

눈동자 시선 추적에 의한 3차원 1인칭 슈팅 게임

이 의 철[†] · 박 강 령^{††}

요 약

본 논문에서는 HMD(Head Mounted Display) 하단에 눈동자의 움직임 영상을 취득할 수 있는 USB 카메라를 부착한 후, 3차원 1인칭 슈팅(First Person Shooting) 게임에서 게임 캐릭터의 시선방향을 눈동자 움직임에 의해 조작하는 방법을 제안한다. 본 논문에서 제안하는 방법은 다음과 같은 3부분으로 이루어져 있다. 첫 번째는 입력 영상으로부터 눈동자의 중심 위치를 실시간 영상 처리 방법으로 추출하는 부분, 두 번째는 HMD 모니터상의 임의 지점을 쳐다볼 때 추출된 눈동자의 위치 정보와 모니터상의 응시 지점사이의 기하학적인 연관관계를 결정하는 캘리브레이션 부분, 그리고 마지막은 캘리브레이션 정보를 기반으로 모니터 상의 최종적인 응시 위치를 결정하고 이 정보에 의해 게임상의 3차원 뷰 방향을 조정하는 부분으로 구성되어 있다.

실험 결과 본 논문의 방법에 의해 손이 불편한 사용자에게 게임을 즐길 수 있는 기회를 제공하고, 게임 캐릭터와 게임 사용자의 시선 방향을 일치시킴으로서 게임의 흥미와 몰입감을 증가시킬 수 있는 결과를 얻음을 알 수 있었다.

키워드 : 눈동자의 움직임 추적, 3차원 1인칭 슈팅 게임

3D First Person Shooting Game by Using Eye Gaze Tracking

Eui Chul Lee[†] · Kang Ryoung Park^{††}

ABSTRACT

In this paper, we propose the method of manipulating the gaze direction of 3D FPS game's character by using eye gaze detection from the successive images captured by USB camera, which is attached beneath HMD. The proposed method is composed of 3 parts. At first, we detect user's pupil center by real-time image processing algorithm from the successive input images. In the second part of calibration, when the user gaze on the monitor plane, the geometric relationship between the gazing position of monitor and the detected position of pupil center is determined. In the last part, the final gaze position on the HMD monitor is tracked and the 3D view in game is controlled by the gaze position based on the calibration information.

Experimental results show that our method can be used for the handicapped game player who cannot use his(or her) hand. Also, it can increase the interest and the immersion by synchronizing the gaze direction of game player and the view direction of game character.

Key Words : Eye Gaze Detection, 3D First Person Shooting Game

1. 서 론

사용자의 눈동자 움직임에 의한 시선 위치를 실시간으로 추적함으로써 자연스러운 컴퓨터 인터페이스를 하려는 방법들이 기존에 많이 연구되어 오고 있다[1~6]. 이처럼 눈동자 움직임에 의한 시선 추적 인터페이스 방법이 가지는 장점은 여러 가지가 있을 수 있다. 첫 번째, 눈동자 움직임에 의한 시선 추적 인터페이스 방법은 실제 마우스의 작동방법과 유사한 프로토콜을 가지고 있어 사용자의 자연스러운 컴퓨터 동작을 흉내 낼 수 있다는 점이다. 또한 손과 같은 신체의

일부를 사용하는 입력 장치들에 비해 컴퓨터를 동작시키는 속도를 매우 빠르게 할 수 있다[2~4]. Yamato의 연구에서는 21인치 모니터의 좌측상단에서 우측하단을 바라볼 때 걸리는 시간이 150ms라고 나타낸 바 있다[4]. 이외에도 눈동자 움직임에 의한 시선 추적 인터페이스는 손이 불편한 장애인들 역시 손쉽게 컴퓨터를 사용할 수 있게 한다는 장점을 가지고 있다.

기존에 눈동자의 움직임에 의한 시선 추적 방법은 크게 2차원적인 접근 방법과 3차원적인 접근 방법으로 나눌 수 있다. 2차원적인 접근 방법에서는 사용자 눈의 3차원 구조 및 움직임을 고려하지 않고, 입력 영상에서 추출된 사용자 눈의 2차원적인 위치 정보를 모니터 상의 시선 위치를 직접 연결시키는 매핑 함수에 의해 모니터 상의 시선 위치를 파악하게 된다[9~11]. 이밖에도 3차원 적인 접근 방법에서는

† 준 회 원 : 상명대학교 일반대학원 컴퓨터학과 석사과정
 †† 정 회 원 : 상명대학교 소프트웨어대학 미디어학부 조교수
 논문접수 : 2005년 1월 19일, 심사완료 : 2005년 6월 8일

카메라, 모니터, 눈동자 좌표계 사이의 3차원 관계 및 눈동자의 3차원 구조, 3차원 위치 정보 등을 같이 모델링 함으로써 모니터상의 시선 위치를 추출하는 연구를 수행하였다 [12~14]. 일반적으로 3차원적인 방법이 시선 위치의 정확도가 우수하며, 초기 캘리브레이션이 쉽다는 장점이 있으나, 2차원적인 방법에 비해 계산량이 많이 소요되고 처리시간이 오래 걸리는 문제점이 있다. 이에 본 논문에서는 일반적인 게임환경을 고려할 때 눈동자 시선 추적 및 게임 렌더링이 일반적인 하나의 컴퓨터에서 동작되어야 하므로, 2차원적인 방법에 의해 시선 추적하는 방안을 사용하고자 한다.

기존에 눈동자 움직임 추적 기술을 게임과 접목시킨 연구는 조사된 바 없으나, 가상현실 시스템과는 접목되어 많은 연구가 진행되었다[5][7]. 이 연구에서는 양 눈의 움직임을 양안 카메라(stereo camera)에 의해 추적함으로써, 비행기내의 짐칸(cargo bay)에서 존재하는 결함을 검사하는 시뮬레이션 훈련 목적으로 개발되었다. 특히 이 연구에서는 눈동자 움직임을 추적하기 위한 영상처리 전용 컴퓨터와 상세한 3D 그래픽의 표현을 위한 고가, 고사양의 그래픽 전용 하드웨어 플랫폼(SGI Onyx2 InfiniteReality™[8])을 동시에 사용하였다. 그러나 본 연구에서와 같이 일반적인 1인칭 슈팅 게임 환경은 눈동자 움직임 추적 및 게임 렌더링이 하나의 컴퓨터에서 동작되어야 하므로, 위와 같은 고사양의 방법은 적용하기 힘든 문제가 있다.

특히 본 연구에서는 사람이 임의의 한 곳을 응시할 때 양쪽 눈은 같은 곳을 응시한다는 일반적인 특징을 이용하여 한쪽 눈의 응시위치만을 추적하고, 동공이 움직일 수 있는 사전 범위를 미리 지정하여 제한된 범위에서만 영상처리를 수행함으로써 처리 시간을 줄일 수 있었다. 또한, 게임 렌더링의 부하를 줄이기 위해 2D 텍스처 매핑으로 3D 효과를 냄으로써 한 장면 당 폴리곤 수를 1000개 이하로 줄이는 low-polygon 모델링 기술을 사용함으로써, 현대의 컴퓨터 환경에서 눈동자의 움직임 추적 및 게임 렌더링을 실시간으로 동작시킬 수 있었다.

2장에서는 HMD와 USB 카메라를 이용해서 눈동자 시선 위치 추적을 위한 장치를 제작하는 과정에 대한 설명을 하였다. 3장에서는 동공의 중심위치 추출, 캘리브레이션 과정, 모니터 상의 사용자 시선 위치 계산을 통해 게임 캐릭터의 시선방향을 조작하는 일련의 과정을 설명하였다. 4장에서는 눈동자 시선 위치 추적 인터페이스를 적용하기에 적절한 게임 장르에 대한 고찰을 통해 3차원 1인칭 슈팅게임이 가장 적절한 이유를 설명하였다. 5장에서는 본 시스템에 대한 주관적인 실험과 객관적인 실험의 결과를 제시하고 분석하였다.

2. 눈동자의 시선 위치 추적을 위한 장치

2.1 HMD (Head Mounted Display)

본 연구에서 사용자는 (그림 1)과 같이 머리에 착용하는 HMD를 통해 일반적인 모니터에서 볼 수 있는 화면을 응시



(그림 1) HMD (Head Mounted Display)

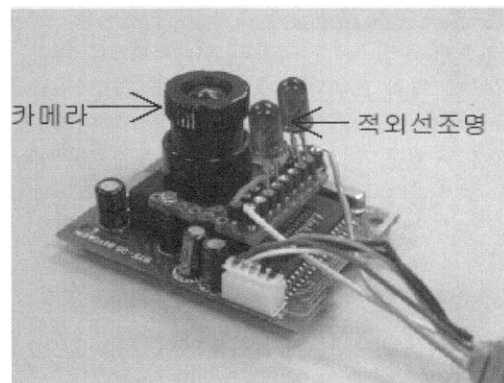
하게 된다. HMD를 통해 화면을 보면서 HMD의 하단에 고정되어 있는 USB카메라는 사용자의 눈동자 움직임을 연속 영상으로 받아들인다. 사용자 머리에 고정하여 착용하는 HMD를 통하여 눈 영상이 머리의 움직임에 영향을 받지 않기 때문에 별도의 머리 움직임 추적 장치는 필요로 하지 않는다. 본 논문에서 사용한 HMD(iglass-SVG3D)는 다음과 같은 특성을 지니고 있다[15].

- 화면 해상도: 800×600 픽셀(SVGA)
- FOV(Field of View): 26°

2.2 눈 영상 취득 장치

(그림 2)와 같이 눈 영상 입력 장치인 USB카메라(CMOS 센서 방식)는 HMD 하단에 고정되어 사용자의 한 쪽 눈을 지속적으로 촬영하여, 응시 위치를 추적하는데 필요한 눈 영상을 획득한다. 사용되는 카메라는 USB(Universal Serial Bus) 방식이므로 Frame Grabber와 같은 별도의 ADC(Analog to Digital Converter) 장비를 필요로 하지 않으며, 전체 시스템의 소형화와 경량화 측면에서 매우 적합하다고 할 수 있다. CMOS 센서방식의 카메라를 이용한 이유는 CMOS의 특성상 카메라 부가 회로의 크기를 작게 만들 수 있으므로 시스템 경량화에 도움이 되기 때문이다. 입력 영상 해상도는 640×480 픽셀이며, JPEG 5:1 압축을 적용하여 USB 1.0에서 초당 15프레임의 속도로 영상을 입력 받을 수 있었다 [16].

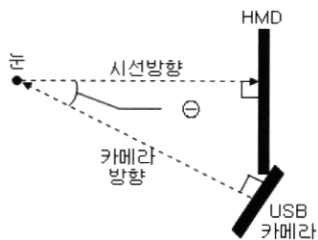
특히, 눈 영상을 획득할 때 뚜렷한 동공의 경계를 가지는 영상을 획득하며, 외부광의 영향을 받지 않고 일정한 밝기



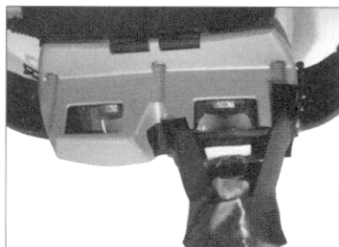
(그림 2) USB 카메라와 적외선 조명

의 영상을 얻기 위해서 카메라 내부에 존재하는 ‘적외선 차단필터’를 제거하고, 렌즈 전면에는 가시광선은 통과시키지 않고 적외선은 통과시키는 ‘적외선 투과필터’를 부착하였다. 그리고 (그림 2)와 같이 적외선 조명(Infra-Red Light Emitting Diode)은 카메라 렌즈의 하단에 부착하여 카메라와 일체형으로 제작, 소형화 되도록 하였다. 일반적으로 700nm 이상의 파장을 가지는 적외선 조명은 파장 대에 따라 다음과 같은 특성을 가지게 된다[6]. 700~800nm의 파장을 가지는 적외선 조명은 눈의 홍채와 흰자의 경계를 뚜렷하게 함으로써, 입력 영상에서 홍채 영역을 추출하는데 사용하기에 적합하다. 800~900nm의 파장을 가지는 적외선 조명은 홍채와 동공의 경계를 뚜렷하게 함으로써, 동공 위치를 추출하는데 사용하기에 적합하다. 일반적으로 홍채 영역은 사용자의 위아래 눈꺼풀(Eyelid) 등에 의해 덮여 있는 경우가 많으므로, 정확하게 추출하기가 어렵다. 반면 동공의 경우 눈꺼풀이나 눈썹 등에 의해 덮여 있는 경우가 작으므로, 본 논문에서는 사용자의 동공 위치를 추출하여 이러한 동공 움직임에 의해 모니터 상의 시선 위치를 파악하는 연구를 수행하였으며, 이를 위하여 850nm의 적외선 조명 2개를 사용하였다. 사용되는 적외선 조명은 조사각도에 비하여 눈과 상당히 인접한 거리에 있으므로 빛이 고르게 퍼지지 못하고 일부분만을 밝게 되는 문제가 있었다. 이를 극복하기 위해서 반투명 유리테이프를 적외선 조명 앞에 부착하여 빛이 고르게 분산(diffusing)될 수 있도록 하였다.

USB 카메라는 HMD의 하단에 부착되므로 사용자의 눈 영상 획득 시에는 (그림 3)과 같이 모니터를 향한 사용자의 시선 방향과 사이 각(θ) 만큼의 차이를 가지게 된다. 이 각이 커질수록 눈동자의 상하 움직임에 대한 영상 해상도가 떨어지고, 각이 적을수록 눈동자의 상하 움직임에 대한 영상 해상도가 높아짐으로써, 사용자의 눈 움직임을 보다 정확하게 추출할 수 있게 된다. 본 연구에서는 이 사이 각을



(그림 3) 카메라와 눈의 시선 방향이 이루는 사이 각(θ)



(그림 4) HMD 및 HMD에 부착된 눈영상 취득용 카메라 장치

줄이기 위해서 HMD의 외곽 케이스로 인해 눈 영상 취득이 방해받지 않을 정도의 범위 내에서 최대한 뒤(눈에 멀리)에 카메라를 부착하도록 하였다.

(그림 4)는 본 연구에서 사용하는 HMD 및 HMD에 부착된 눈 영상 취득용 카메라 장치를 나타낸 것이다.

3. 눈동자 시선 추적 인터페이스의 구현과 게임 캐릭터의 시선방향 조정

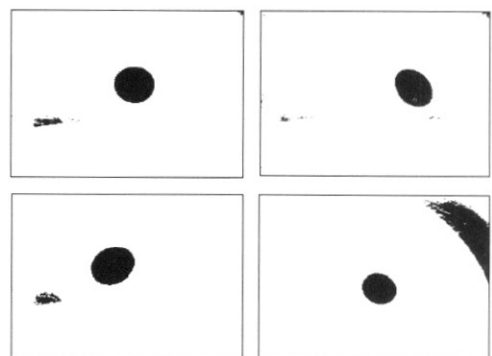
3.1 동공의 중심 위치 추출

진술한 바와 같이 본 논문에서는 입력영상에서 동공의 중심 위치를 추출함으로써 사용자의 시선 위치를 파악하였다. 동공 위치를 추출하기 위하여 다음과 같은 과정을 거치게 된다.

3.1.1 동적 이진화 과정

본 연구에서 제안하는 눈영상 취득용 카메라 장치에서는 USB 카메라 앞에 부착된 ‘적외선 투과 필터’에 의해 가시광선이 차단되고, 별도 부착된 적외선 조명에 의해 영상이 조사(Illumination)되므로 어느 정도 외부광의 영향을 받지 않고 일정한 밝기의 영상을 얻을 수 있게 된다. 그러나 적외선 이상의 파장을 지니고 있는 외부광(예, 태양광, 할로젠, 백열등 등)이 있는 경우에는 그렇지 않은 경우보다 입력 영상의 밝기가 밝게 된다. 그러므로 본 연구에서는 입력영상의 평균 밝기를 측정하여 이를 이진화 임계치로 반영하는 방식을 사용하였다. 그 결과 입력 영상의 밝기에 상관없이 적절한 이진화 임계치를 동적으로 결정할 수 있었다.

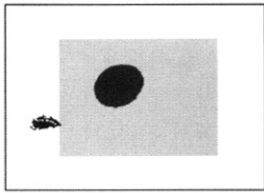
일반적으로 동공은 그 내부가 뚫린 구조를 가지고 있으므로[17], 입력 영상에서도 동공은 다른 부분보다 어두운 밝기를 가지게 된다. 그러므로 이진화를 통하여 동공의 대략적인 위치를 알 수 있다. (그림 5)는 이진화를 통해 나타나는 대략적인 동공의 영상을 나타낸 것이다.



(그림 5) 이진화를 통해 얻어진 동공 영상

3.1.2 동공 중심 위치 추출

(그림 5)와 같이 이진화를 거친 영상의 흑화소들은 라벨링을 거치게 된다 [17]. 사용자가 HMD를 일반적으로 착용



(그림 6) 제한 영역에서의 라벨링 수행 결과

하였을 때 동공 위치가 입력 영상의 중앙부에 위치하고, 동공의 이동범위는 영상의 크기에 비해 제한적인 점을 감안하여 640×480 픽셀 영상에서 중심을 기준으로 400×300 픽셀의 영역만을 대상으로 라벨링(Labeling)을 수행함으로써 처리시간을 단축하였다. 실제로 50명을 대상으로 눈 영상에서 동공의 이동 범위를 측정된 결과 평균 가로 94.5 pixel, 세로 44.7 pixel로 나타났다. 라벨링 처리 결과 가로대 세로비가 어느 정도 원이나 타원에 가까운 영역만을 동공의 후보로 간주하였으며, 그 외의 영역은 제거하였다(그림 6 참조). 라벨링 처리 이후, (그림 6)과 같이 제한된 영역 내에는 동공 이외에 다른 영역은 존재하지 않게 된다. 이후 본 논문에서는 이 제한된 영역(중심을 기준으로 400×300 픽셀의 영역) 내에서 흑화소들에 대한 수평, 수직 히스토그램을 구하고 이 정보로부터 동공의 대략적인 위치와 반경을 구하게 된다. 이로부터 보다 작은 세밀한 탐색 영역을 선정하여 이 영역 내에서만 흑화소들의 X, Y축 무게중심을 구함으로써 최종적인 동공의 중심 위치를 구하게 된다. (식 1)과 (식 2)는 흑화소들로부터 동공의 무게중심을 구하는 식이다. 이후 매 입력 영상마다 이 과정을 반복함으로써 연속 영상에서 동공 위치를 계속 추적할 수 있게 된다. 추출된 동공 위치의 정확도를 파악하기 위하여, 눈으로 직접 보면서 표시한 동공의 중심 위치와 본 논문의 방법에 의해 자동으로 추출된 위치사이의 픽셀 RMS 에러를 측정하였다. 총 20명의 실험자로부터 각각 100장씩, 총 2,000장의 영상에 대해 측정된 결과 약 2.43 픽셀(0.06°) 에러(X축 방향으로 1.2 픽셀(0.03°) 에러, Y축 방향으로 2.1 픽셀(0.05°) 에러)가 발생함을 알 수 있었다.

$$x_{ic} = \frac{1}{n_x} \sum x \quad (if I(x, y) = 0) \quad (식 1)$$

$$y_{ic} = \frac{1}{n_y} \sum y \quad (if I(x, y) = 0) \quad (식 2)$$

(식 1, 2)에서 $I(x, y)$ 는 입력 영상의 (x, y) 위치에서의 픽셀 그레이값을 나타내며, n_x, n_y 는 각각 흑화소의 총 수를 나타낸 것이다.

3.2 캘리브레이션 과정

HMD 모니터 상에 사용자 시선 위치를 파악하기 위해서는, 추출된 동공의 중심 위치가 HMD 모니터상의 어느 위치에 해당하는지를 계산하기 위한 매핑 함수의 두 개의 입력

값이 필요하다. 본 과정을 캘리브레이션 과정이라 한다. 본 논문에서 캘리브레이션 과정은 모니터 화면의 우측상단(X_{ru}, Y_{ru})과 좌측하단(X_{ld}, Y_{ld})을 응시했을 때, 각각의 눈 영상에서의 동공 중심 좌표를 저장하는 과정으로 이루어진다.

3.3 모니터 상의 시선 위치 추출

캘리브레이션 과정을 거친 후에는 화면의 우측상단과 좌측 하단을 응시했을 때의 동공 중심좌표를 알고 있으므로, (식 3) 및 (식 4)와 같이 선형 보간법(Linear Interpolation)을 이용한 매핑함수에 의해 사용자가 화면의 어느 곳을 응시(X_{gaze}, Y_{gaze})하고 있는지를 계산할 수 있다.

$$X_{gaze} = 800 \times (X_{rec} - X_{ru}) / (X_{ld} - X_{ru}) \quad (식 3)$$

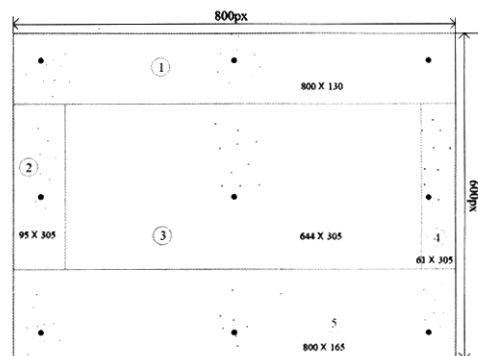
$$Y_{gaze} = 600 \times (Y_{rec} - Y_{ru}) / (Y_{ld} - Y_{ru}) \quad (식 4)$$

(식 3)과 (식 4)에서 X_{ld}, Y_{ld} 는 캘리브레이션 과정에서 좌측하단을 응시했을 때 추출된 동공의 중심 좌표를 나타내며, X_{ru}, Y_{ru} 는 우측상단을 응시했을 때 추출된 동공의 중심 좌표를 나타낸다. X_{rec}, Y_{rec} 는 현재 사용자가 모니터상의 한 지점을 응시하고 있는 순간에 추출된 동공의 중심 좌표를 나타낸다. 800과 600은 각각 모니터의 X, Y축 해상도를 나타낸다.

3.4 마우스 커서 및 게임 캐릭터의 시선 방향 조정

3.3절에서 구한 화면상의 시선 위치 (X_{gaze}, Y_{gaze})에 마우스 커서를 위치시킬 수 있다. 그런데, 사람의 눈은 한곳을 응시하고 있다고 해도 일반적으로 미세한 떨림(saccadic movement)이 존재한다. 이러한 미세한 떨림으로 인해 마우스의 커서 역시 의도하지 않게 떨리는 것을 방지해야 한다. 이런 이유로 본 연구에서는 이전 영상에서 추출된 X_{rec}, Y_{rec} 값과 현재 영상에서 추출된 X_{rec}, Y_{rec} 와의 거리차를 구해서 정해진 임계치 이하의 값(본 연구에서는 5 픽셀을 사용함) 이하이면 현재 영상에서 추출된 값을 무시하고 이전 값을 그대로 취하는 방법을 사용하였다.

이러한 시선 위치 및 마우스 커서 위치를 이용하여 게임 화면에서 캐릭터의 시선 방향에 따른 화면 조정을 하게 된다. 본 연구에서는 게임화면에서 화면 조정을 위해 (그림 7)



(그림 7) 게임화면에서 응시위치 판단을 위한 5개 화면 구역

과 같이 전체 화면을 크게 5개 영역으로 나누어 화면 위쪽을 쳐다보는 경우(①), 화면 아래를 쳐다보는 경우(⑤), 화면 왼쪽을 쳐다보는 경우(②), 화면 오른쪽을 쳐다보는 경우(④), 그리고 화면 중앙을 쳐다보는 경우(③)로 정의함으로써, 각각의 영역을 응시하는 경우 게임 캐릭터의 시선 방향을 해당 방향으로 회전시키도록 하였다.

5개 응시 영역의 크기를 정하기 위하여, 10명을 대상으로 실험을 실시하였다. 실험을 위해 사용자에게 (그림 7)에서와 같은 총 9개의 검은색 점을 응시하게 하고, 해당 위치를 응시했을 때 계산된 시선 위치를 측정하였다. 실험결과 (그림 7)의 9개 검은색 점 부근에 작은 점들로 표시되어 있다.

실험 결과, 사용자가 모니터 중앙과 하단을 응시할 때 보다 많은 시선 위치 추출 에러가 발생함을 알 수 있었으며, 이러한 에러를 보정하기 위하여 본 연구에서는 모니터 중앙 부분과 하단 응시 영역을 보다 크게 설정하였다.

4. 시선 추적 인터페이스를 이용한 게임 구동

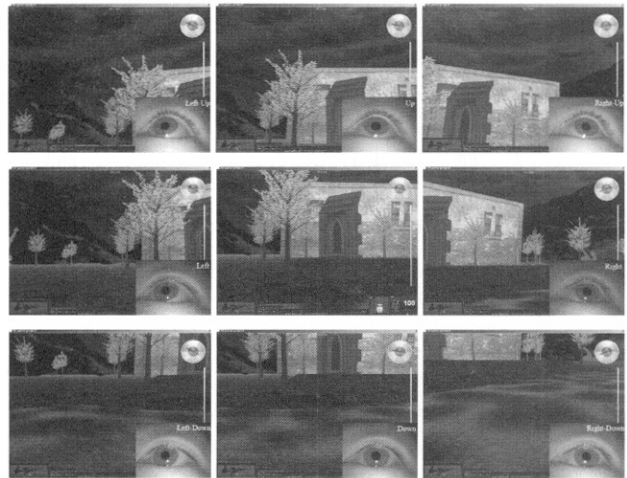
4.1 시선 추적 인터페이스를 적용할 게임 장르 선택

기존의 게임 장르는 게임 주인공의 시점에 따라 크게 1인칭 게임과 3인칭 게임으로 나눌 수 있다. 1인칭 게임은 게임의 주인공이 모니터에 등장하지 않고, 사용자가 배경과 활동무대를 직접 바라보며 게임이 진행된다. 그러므로 캐릭터라는 중간 매개물이 없이 자신이 직접 게임의 주인공인 듯한 느낌을 주어 큰 몰입감을 가질 수 있는 장점이 있다. 반면, 3인칭 게임은 사용자가 조작하는 주인공이 모니터에 등장하며, 사용자와 게임의 배경, 활동무대 간에 캐릭터라는 매개물이 사용자의 눈에 보인다. 그러므로 1인칭 게임과 같은 몰입감을 가질 수는 없지만, 사용자에게 ‘어떤 물체 또는 캐릭터를 내가 조작할 수 있다’는 매력을 주는 장점이 있다.

본 연구에서는 시선 추적 인터페이스를 이용하여 게임의 주인공을 움직이게 하기보다는 ‘게임 사용자가 바라보는 곳을 게임 캐릭터가 바라본다’라는 장점을 부각시켜서 몰입감을 증대시키고, 사용자가 직접 임의의 목적달성을 위한 게임의 주인공이 되기에 적당한 1인칭 게임을 적용 장르로 선택하고자 한다.

또한, 기존의 1인칭 게임은 게임 내용과 주인공에 따라 몇 가지로 분류할 수 있다. 비행기를 조정하여 적을 물리쳐서 정해진 임무를 수행하는 비행시뮬레이션 게임(Flight Simulation Game), 자동차를 조정하여 순위를 다투거나 기록 단축을 목적으로 하는 자동차 레이싱 게임(Car Racing Game), 사람 또는 로봇이 등장해서 총과 같은 무기로 적을 물리치는 1인칭 슈팅 게임(First Person Shooting Game)이 있다. 각 게임의 장르에 시선 추적 인터페이스를 적용하고자 할 때의 장단점을 살펴보기로 한다.

먼저, 비행시뮬레이션 게임에서 시선 추적 인터페이스를 적용할 경우에는 고속으로 비행하는 중에 장애물을 피하거나 좁은 계곡등도 효과적으로 피할 수 있어야 하므로, 사용자의 시선 위치에 따라 게임 뷰 화면이 고속으로 변화되어



(그림 8) 게임 사용자의 시선방향에 따른 1인칭 게임 캐릭터의 시선위치 변화

야 한다. 이러한 것은 게임 플레이어에게 어지러움과 메스꺼움등 악영향을 줄 수가 있어서 시선 추적 인터페이스를 적용하는 것에 적당치 않은 것으로 판단된다.

레이싱 게임 역시 고속 운전을 하는 특징이 있으므로, 비행시뮬레이션에서와 같이 사용자에게 어지러움등을 유발할 확률이 있으며, 특히 레이싱 게임에서는 게이머가 정해진 주행 길만 쳐다보고 자동차를 움직이며 고속 주행 중에 주변을 돌아보지 못하는 특징이 있으므로, 역시 시선 추적 인터페이스를 적용하기에 적합하지 않은 것으로 판단된다.

3차원 1인칭 슈팅 게임에서는 사람이나 로봇 등의 캐릭터가 주인공이고, 1인칭이므로 캐릭터의 모습이 보이지 않고 기껏해야 손과 무기가 보일 뿐이다. 또한 걸음을 통한 캐릭터 자체의 이동은 키보드를 이용하며 시선방향의 이동은 마우스를 이용함으로써 두 입력 장치의 역할이 엄격히 분리되어 있다는 특징이 있다. 그리고 일반적으로 캐릭터가 비행시뮬레이션이나 레이싱 게임보다는 저속으로 움직이면서 주변 환경을 세심히 살펴야 하는 특징이 있으므로, 본 연구에서는 시선 추적 인터페이스를 접목하기 위한 1인칭 게임의 세부장르로 1인칭 슈팅 게임을 선택하고자 한다. 시선 추적을 이용하여 게임 캐릭터의 시선방향이 전 방향으로 이동하는 것을 조정할 수 있으므로 그 활용성이 높다는 장점이 있으며, 또한 사용자의 시선이 곧 캐릭터의 시선이 된다는 일치성이 게임의 몰입감을 증가시킬 수 있을 것으로 예상된다.

본 연구에서는 실험을 위해 높낮이를 가지고 있는 지형 및 주인공을 기준으로 적이 어느 방향에 있는지 알아볼 수 있는 레이더를 가지는 3차원 1인칭 슈팅 게임을 제작하여 실험에 활용하였다. 본 시스템을 적용한 게임화면은 (그림 8)과 같다. 사용자의 시선 방향에 따라 캐릭터가 바라보는 게임상의 뷰 방향이 변화됨을 알 수 있다.

5. 실험 결과 및 분석

실험 환경은 다음과 같다. 하드웨어는 Pentium-4 CPU

2.40GHz, 512MB RAM과 Geforce-4 FX 5200, 64MB RAM 비디오 카드를 사용하였다. 시선 추적 인터페이스는 Microsoft사의 Visual Studio C++ 6.0, DirectX 9.0 SDK DirectShow를 사용하여 구현하였다. 3D FPS Game은 DirectX 9.0 SDK의 Direct3D, DirectX Audio, DirectPlay, DirectInput등을 사용하여 구현하였다.

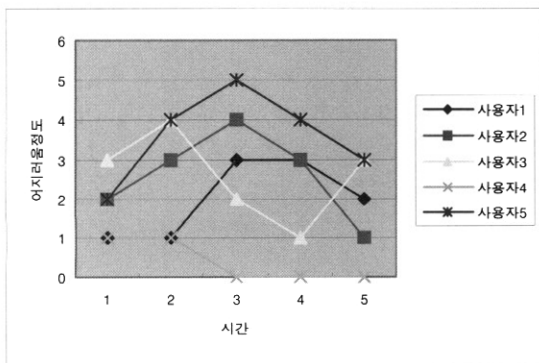
시선 추적 인터페이스에 의한 3차원 1인칭 슈팅게임의 효율성, 흥미, 몰입감, 편의성 등을 측정하기 위해, 총 20명의 실험자들을 대상으로 크게 주관적인 평가(subjective test)와 객관적인 평가(objective test)로 나누어 실험하였다.

5.1 주관적인 평가(Subjective Test)

주관적인 평가에서는 먼저 사용자가 본 시스템을 사용하면서 어느 정도의 어지러움(Dizziness)을 느끼는지를 조사하여 본 시스템이 사용자에게 끼치는 영향을 분석한다. 그리고 일반 마우스를 사용해서 게임을 했을 때와 시선 추적 인터페이스 시스템을 사용하여 게임을 했을 때 사용자는 어느 것에 더 흥미(Interest)와 몰입감(Immersion)을 느끼는지를 설문문을 통해 측정한다.

5.1.1 사용자 어지러움 측정

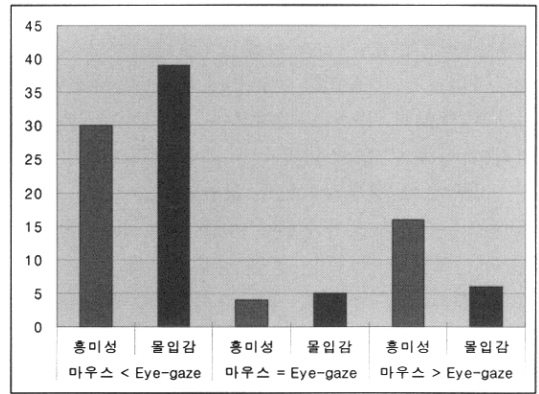
사용자들에게 본 시스템을 5분간 사용하게 하여 1분 간격으로 느끼는 어지러움 정도를 측정하였다. 어지러움 정도는 5단계의 값으로 측정하였다. '5'는 '매우 어지러움', '4'는 '약간 어지러움', '3'은 '보통', '2'는 '거의 어지럽지 않음', '1'은 '전혀 어지럽지 않음'으로 하여 측정한 결과는 (그림 9)와 같다. 실험 결과 대부분의 사용자들이 처음 시작부터 3분 정도까지는 어지러움이 많이 증가된다고 대답하고 있으나, 이후에는 이후 시스템에 대한 반복 사용과 적응을 통해 점차 어지러움이 없는 것으로 조사되고 있다.



(그림 9) 5명의 사용자의 어지러움 정도 변화 추이에

5.1.2 일반 마우스와 시선 추적 인터페이스의 몰입감과 흥미성 비교

본 실험에서는 사용자들에게 기계적인 마우스와 시선 추적 인터페이스를 각각 5분간 사용하여 게임을 하도록 하여 어느 쪽에 더 몰입감(immersion)과 흥미성(interest)이 높은지를 비교하였다. 결과는 (그림 10)과 같다. (그림 10)에서



(그림 10) 흥미성과 몰입감에 대한 설문 결과

“마우스 < Eye-gaze”는 마우스보다 시선 추적 방법을 사용했을 때 흥미성과 몰입감이 높다는 의미이다. 이외에 “마우스 = Eye-gaze”와 “마우스 > Eye-gaze”는 각각 마우스와 시선 추적 방법을 사용했을 때 흥미성과 몰입감이 같거나, 오히려 마우스 사용 시가 높다는 것을 의미한다.

몰입감에 대한 설문결과를 보면 ‘시선 추적 인터페이스를 사용한 게임이 더 몰입감이 크다’라는 의견이 압도적으로 많은 것을 알 수 있다. 이 의견을 선택한 사용자들은 그 이유로 ‘HMD를 사용하여 큰 화면이 외부환경에 영향을 받지 않고 집중할 수 있도록 하는 것’과 ‘내 눈에 따라 캐릭터의 시선이 바뀌는 것’을 꼽았다.

흥미성에 대한 설문결과를 봐도 ‘시선 추적 인터페이스를 사용한 게임이 더 흥미성이 높다’라는 의견이 압도적으로 많았다. 그 이유로는 ‘신기하다’, ‘내 눈에 따라 캐릭터의 시선이 움직인다’, ‘새로운 장치에 대한 호기심’을 꼽았다.

마우스가 시선 추적 인터페이스보다 몰입감이나 흥미성이 높다고 한 응답자들은 그 이유로 ‘눈의 움직임에 신경을 쓰다보니 게임에 집중할 수 없다’를 꼽았다.

실험결과 사용자들이 시선 추적 인터페이스의 사용을 통해 보다 큰 몰입감과 흥미를 느끼는 것으로 나타남으로써, 시선 추적 인터페이스와 게임의 접목에 더 큰 가능성을 가지게 되었다.

5.3 객관적인 평가(Objective test)

객관적인 평가에서는 사용자의 시선 위치 정확도 및 게임이 시작되어 게임 상에 적이 있는 위치까지 도달하는 시간을 기계적인 마우스를 사용했을 때와 시선 추적 인터페이스의 두 가지 경우로 나누어 측정, 비교분석하였다.

5.3.1 사용자의 시선 위치 정확도

모니터 상에 사용자의 시선 위치 정확도를 파악하기 위하여 (그림 7)과 같이 모니터 상의 9군데(검은색 점)를 사용자가 응시하도록 하고 이때 계산된 시선 위치와 실제 응시해야 할 위치사이의 RMS 에러를 측정하였다. 실험 결과 약 57픽셀(1.31°)(X축으로는 38픽셀(0.87°)의 에러, Y축으로는 43픽셀(0.99°)의 에러 (HMD 모니터 해상도: 800×600 픽셀)의

시선 위치 추출 오차 성능을 나타냈다. 하지만 본 논문에서는 정확한 시선 위치보다 모니터의 어느 방향을 응시하고 있는지가 중요하다. 그러므로 (그림 7)과 같이 응시 영역을 5군데로 나누어 설정한다면 캐릭터 시선 방향의 이동에는 문제가 없다. 실제로 50명을 대상으로 한 실험에서, 실험 진행자가 지시하는 임의의 방향으로 캐릭터 시선 방향을 이동하도록 했을 때, 4500번중 4339번을 지시대로 이동하여, 96.7%의 정확도를 나타내었다.

5.3.2 적의 위치까지 도달하는 시간

본 실험은 <표 1>과 같이 사용자들이 5번 반복 사용하여 게임 중 임의로(Randomly) 생성된 적 캐릭터의 위치까지 도달하는 데 걸리는 시간을 매번 측정하여 변화 추이를 분석하였다. 또한 <표 2>와 같이 일반 기계적인 마우스를 사용하여 걸리는 시간도 측정하여 비교해 보았다.

<표 1> 시선 추적 인터페이스의 반복 사용함에 따라 게임상의 적을 찾는데 걸리는 시간

사용자 \ 횟수	1	2	3	4	5
사용자 1	33초	27초	29초	19초	16초
사용자 2	30초	19초	22초	17초	14초
사용자 3	24초	25초	23초	17초	16초
사용자 4	32초	31초	26초	22초	16초
사용자 5	27초	19초	17초	14초	15초

<표 2> 기계적인 마우스를 반복 사용함에 따라 게임상의 적을 찾는데 걸리는 시간

사용자 \ 횟수	1	2	3	4	5
사용자 1	15초	15초	14초	15초	15초
사용자 2	14초	15초	17초	15초	15초
사용자 3	16초	18초	17초	16초	16초
사용자 4	14초	17초	18초	15초	15초
사용자 5	18초	18초	17초	16초	16초

실험 결과 시선 추적 인터페이스를 사용하는 초기에는 시간이 많이 소요됨을 알 수 있었으나, 반복 횟수가 증가됨에 따라 기계적인 마우스를 사용했을 때의 시간과 거의 비슷해 짐을 알 수 있었다. 기계적인 마우스를 사용했을 때는 반복 회수의 증가에 따라 소요시간에 큰 변화가 없었다. 이는 게임상의 적 위치가 임의로(Randomly)로 생성되었기 때문이며, 대부분의 사용자 이미 기계적인 마우스에 익숙해져 있으므로, 반복회수 증가에 따른 성능 향상의 효과가 없기 때문이다.

5. 결 론

본 논문에서는 HMD(Head Mounted Display) 하단에 노동자의 움직임 영상을 취득할 수 있는 USB 카메라를 부착한 후, 3차원 1인칭 슈팅(First Person Shooting) 게임에서

게임 캐릭터의 시선방향을 노동자 움직임에 의해 조작하는 방법을 제안한다. 본 논문에서 제안하는 방법은 다음과 같은 3부분으로 이루어져 있다. 첫 번째는 입력 영상으로부터 노동자의 중심 위치를 실시간 영상 처리 방법으로 추출하는 부분, 두 번째는 HMD 모니터상의 임의 지점을 쳐다볼 때 추출된 노동자의 위치 정보와 모니터상의 응시 지점사이의 기하학적인 연관관계를 결정하는 캘리브레이션 부분, 그리고 마지막은 캘리브레이션 정보를 기반으로 모니터 상의 최종적인 응시 위치를 결정하고 이 정보에 의해 게임상의 3차원 뷰 방향을 조정하는 부분으로 구성되어 있다.

실험 결과 본 논문의 방법에 의해 손이 불편한 사용자에게 게임을 즐길 수 있는 기회를 제공하고, 게임 캐릭터와 게임 사용자의 시선 방향을 일치시킴으로서 게임의 흥미와 몰입감을 증가시킬 수 있는 결과를 얻을 수 있었다.

특히 본 논문에서는 노동자의 움직임을 통해 캐릭터의 시선방향을 바꾸는 것을 현대의 PC에서 실행할 수 있을 만큼 저 연산으로 구현하였다. 비교적 적은 비용으로 시선 추적 인터페이스를 구현하고, 이를 통해 흥미롭고 몰입감이 높은 실감나는 게임을 즐길 수 있는 것 또한 큰 장점이다.

그러나, 실험 결과에서 나타났듯이 몇몇의 사용자들은 초기 사용 시 어지러움과 조작의 미숙함을 느끼는 것이 문제점으로 나타났다. 이런 문제점들은 사용자의 반복사용을 통해 개선될 수 있을 것이다.

USB 카메라와 눈의 시선 방향이 이루는 사이 각에 의해 노동자의 상하 움직임을 잘 구별하지 못하는 문제점은 해상도가 높은 영상을 받을 수 있는 USB 카메라를 사용하거나 HMD의 구조적인 문제를 바꾼다면 해결할 수 있을 것으로 기대된다.

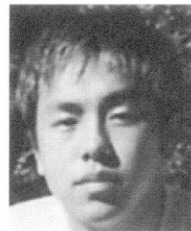
현재는 캐릭터의 시선방향을 움직이는 것만 가능하지만, 동공중심 추출의 정확도를 더욱 높인다면 게임에서 적에게 사격 시 조준을 할 때에도 시선 추적 인터페이스를 사용하여 게임의 흥미를 높이는 것이 가능할 것이다. 또한 다른 장르의 게임에서도 시선 추적 인터페이스를 사용할 수 있을 것으로 기대된다. 단, 정확도를 높이기 위해서는 그만큼 연산량이 증가하여 현대의 PC에서 게임과 동시해 실행하기에 어렵다는 문제점이 생긴다. 이러한 문제점은 두 대의 PC를 사용해서 눈영상의 처리와 게임 실행을 분리하여 추출된 동공중심의 좌표 값을 네트워크를 통해 게임 처리용 PC 상에 전송하는 방법을 사용함으로써 해결할 수 있다.

또한, 눈영상에서 홍채 영역을 추출하여 본인임을 인증하는 기능을 추가한다면 온라인 게임으로 확장 시 사이버머니 결제 및 로그인 시스템으로 활용할 수 있어, 게임 조작과 인증(Authentication)을 동시에 해결할 수 있는 효과를 얻을 수 있을 것이다.

참 고 문 헌

[1] Jacob, R. J. K, "Eye Movement-Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces",

- Advances in Human-Computer Interaction*, H. R. Hartson and D. Hix, Editors., Ablex Publishing Co., pp.151-190, 1993.
- [2] L. E. Sibert, R. J. K. Jacob, "Evaluation of Eye Gaze Interaction", *Proc. of the CHI*, ACM in New York, pp.281-288, 2000.
- [3] Ohno, T. "Quick Menu Selection Task with Eye Mark", *Transactions of Information Processing Society of Japan*, Vol.40, No.2, pp.602-612, 1999.
- [4] Yamato, M., Monden, A., Matsumoto, K., Inoue, K. and Torii, K. "Quick Button Selection with Eye Gazing for General GUI Environments", *International Conference on Software: Theory and Practice*, August, 2000.
- [5] Andrew T. Duchowski, Vinay Shivashankariah, Tim Rawls, Anand K. Gramopadhye, Brian J. Melloy, Barbara Kanki, "Binocular Eye Tracking in Virtual Reality for Inspection Training", ACM Press New York, NY, USA, 2002.
- [6] Jeong Jun Lee, Kang Ryoung Park, Jaihie Kim, "Gaze detection system under HMD environment for user interface", the Joint Conference of ICANN/ICONIP, p.513, June, 2003.
- [7] Anand K. Gramopadhye et al., "Use of Computer Based Training for Aircraft Inspectors: Findings and Recommendations", In Proceedings of the HFES/IEA Annual Meeting (San Diego, CA, August, 2000).
- [8] <http://www.sgi.com>.
- [9] Jie Zhu and Jie Yang, "Subpixel Eye Gaze Tracking", in Proc. Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp.124-129, May, 2002.
- [10] M. R. M. Mimica and C.H. Morimoto, "A Computer Vision Framework for Eye Gaze Tracking", in *Proc. XVI Brazilian Symposium on Computer Graphics and Image Processing*, pp.406-412, Oct., 2003.
- [11] D. H. Yoo, J. H. Kim, B. R. Lee and M. J. Chung, "Non-contact Eye Gaze Tracking System by Mapping of Corneal Reflections", in *Proc. Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pp.101-106, May, 2002.
- [12] J. G. Wang and E. Sung, "Study on Eye Gaze Estimation", *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol.32, No.3, pp.332-350, June, 2002.
- [13] S. W. Shih and J. Liu, "A Novel Approach to 3D Gaze Tracking Using Stereo Cameras", *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol.34, No.1, pp.234-245, Feb., 2004.
- [14] T. Ohno, N. Mukawa and A. Yoshikawa, "FreeGaze: a Gaze Tracking System for Everyday Gaze Interaction", in *Proc. Symposium on Eye Tracking Research and Applications*, pp.125-132, 2002.
- [15] <http://www.serrata.com.au/iglassessvga.html#i-glasses%20SVG%203D>.
- [16] <http://www.alphacams.com>.
- [17] Gregory A. Baxes, "Digital Image Processing-Principles and Application", Wiley.



이 의 철

e-mail : oryong@smu.ac.kr
 2005년 상명대학교 소프트웨어학과(학사)
 2005년~현재 상명대학교 컴퓨터학과
 석사과정
 관심분야: Biometric 영상처리, 컴퓨터 vision



박 강 령

e-mail : parkgr@smu.ac.kr
 1994년 연세대학교 전자공학과(학사)
 1996년 연세대학교 전자공학과(석사)
 2000년 연세대학교 전기·컴퓨터공학과(박사)
 2000년~2003년 LG전자기술원 Digital Vision Group 홍채인식팀
 2003년~현재 상명대학교 소프트웨어대학 미디어학부 조교수
 관심분야: Biometric 영상처리, 패턴인식, 컴퓨터 vision, 컴퓨터 그래픽스