애물 탐색이 가능하게 되어 주행의 속도저하를 개선하여 가상환경의 몰입감을 증대시킬 수 있게 되는 것이다. 실제 능동형 속도 타원 알고리즘의 적용 실험은 장애물의 위치를 임의로 설정한 것과 의도에 의해 설정한 가상환경으로 나누어 실험하였다.

3.2.2 장애물위치의 의도된 설정

복잡한 형태의 장애물 정보를 가상환경의 위치에 맞게 직접 설정하고 속도타원 알고리즘을 적용하였다. 프로그램에서 static 형태로 직접 좌표를 설정하여 구성한 장애물 블록들과 네비게이션 실험 결과를 Fig. 15에서 보인다. 능동형 속도 알고리즘이 적용되

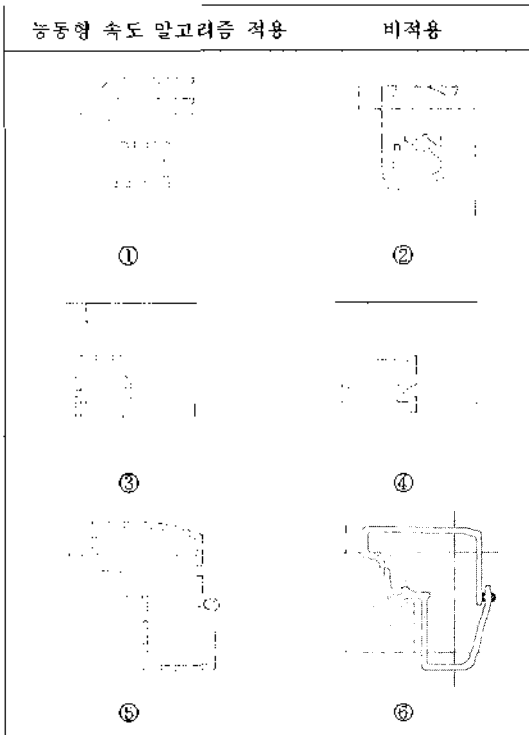


Fig. 15. Test results of the velocity ellipse navigation algorithm in pre-defined obstacles.

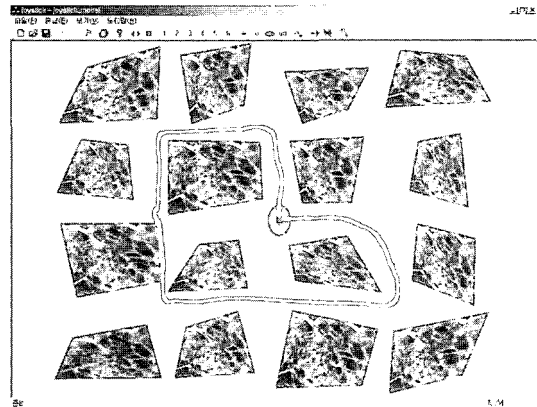
지 않았을 경우 주행속도가 느리기 때문에 사용자 움직임의 궤적에 불필요한 주행이 있는 것을 볼 수 있는 반면에 알고리즘이 적용되었을 경우 깨끗한 주행과 동시에 증가된 속도로 인하여 궤적의 선이 끊기는 것을 볼 수 있다. 따라서 본 논문에서 제안한 능동적 네비게이션 알고리즘의 적용으로 사용자의 주행 속도와 효율 측면에서 알고리즘이 없는 경우에 비해 몰입감 저하를 개선할 수 있었다.

3.2.3 장애물위치의 임의설정

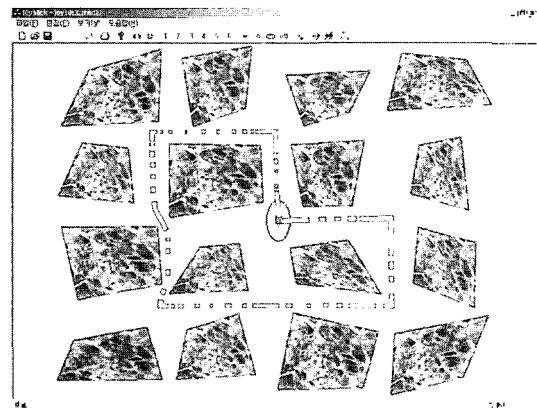
다음으로 가상환경의 장애물을 임의로 설정하여 알고리즘을 적용한 실험을 하였다.

Fig. 16(b)에서 보는 바와 같이 알고리즘을 적용한 경우가 속도의 향상이 있었기 때문에, 불필요한 주행을 줄이고 빠른 주행으로 인해 주행 궤적이 끊겨서 드문드문 침선이 생기는 것을 볼 수 있는 반면 Fig. 16(a)의 주행에서는 선정된 경로를 잘 따르기 위해 저속에서 주행이 이루어졌기 때문에 주행 궤적의 선이 연속적인 것을 볼 수 있다. 이는 알고리즘의 적용으로 인한 속도 향상을 보여준다.

Table 1에서 총 이동거리와 시간을 통한 전체 평균



(a) Without the algorithm



(b) With the velocity ellipse algorithm

Fig. 16. Navigation test results.

Table 1. Results of the navigation test

항목	방향	알고리즘 비적용	알고리즘 적용	단위
1. 이동거리	x	238.62	76.44	m
	y	222.01	63.37	m
전체 이동거리		460.63	139.80	m
2. 걸린 시간		11.31	2.61	sec
3. 평균속도	x	21.10	29.29	m/sec
	y	19.63	24.28	m/sec
전체 평균속도		40.73	53.56	m/sec

속도에서 알고리즘을 적용(with force feedback)한 주행이 알고리즘을 적용하지 않은(without force feedback) 주행에 비해 약 31.5% 정도 빠르다는 것을 보여준다. 또한 Table 1의 두 경우에서 전체 주행거리와 주행하는데 걸린 시간은 많은 차이가 있음을 알 수

있고, 이것은 빠른 주행 속도에서도 반응하는 경계타원으로 인해 장애물을 검색하고 합반향으로 통하여 속도의 감속 없이 장애물을 피할 수 있기 때문이다. 이는 속도타원 알고리즘을 적용함으로써 네비게이션의 속도감을 향상시키게 되어 몰입감 증대를 기대할 수 있음을 보여준다.

4. 결 론

본 연구에서 개발된 햅틱 조이스틱 장치는 요잉(yawing)운동과 피칭(pitching)운동을 만족할 수 있고 z축 손잡이 내부에 모터를 장착함으로 1자유도 형식이지만 2개의 모터를 사용하여 조작성이 좋아 가상 환경과의 상호 작용성을 높일 수 있다.

또한 본 연구에서 개발된 속도타원 네비게이션 알고리즘은 사용자의 주행방향을 제어기가 능동적으로 판단하여 주행거리의 단축과 전체 주행 평균속도의 증가로 몰입감을 증가시켰다. 제어가 사용자의 진행 방향으로 역감을 제시하여 진행을 도와줌으로써 장애물을 쉽게 피할 수 있도록 한 것이다. 이것은 럭비공 모양의 경계를 설정함으로써 3차원에서도 적용 가능하다. 개발된 2자유도 햅틱 yawing 조이스틱과 속도 타원 알고리즘을 기반으로 향후 다양한 인터랙티브 라이터의 역감제현 인터랙션 인터페이스를 기대할 수 있게 된다.

감사의 글

이 논문은 2003년도 중앙대학교 학술연구비(일반연구비) 지원에 의한 것임.

참고문헌

1. Ecole polytechnique federale de Lausanne(EPFL) <http://vrlab.epfl.ch/~thalmann/VRcourse/Haptic-s.pdf>.
2. PHANTOM™, <http://www.sensable.com>.
3. Grigore C. Burdea, *Force and Touch Feedback for*

- Virtual Reality*, John Wiley & Sons, New-York, 1996.
4. P. J. Berkelman, R. L. Hollis and S. E. Salcudean, "Interacting with Virtual Environments using a Magnetic Levitation Haptic Interface," *International Conference on Intelligent Robotics and Systems*, Pittsburgh, pp. 236-242, 1995.
5. 김경환, "인간 친화형 Haptic Interface 설계," 대한기계학회 2001년도 동역학 및 제어부분 동계 워크숍 논문집, pp. 25-45, 2001.
6. 윤정원, 류제하, 이혁기, "병렬기구를 이용한 6자유도 역감 제시장치의 제작 및 접도 시뮬레이션의 적용," 대한기계학회 2000년도 동역학 및 제어 부분 하계 학술대회 논문집, pp. 348-353, 2000.
7. 4자유도 햅틱다바이스, <http://dycon.kjist.ac.kr>.
8. Overview of Cybernet Systems Corporation Robotics, <http://www.cybernet.com>.
9. Matt Arcy, Design of a single degree of freedom "mechanical breadboard" haptic display, MS thesis, Department of Mechanical Engineering, Northwestern University, July, 1996.
10. J. Michael Brown, J Edward Colgate, "Physics-based Approach to Haptic Display," *ISMRC*, Vol. 3, pp. 101-106, 1994.
11. Immersion co., <http://www.immersion.com>.
12. C. Ramstein, V. Hayward, "The Pantograph: a Large Workspace Haptic Device for a Multi-modal Computer-Human Interaction," *Conference on Human Factors in Computing Systems, ACM/SIGCHI*, Boston, pp. 57-58, 1994.
13. Mohammad R. Siroospour, S. P. DiMaio, S. E. Salcudean, P. Abolmaesumi and C. Jones, "Haptic Interface Control Design Issues and Experiments with a Planar Device," *International Conference on Robotics and Automation*, San Francisco, pp. 24-31, 2000.
14. Kevin Hawkins, Dave Astle, *OpenGL game programming*, 정보문화사, pp. 715-724, 2001.
15. 이혁진, "동적 가상환경에서의 햅틱시뮬레이터 개발," 영남대학교 대학원 기계공학과 석사 학위논문, pp. 16-17, 2002.
16. William R. Mark, Scott C. Randolph, Mark F. inch, James M. Van Verth, Russell M. TaylorII, "Adding force feedback to graphics system: issues and solutions," *SIGGRAPH 96*, pp. 447-452, 1996.



채 영 호

1989년 중앙대학교 기계공학과
1994년 SUNY at Buffalo 기계공학석사
1997년 Iowa State University 기계공학
박사
1989년~1992년 (주)삼성전기 CAD/
CAM실 연구원
1998년 CASE Virtual Prototyping Lab.
Consultant
1998년~1999년 중앙대학교 기계공학부
조교수
1999년~현재 중앙대학교 첨단영상대학원
부교수
관심분야: Haptics, Virtual Design,
Physically based Modeling



윤 인 복

2001년 중앙대학교 기계공학과
2003년 중앙대학교 첨단영상대학원 영상
공학석사
관심분야: Virtual Environments, Haptics