論文2001-38CI-6-5

눈의 응시 방향을 이용한 인간-컴퓨터간의 인터페이스에 관한 연구

(Human-Computer Interface using the Eye-Gaze Direction)

金 度 亨 *, 金 在 憲 *, 鄭 明 振 *

(Do Hyoung Kim, Jea Hean Kim, and Myung Jin Chung)

요 약

본 논문에서는 연속적인 영상 정보와 자기 센서의 정보로부터 실시간으로 눈의 응시 방향을 추적하기 위한 효과적인 접근 방법에 대해 제안한다. 이러한 눈 응시 추적 시스템의 입력으로는 한 대의 카메라를 통해 얻어지는 영상들과 자기 센서의 데이터들이다. 눈의 영상을 받아들이는 카메라와 자기 센서의 수신부가 머리에 고정되어 있어서 측정된 데이터들은 눈과 머리의 움직임을 표현하는데 충분하다. 실험 결과에서는 제안한 시스템이 실시간으로 동작하는 측면에 대해 그 유용성을 보여줄 것이고 또, 그러한 시스템을 이용하여 마우스 대신 새로운 컴퓨터 인터페이스로 사용할 수 있다는 가능성을 보여줄 것이다.

Abstract

In this paper we propose an efficient approach for real-time eye-gaze tracking from image sequence and magnetic sensory information. The inputs to the eye-gaze tracking system are images taken by a camera and data from a magnetic sensor. The measuring data are sufficient to describe the eye and head movement, because the camera and the receiver of a magnetic sensor are stationary with respect to the head. Experimental result shows the validity of real time application aspect of the proposed system and also shows the feasibility of the system as using a new computer interface instead of the mouse.

Ⅰ. 서 론

사람들은 일상 생활에서 다양한 목적을 위해 눈을 사용한다. 대부분의 사람들은 사람의 시각 시스템이 주위를 관측하는 즉, 빛에 의해 반사되는 광학적인 정보를 받아들이고 처리하는 과정만을 수행하는 시스템으로 고려하고 있다. 하지만 눈은 주위를 바라보고 그 정보를 처리하는 입력 기관의 역할만을 수행하는 것이 아니고 눈의 응시 방향을 통해 출력 기관으로써의 역할도 수행하게 된다. 이때 눈이 생성해 내는 출력은 눈의 응시 방향으로 사람이 무엇에 초점을 맞추고 있는지를 의미하며 다시 말하면 사람이 관심 있는 부분이나 물체가 무엇인지를 가리키는 역할을 하는 것이다. 이러한 출력을 생성해 내는 원인은 눈의 생리학적인 측면에서 살펴볼 수 있다. 사람이 물체를 자세하게 관측하기 위해서는 그 물체의 상을 시각 시스템 내부에 존재하는 망막의 중심(fovea) 주위에 위치시켜야 한다는 생리학적인 제한 때문에 눈이 응시하는 물리적인 방향은 사람들이 생각하고 있는 바를 대부분의 경우에

* 正會員, 韓國科學技術員 電子電算學科

(Department of Electrical Engineering and Computer Science, KAIST)

接受日字:2000年8月7日, 수정완료일:2001年10月11日

암시하고 있다고 간주할 수 있다[1].

1936년에 Mowrer가 눈의 회전 정보와 응시 방향을 자동으로 기록하는 장치를 만드는데 성공한 이 후 20세기 동안 이러한 저장 기술들은 점차적으로 향상되어 왔다. 오늘날에는 눈의 응시 방향을 측정하기 위한 여러 가지 방법이 존재한다. 눈의 움직임에 의해 눈 주위의 피부에서 검출되는 매우 작은 전기적 신호를 이용하는 EOG(electro-oculography) 방법과 특수한 콘택트 렌즈를 장착하여 눈의 움직임을 측정하는 방법, 적외선과 같은 외부의 조명을 사용하여 눈동자에서 그 빛의 반사 정도나 눈동자의 여러 가지 막에서 반사되는 위치를 이용한 방법들이 사용되고 있다[1],[2].

이러한 눈의 응시 방향을 추적하는 연구는 그 연구 자체뿐만이 아니라 사용자와 컴퓨터간의 인터페이스를 향상시키는 부분에 대한 연구 방향에도 활기를 불어넣어 왔다. 지금까지는 컴퓨터의 화면을 그래픽으로 표현하거나 운영 체제를 윈도우 시스템 형태로 변화시키는 등의 컴퓨터에게서 사용자에게 보여지는 부분에 대해서 많은 개발과 발전이 있었다. 반면에 사용자의 의도를 컴퓨터에게 표현하는 방법으로는 키보드, 조이스틱, 마우스 등으로 제한되어 있다. 따라서, 사용자의 눈 응시 방향을 추적하는 것을 이용하면 기존의 키보드나 마우스 같은 인터페이스에서 벗어나 사용자와 컴퓨터 간의 상호 통신의 대역폭을 증가시킬 수 있다. 예를 들면, 사용자의 눈의 움직임으로 나타나는 의도를 컴퓨터가 살펴보고 그에 따른 반응을 보이게 할 수 있다. 이로써 컴퓨터가 단순한 도구와 같은 것이 아니라 사용자를 관찰하는 중개인으로 고려될 것이다.

본 논문에서는 앞에서 살펴본 바와 같이 눈의 응시 방향을 효과적인 방법으로 찾아내고 그 정보를 이용하여 컴퓨터 인터페이스로의 활용이 가능함을 실험을 통해 살펴보도록 한다. 논문의 구성은 2장에서는 전체적인 눈 응시 방향 추적 시스템에 대해 설명하고, 3장에서는 눈동자의 움직임을 추출하고 추적하는 방법에 대해 제안한다. 그리고 4장에서는 본 논문에서 사용한 머리의 움직임을 추적하는 방법에 대해 설명하고 5장에서는 두 가지의 움직임을 통합하는데 필요한 여러 가지 교정 과정에 대해 설명한 후 6장에서 제안한 눈 응시 추적 시스템을 이용하여 실험 및 성능평가를 하고 끝으로 결론을 맺는다.

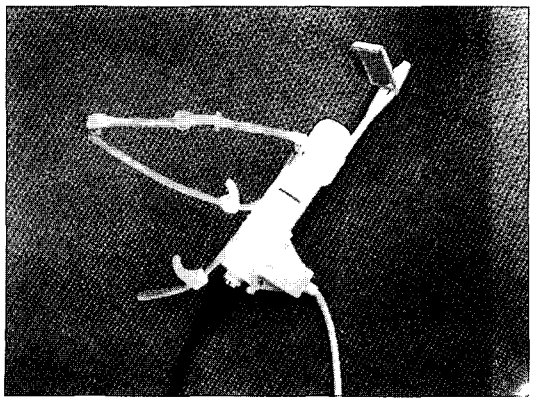


그림 1. 눈 응시 방향 추적 시스템
Fig. 1. An eye-gaze tracking system.

Ⅱ. 눈 응시 방향 추적 시스템

본 논문에서는 머리의 움직임을 고려한 눈의 응시 방향을 알아내기 위해서 머리의 움직임과 눈의 움직임을 모두 찾아내어야만 한다. 하지만 머리의 움직임과 눈의 움직임은 서로 상관 관계를 가지고 있다. 그래서 본 연구에서는 두 개의 움직임을 서로 독립적으로 찾아내기 위해서 머리에 장착하는 형태인 시스템을 디자인하여 눈의 움직임과 머리의 움직임을 각각 추적하도록 하였다.

그림 1은 눈 응시 방향을 추적하기 위해 디자인된 전체 시스템을 보여주고 있다. 그림에서 보여지는 안경 구조물 옆에 부착된 주조물은 총 길이가 17㎝이고 그 앞부분에 거울이 부착되어 있는 형태이고 그 안에 JAI에서 판매하고 있는 마이크로 헤드용 CCD 카메라가 장착되어 있다. 그리고, 그 위에 부착되어 있는 육면체 모양의 센서는 머리의 움직임을 추적하기 위한 자기 센서의 수신부로 Acsension Technology에서 판매하고 있는 FOB(Flock of Birds)[3]를 사용하고 있다.

Ⅲ. 눈의 움직임 추적

본 연구에서는 앞에서 언급한 바와 같이 눈의 움직임을 추적하기 위해서 CCD 카메라를 사용하여 눈의 움직임에 대한 영상을 연속적으로 받아들이고 그 영상을 처리하여 눈의 움직임에 대한 정보를 추출하고자 한다. 여기에서는 기존의 홍채의 경계 추적 방법의 문제점들인 눈꺼풀에 의해 가려지는 홍채의 경계 때문에 수직인 방향의 움직임을 찾아낼 수 없는 점과 조명에 강인하게 하기 위해 적외선을 사용하여 시스템 가격이

비싸게 되는 점들을 해결하기 위해서 보통 조명 환경하에서 조명 변화에 강인하고 홍채 경계를 이용하여 수직인 방향에 대한 움직임을 찾아내도록 보완하여 눈의 움직임을 추적하는 방법을 제안하였다[7]. 이는 눈꺼풀에 의해 가려지는 홍채의 경계 부분까지 추출하기 위해 Xie가 제안하였던 눈의 특징들을 찾아내는 알고리즘을 본 연구에 맞게 수정된 것이다[11]. 그림 2는 여기에서 제안한 알고리즘의 순서도 이다.

본 논문에서 제안한 알고리즘은 다음과 같다.

(1) 홍채의 경계를 원으로 가정하고 그것을 홍채의 경계 모델이라 설정한다.

(2) 카메라를 통해 얻어지는 연속적인 눈의 영상으로부터 3개의 기본 영상을 연속적으로 추출한다.

(3) 설정된 모델과 얻어진 영상과의 일치되는 정도를 나타내는 기준들을 매칭 함수라 설정하고 그 함수를 최대화하는 방향으로 홍채의 경계 모델의 위치를 검색해 나간다.

(4) (3)번의 과정을 반복하면서 홍채의 경계 모델의 최적인 위치를 찾아낸다.

(5) 찾아낸 홍채의 경계 모델의 위치로부터 눈동자의 움직인 각도를 측정한다.

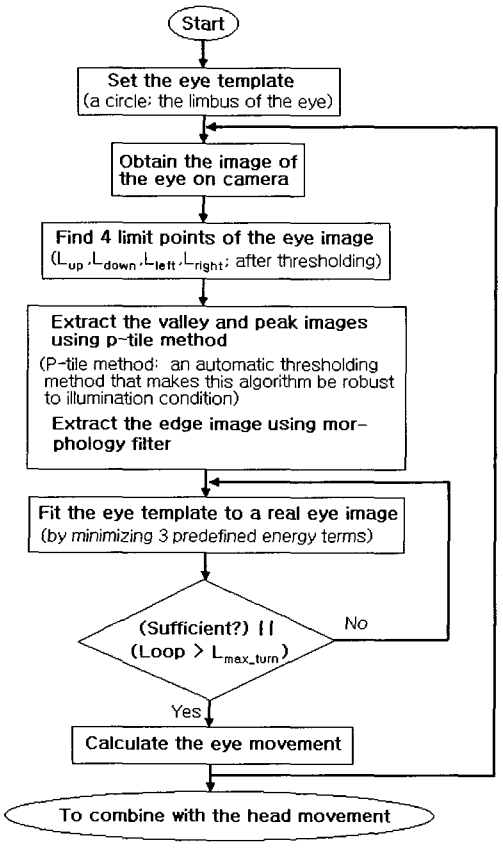


그림 2. 제안한 홍채 경계 추적 알고리즘의 순서도
Fig. 2. A flow chart of the proposed *limbus tracking algorithm(*limbus : the boundary between the dark iris and the white sclera of the eye).

1. 기본 영상 추출 과정

본 논문에서는 카메라를 통해 얻어지는 눈의 영상에서 3개의 기본 영상을 추출한다. 이는 눈의 특징을 가장 잘 표현하는 영상으로 눈의 특징을 고려하여 다음과 같이 선택하였다. 여기에서 구분하고자 하는 홍채의 경계는 눈동자에서 밝은 부분과 어두운 부분을 나누는 경계이다. 그러므로, 카메라를 통해 얻어지는 눈의 영상의 밝은 부분과 어두운 부분을 표현할 기본 영상이 필요하게 된다[10],[11]. 본 논문에서는 광량에 의해 영상의 밝고 어둡게 표현되는 부분을 구분하기 위한 영상으로 2 가지의 기본 영상을 선택하였다. 어두운 부분일수록 높은 값을 가지는 기본 영상을 valley image, 밝은 부분일수록 높은 값을 가지는 영상을 peak image로 고려하여 눈의 특징을 표현하였다. 이러한 valley image와 peak image를 우리가 관심 있는 부분의 구분을 위해서 임계값을 정하여 이 진화된 영상으로 표현하였다. 이러한 임계값을 보통의 조명 환경에서 조명 변화에 강인하도록 설정하기 위해서 자동으로 임계값을 설정하는 p-tile 방법을 적용하였다. 이러한 p-tile 방법은 구분하고자 하는 물체의 영역이나 크기를 알고 있을 때 사용하는 방법이다.

그리고, 관심 있는 부분인 홍채의 가장자리 부분에서 영상의 밝기에 의해 발생되는 영상의 가장자리 부분을 edge image라고 설정하여 이를 추가함으로써 눈의 홍채 경계에 대한 정보를 가장 잘 나타내는 3가지의 기본 영상을 선택하였다. 이 때 홍채의 가장자리 정보를 추출하기 위해서 홍채 부분의 영역을 획득한 후 morphology filter를 이용하였고, 3가지의 기본 영상을 얻는 과정에서 불필요한 영역을 제거하는 처리 과정도 필요하게 되는데 이때 사용된 알고리즘은 영상 분할과 구분에서 많이 사용되는 4 방향에 대한 연속성 정보를 고려한 blob coloring 알고리즘을 사용하였다.

이러한 여러 가지 영상 처리 과정을 통해 얻어진 눈에 대한 3가지의 기본 영상은 그림 3과 같다. 그림 3의 a)는 눈동자의 밝은 부분을 나타내는 이진화된 peak image에 해당하고 b)는 어두운 부분을 나타내는 valley

(a) (b) (c)

그림 3. 눈동자의 3가지 기본 영상 (a) peak image, b) valley image, c) edge image)
Fig. 3. Three basis images of the eye (a) peak image, b) valley image, c) edge image).

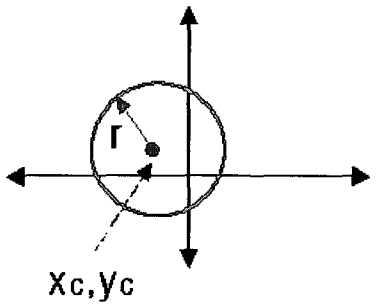


그림 4. 홍채 경계 모델
Fig. 4. Iris boundary(limbus) model.

image c)는 눈동자의 가장자리 부분을 나타내는 edge image에 해당한다.

2. 홍채의 경계 모델 설정

우리는 일반적으로 관찰하기에는 홍채의 경계가 원으로 관측되는 것을 쉽게 알 수 있다. 그래서, 본 논문에서는 눈의 홍채 가장자리 부분을 원으로 고려하는 것에 대한 오차는 눈동자가 움직임에 따라 찌그러지는 오차에 비해서 매우 작으므로 무시하기로 한다. 실제로 Haslwanter와 Moore는 눈의 움직임과 눈동자의 경계를 강인하게 찾아내는 방법에 대해 연구하는 과정 중 홍채의 경계를 원으로 모델링에 의한 오차는 매우 작다고 언급한 바 있다[4].

그러므로 눈의 홍채 가장자리에 대한 영역을 원으로 모델링하고 원을 나타내기 위해 필요한 3개의 변수인 원의 중심점 위치에 대한 좌표값(X_c, Y_c)와 그 원의 반지름(r)을 찾아내면 된다. 그림 4는 설정한 홍채 경계의 모델을 나타낸다.

3. Model matching

본 논문에서는 위에서 언급한 현재의 입력 영상에서 홍채의 경계를 가장 잘 표현하는 기본 영상 3개와 홍채 가장자리를 원으로 근사화하여 모델링 한 홍채 가장자리 모델인 원과 일치시키기 위해서 3가지의 매칭 함수(matching function)를 설정하였다. 여기에서 설정된 매칭 함수는 다음과 같다.

(1) Valley image에 대한 매칭 함수는 Mv라고 표시하며 다음과 같이 나타낼 수 있다.

$$M_v = \sum_{i \in circle} valley\_image_i \quad (1)$$

여기에서 valley_image는 앞에서 언급한 이진화된 valley image를 나타낸다. 그리고 circle은 홍채의 경계 모델의 내부를 나타낸다.

(2) Peak image에 대한 매칭 함수는 Mp라고 표시하며 다음과 같이 나타낼 수 있다.

$$M_p = -\sum_{i \in circle} peak\_image_i \quad (2)$$

여기에서 peak_image는 앞에서 언급한 이진화된 peak image를 나타낸다.

(3) Edge image에 대한 매칭 함수는 Me라고 표시하며 다음과 같이 나타낼 수 있다.

$$M_e = \sum_{i \in boundary} edge\_image_i \quad (3)$$

여기에서 edge_image는 앞에서 언급한 이진화된 edge image를 나타낸다. 그리고 boundary는 홍채 모델의 가장자리 부분을 나타낸다.

이러한 3가지의 매칭 함수를 이용하여 최종적인 매칭 함수는 각 매칭 함수에 가중치를 고려하여 합한 것으로 나타내어진다. 이를 M_{total}이라고 나타내고 다음과 같이 표현한다.

$$M_{total} = W_v M_v + W_p M_p + W_e M_e \quad (4)$$

위 식에서 사용된 각 가중치들은 heuristic한 방법으로 결정된다. 최종적으로 결정되는 가중치의 특성을 살펴보면, W_v, W_p는 크게 설정되고 W_e는 작게 설정된다. 이러한 현상은 edge image의 부정확성에 기인한 것이다. 실제로 사용된 가중치 값들은 W_v, W_p는 거의 1:1 정도의 비율로 비슷하게 값을 가지고, W_e는 0.1배 정도의 매우 작은 값을 가지게 된다.

식 (4)에서 표현된 매칭 함수들에 의해 제대로 모델이 원하는 방향으로 수렴해 가는지를 알아보기 위해서 앞에서 설정한 매칭 함수들의 특성들을 여기에서 살펴보기로 하자. M_v는 이진화된 valley image와 모델과의

매칭에서 홍채 가장자리 모델이 눈에서 가장 어두운 부분인 동공과 홍채를 나타내는 부분을 가장 많이 포함하는 곳으로 위치시키는 역할을 하고, M_p는 홍채 가장자리의 모델이 눈의 밝은 영역인 흰 공막을 포함하지 않는 곳으로 모델을 위치시키도록 하는 특성을 가지고 있다. 그리고, M_e는 홍채의 가장자리와 모델의 가장자리가 가장 잘 일치되는 곳으로 모델을 위치시키는 역할을 하게 된다. 이러한 세 개의 매칭 함수에 의해 홍채의 가장 자리 모델은 눈의 영상에서 실제의 홍채 경계로 일치되도록 이동하게 된다.

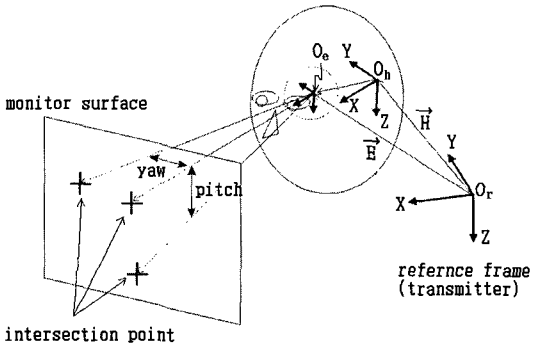


그림 5. 눈의 응시 방향 계산 방법
Fig. 5. The calculation of the eye gaze direction.

Ⅳ. 머리의 움직임 추적

본 논문에서는 머리의 움직임의 6 자유도(position, orientation)를 측정하기 위해서 자기 센서를 도입하여 사용하였다[7]. 여기에서 사용되는 자기 센서는 크게 두 부분으로 나뉘어 지는데 하나는 자기장을 형성하는 전송부(transmitter)와 그 자장을 감지하는 수신부(receiver)로 구성되어 있다. 자기 센서의 수신부는 작고 가벼워서 쉽게 눈의 움직임을 추적하기 위한 시스템 부분과 접목시킬 수 있다. 그리고, 자기 센서는 전송부에서 발생되는 자기장을 수신부에서 감지하여 그것을 처리한 다음 3차원의 위치 정보와 3차원의 방향 정보(azimuth, elevation, roll)를 제공하게 된다. 이 센서의 측정 정밀도는 저주파 대역의 EMI(Electro-magnetic interference)와 자기장 안의 큰 금속 물체와 같은 것에 의해 영향을 받는다. 이 센서의 성능을 제대로 유지하기 위해서는 전송부와 수신부가 CRT 모니터나 전압 공급기와 같은 EMI 소스에 대해서 어느 정도 거리를 유지해야 한다. 하지만, 요구되는 거리가 아주 먼 것이 아니기 때문에 컴퓨터의 인터페이스로 쓰기엔 무관하다.

머리 추적 시스템의 측정 범위를 살펴보면 병진 운동의 범위는 세 방향에 대해 91cm까지 가능하고 회전 운동에서 azimuth와 roll 방향은 ±180° 까지 가능하고 elevation 운동은 ±90° 범위까지 가능하다. 따라서 자기 센서 수신부의 위치가 전송부의 24cm 범위 안에 있으면 이 시스템에서 제공하는 정밀도는 회전 운동에 대해서 0.1rms를 넘지 않고, 병진 운동에 대해서는 최대 0.75mm 정도가 된다. 반면에 수신부의 위치가 24cm의 범위 안에 있지 않고 48cm 정도 떨어지게 되면 azimuth 운동에 대한 정밀도만 살펴보더라도 0.75rms가 되고 60cm 정도 떨어지게 되면 1.78rms로 정밀도가 많이 떨어지게 된다[3],[5].

Ⅴ. 눈의 응시 방향 계산을 위한 여러 교정(calibration) 과정

눈의 응시 방향은 3장과 4장에서 언급한 눈의 움직임과 머리의 움직임을 추적하여 얻어낸 데이터를 통합하여 최종적으로 계산해야 한다[7]. 그림 5는 눈의 응시 방향을 계산하는 방법에 대해 개략적으로 보여주고 있다. 본 연구에서 눈의 응시 방향을 찾는 알고리즘은 다음과 같은 단계들로 이루어져 있다.

(1) 자기 센서의 수신부를 이용하여 머리의 자세를 나타내는 정보를 얻는다.
(2) 기구부의 구조를 통해 눈동자의 회전 중심점을 찾는다.
(3) 눈동자가 움직이는 회전각도(pitch, yaw)를 카메라로부터 찾아낸다. 이때, 영상의 좌표와 눈동자의 회전 중심 좌표간의 교정 과정이 필요하다.
(4) 모니터의 좌표를 기준으로 눈동자의 바라보는 방향 벡터와 시작점을 위에서 구한 회전각도 정보로부터 계산한다. 이때 모니터와 자기 센서의 전송부 좌표간의 변환 관계에 대한 교정 과정이 필요하다.
(5) 구해진 방향 벡터와 시작점을 이용하여 그 벡터 방향과 모니터와의 만나는 한 점을 구한다.

1. 눈의 회전각도 계산 방법

앞의 계산 방법 중에 눈의 회전각도를 계산하는 방법은 좀 더 고려해 보아야할 필요가 있다. 앞에서도 언급하였지만 카메라로부터 얻어지는 영상의 좌표를 기준으로 하는 눈의 움직임과 실제로 눈동자의 회전 중

심을 원점으로 가지는 3차원 좌표간의 교정 과정이 필요하게 된다[6]. 따라서, 머리의 움직임에 관련된 눈의 회전각도(pitch, yaw)를 계산하기 위해서 얻어진 눈동자의 중심을 사용하게 된다. 그러한 눈동자의 중심을 이용하여 눈의 회전 각도를 계산하는 방법의 과정은 다음과 같다.

(1) 눈동자의 중심을 P로 나타낸다.

(2) 우선 눈동자의 회전을 고려하지 않은 경우를 이용하여 눈동자의 회전 좌표의 원점을 눈의 영상에서 찾아내고, 그 원점으로부터 X, Y축에 존재하는 4가지 방향의 단위 벡터를 계산하고 눈동자 중심점의 위치 벡터(P)도 찾아낸다. 그 다음으로 눈동자의 회전의 제한 각도에 따라 표시되는 각 경계들을 찾아낸다.(그림 6)

(3) 찾아낸 눈동자 중심점(P)이 설정된 회전의 제한 각도를 벗어난 위치이면 가장 가까운 경계 위의 한 점에 위치시키고, 그 범위 안에 중심점(P)이 존재하면 4가지 방향의 단위 벡터의 합으로 P 벡터를 표시하여 눈동자의 움직인 각도(pitch, yaw)를 구한다.

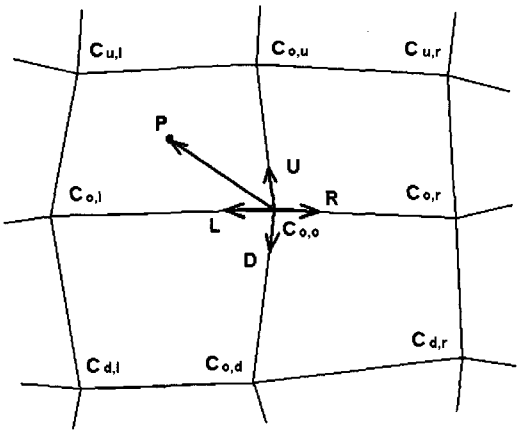


그림 6. 눈동자의 움직임 계산을 위한 교정 맵
Fig. 6. The calibration map for calculating the eye movement.

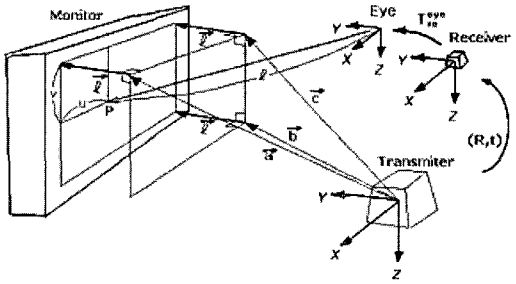


그림 7. 모니터 좌표와 자기 센서의 좌표 사이의 변환 관계 규명을 위한 교정
Fig. 7. The calibration to identify a transformation between the monitor coordinate and the magnetic sensor coordinate.

2. 모니터 좌표와 변환 관계 규명을 위한 교정 과정

사용자의 시선 방향을 검출하기 위해서 필요한 또 한가지의 교정과정은 모니터의 3차원 평면을 나타내는 기준 좌표계와 눈 응시 추적 시스템의 끝 부분에 부착되어 머리의 움직임을 나타내는 자기 센서 수신부의 좌표계간의 변환 관계를 규명해야 하는 과정이다.

이러한 측정의 과정을 수행하기 위해서는 우선 자기 센서의 전송부와 모니터 사이의 상대적인 위치와 방향 관계를 알아내는 교정 과정이 필요하다. 그림 7은 그러한 모니터와 자기 센서의 전송부의 좌표간의 변환 관계를 규명하기 위한 상관 관계를 나타내고 있다. 상관 관계를 알기 위해서 그림 7에서 나타나는 것과 같이 자기 센서의 수신부를 모니터의 모서리 중 3곳에 순서대로 배치시킨 다음 센서의 전송부의 3차원 좌표를 기준으로 모니터의 평면을 나타내는 3곳의 좌표 값을 알아낼 수 있게 된다. 그 값들을 $\vec{a}$, $\vec{b}$, $\vec{c}$로 표현하기로 한다. 여기에서 구한 벡터 $\vec{a}$, $\vec{b}$, $\vec{c}$를 통해 자기 센서의 수신부의 좌표를 기준으로 모니터의 위치를 나타내는 하나의 평면을 결정할 수 있다. 즉, 모니터 평면을 자기 센서의 전송부의 좌표를 기준으로 원점의 위치 벡터와 모니터 평면의 u방향, v방향의 단위 벡터로 표현할 수 있다. 자기 센서의 전송부 좌표를 기준으로 모니터의 평면의 원점을 나타내는 벡터를 $\overrightarrow{O_{mt}}$로 나타내고 모니터 평면의 u, v 방향의 단위 벡터를 $\vec{a_u}$, $\vec{a_v}$로 표현하기로 하자. 그리고, 눈동자 회전 좌표의 원점을 $\overrightarrow{O_{eye}}$로 표현했을 때, 눈 응시에 의해 표현되는 임의의 방향 벡터, $\vec{d}$,와 모니터 평면과의 만나는 위치(u, v)를 모니터 평면의 원점을 기준으로 구하면 식 (5)와 같이 표현할 수 있다.

$$\begin{bmatrix} u \\ v \\ l \end{bmatrix} = [\ \vec{a_u}\ \ \vec{a_v}\ \ -\vec{d}\ ]^{-1} \cdot [\ \overrightarrow{O_{eye}}\ \ -\overrightarrow{O_{mt}}\ ] \quad (5)$$

식 (5)는 다음과 같은 과정에서 쉽게 구할 수 있다. 그림 7에서 살펴볼 수 있듯이 모니터 상의 한 점 $P=[x_p\ y_p\ z_p]^T$를 두 가지 형태로 표현하여 같다고 나

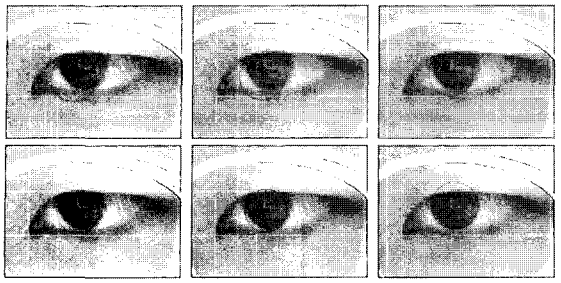


그림 8. 홍채 경계 모델 매칭 순서(중심점)
Fig. 8. A sequence of the iris boundary(limbus) model matching(the center point).

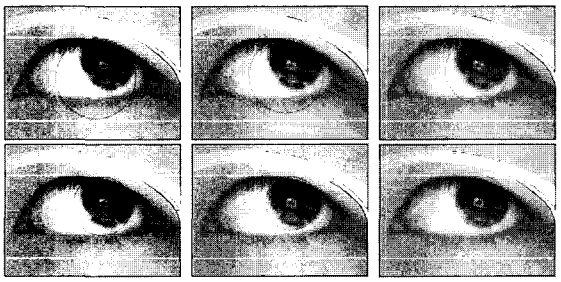


그림 9. 홍채 경계 모델 매칭 순서(반지름)
Fig. 9. A sequence of the iris boundary(limbus) model matching(the radius).

타내고 정리하면 식 (6), (7)과 같이 된다.

$$\overrightarrow{O_{mt}} + u \cdot \overrightarrow{a_u} + v \cdot \overrightarrow{a_v} = \overrightarrow{O_{eye}} + l \cdot \vec{d} \quad (6)$$

$$[\ \overrightarrow{a_u}\ \ \overrightarrow{a_v}\ \ -\vec{d}\ ] \cdot \begin{bmatrix} u \\ v \\ l \end{bmatrix} = [\ \overrightarrow{O_{eye}} - \overrightarrow{O_{mt}}\ ] \quad (7)$$

따라서, 식 (7)에서 $[u\ v\ l]^T$를 좌변으로 하고 정리하면 간단히 식 (5)가 유도됨을 알 수 있다.

모니터 좌표와의 교정과정이 선행된 상태에서 앞에서 언급한 알고리즘을 수행하여 눈의 응시 방향 벡터를 찾아내면 식 (5)에서 눈의 응시 방향과 모니터 평면과의 교점을 계산해 낼 수 있다.

VI. 실험 및 결과

본 논문에서 제안된 알고리즘은 Intel PentiumIII 600MHz 프로세서를 탑재한 PC(personal computer)와 Matrox에서 상용화되어 판매되는 Meteor image grabber를 사용하여 연속적인 영상을 획득하였다. Image grabber는 눈의 실제 영상을 계속해서 얻어내어 그 영상을 가지고 컴퓨터를 이용하여 제안된 알고리즘을 수행하게 된다. 알고리즘을 구현한 프로그램은 IBM 호환 컴퓨터에서 Visual C++ 6.0을 사용하여 제작하였다.

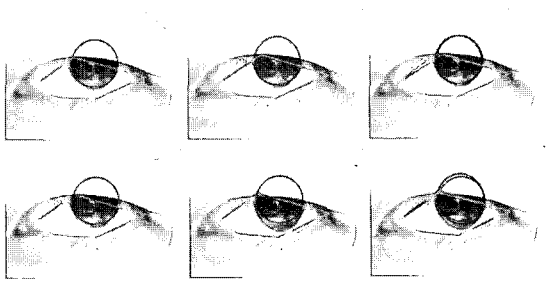


그림 10. 연속적인 영상에 대해 홍채 경계 모델 매칭 첫 번째 실험 결과
Fig. 10. The first experimental result of the iris boundary (limbus) model matching in case of the sequential image.

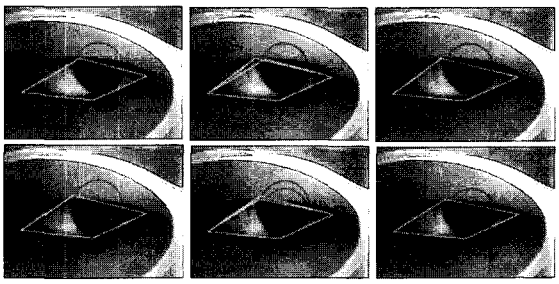


그림 11. 연속적인 영상에 대해 홍채 경계 모델 매칭 두 번째 실험 결과
Fig. 11. The second experimental result of the iris boundary(limbus) model matching in case of the sequential image.

1. 눈의 움직임 추적에 대한 실험

우선 앞에서 제안한 눈의 움직임 추적 알고리즘에 대한 수렴성과 강인성에 대해 살펴보기 위해서 정지된 눈의 영상을 가지고 타당성을 검증해 보았다. 그림 8과 그림 9는 임의의 눈동자의 영상에 제안한 알고리즘을 적용하여 홍채 경계 모델인 원과 매칭시킨 결과이다. 모델의 임의의 초기 위치에서 우리가 찾고자 하는 홍채 경계를 제대로 찾아가는 과정을 보여주고 있다. 좀 더 제안한 알고리즘에 대한 타당성을 살펴보기 위해 여러 장의 눈의 영상을 얻어서 실험해 보았다. 표 1은 40가지의 경우를 고려해서 홍채 경계 모델이 매칭될 때 발생하는 오차를 살펴본 것이다. 표 1에서 살펴볼 수 있듯이 중심점의 X축에 대한 오차는 평균 4픽셀 정도로 약 0.4° 의 X축 방향에 대한 회전 오차를 가지고 있고, Y축에 대한 오차는 평균 12픽셀로 약 2.4° 정도

의 Y 방향에 대한 회전 오차가 있다는 의미가 된다.

앞에서 살펴본 결과는 정지된 영상에 대해 적용한 실험 결과였다. 지금부터는 눈의 위치가 계속하여 변하는 여러 장의 연속적인 영상에서도 제안한 알고리즘이 타당성이 있는지에 대해 살펴보자. 그림 10과 그림 11은 움직이는 눈동자를 연속적으로 CCD 카메라에서 영상으로 획득하여 그 연속적인 영상에 대해 제안한 알고리즘을 적용한 실험 결과들이다. 그림 10은 대상이 되는 사용자 앞에 조명을 비추어 밝은 상태에서 적용한 실험 결과이고 그림 11은 그 조명을 제거하여 어두운 상태에서 적용한 실험 결과이다. 그림 10과 그림 11에서 살펴볼 수 있듯이 제안한 알고리즘은 움직이는 눈의 영상에 대한 적용에서도 그 타당성이 검증되었으며 조명의 변화에 강인함을 보여주고 있다.

여기에서는 실시간으로 동작이 가능한 시스템 구현

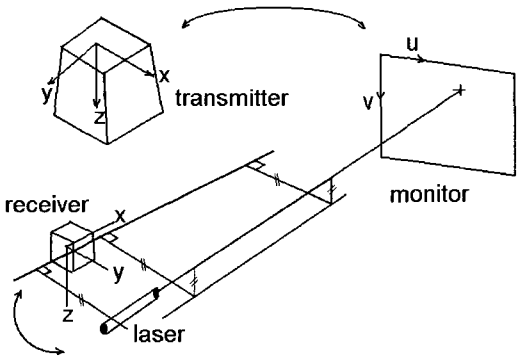


그림 12. 자기 센서 정밀도 측정을 위한 실험 환경
Fig. 12. Experimental environment to measure the resolution of the magnetic sensor.

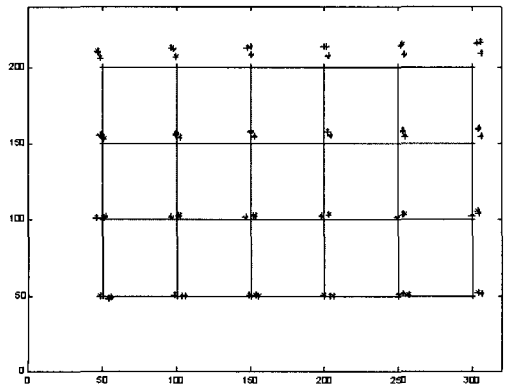


그림 13. 자기 센서 정밀도 실험 결과
Fig. 13. Experimental result for the resolution of the magnetic sensor.

에 그 목적이 있으므로 이러한 연속적인 영상에 대한 실험 결과에서 알고리즘의 수행시간 또한 고려해 보아야 한다. 제안한 알고리즘의 수행시간은 320×240 크기의 영상에 대해 110ms 정도가 소요된다. 이는 눈의 움직임이 초당 약 30° (deg)인 경우의 움직임까지 처리될 수 있고 눈의 움직임이 더 빠른 경우에는 여러 번의 적용에 의해 단계적으로 찾아나가게 된다. 하지만 앞에서 살펴본 결과에서 알 수 있듯이 눈이 30 (deg/sec) 이내의 속도로 움직일 경우에는 160ms 이상 한 위치에 고정되어 있어야만 하고 그 이상일 경우에는 눈동자의 움직임을 알고리즘이 추적하는 시간까지 고정되어 있어야 눈의 위치 검출이 타당하다고 할 수 있다.

2. 머리의 움직임 추적에 대한 실험

머리 움직임을 추적하는 시스템의 에러는 여기에서 사용하는 자기 센서의 정밀도와 같다. 자기 센서의 정밀도를 측정하기 위해서 그림 12에서 나타나는 것과 같이 자기 센서의 수신부와 레이저 포인터를 서로 연결하여 수신부의 움직임에 따라 레이저 포인터가 모니터에 지시하는 점들을 표시하여 그 정밀도를 분석하였다. 물론 이러한 실험은 교정과정이나 기구부에서 발생되는 오차도 첨가가 될 것이다. 하지만, 실제로 머리의 움직임을 측정할 때도 이러한 과정을 포함하므로 그 오차는 머리의 움직임을 측정할 때 발생되는 오차로 간주할 수 있다. 그림 13은 미리 그려진 격자의 각 교차점들을 레이저 포인터로 지시하게 한 다음 자기 센서의 데이터를 가지고 계산한 위치를 십자가 모양의 + 표시로 나타낸 것이다. 이 실험은 모니터에서 자기센서의 수신부의 위치가 40cm 떨어진 곳에서 수행한 실험이며 이를 여러 번 되풀이하였다. 그 실험 오차에 대한 분석 결과는 표 2에 나타내었다. 이는 자기 센서의 회전에 대한 움직임과 이동에 대한 움직임을 동시에 발생시켜 측정한 실험이므로 이 두 가지 운동에서 발생될 수 있는 에러가 종합적으로 표현된 오차 분포이다.

표 2. 자기 센서 정밀도 실험 결과
Table 2. Experimental result for the resolution of the magnetic sensor.

(단위 : mm)	최소 오차	최대 오차	평균	편차
u	0.08	6.69	2.52	1.93
v	0.02	8.88	3.72	2.76

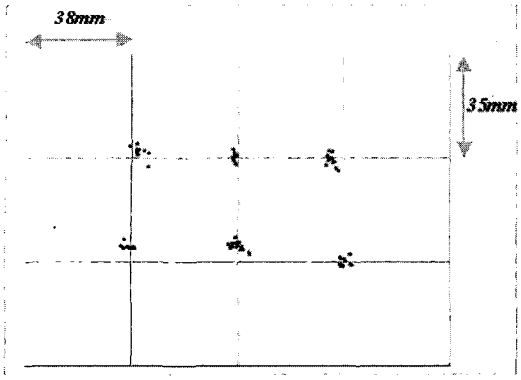


그림 14. 눈 응시 추적 시스템 정밀도 실험 결과
Fig. 14. Experimental result for the resolution of an eye gaze tracking system.

3. 검출된 눈의 응시 방향 실험 결과

앞의 두 가지 실험에서 살펴본 것은 눈의 응시 방향을 검출하기 위해 눈의 움직임과 머리의 움직임에 대해 각각 실험하여 그 결과를 분석한 것이다. 여기에서는 머리와 눈의 움직임을 모두 고려하여 최종적인 눈의 응시 방향에 대한 실험을 하고 그 결과에 대해 분석하고자 한다. 우선 눈의 응시 방향의 정밀도를 계산하기 위해서 사용자가 눈의 응시 방향 추적 시스템을 장착하고 모니터와 약 50cm 정도 떨어진 거리에서 모니터 화면에 그려진 격자의 교차점을 사용자가 바라보도록 한 후 제안한 알고리즘을 통해 계산된 눈의 응시 방향을 화면에 표시하고 이를 비교하여 그 정밀도를 분석하고자 하였다. 그림 14는 그 실험 결과 화면을 보여주고 있고 눈의 응시 방향의 오차에 대한 분석은 표 3에서 나타내고 있다.

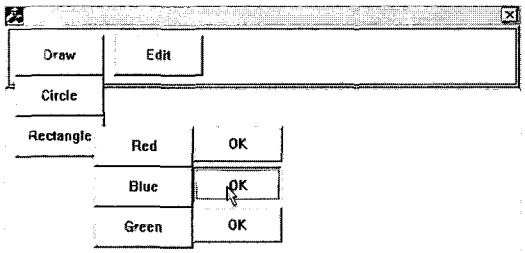


그림 15. 눈 응시 방향을 이용한 메뉴 선택
Fig. 15. The menu selection using eye gaze direction.

그림 16. 눈 응시 방향을 이용한 pan-tilt 제어
Fig. 16. Control of a pan-tilt unit using eye gaze direction.

표 3. 눈 응시 추적 시스템 정밀도 실험 결과
Table 3. Experimental result for the resolution of an eye gaze tracking system.

(단위: mm)	최소 오차	최대 오차	평균	편차
u	0.28	6.75	0.68	8.1
v	0.27	8.12	2.45	9.3

표 2에서 나타내는 자기 센서 정밀도와 비교해 보면, 표 3에서 살펴볼 수 있듯이 최종적인 눈 응시 추적 시스템의 정밀도는 최대 오차와 평균은 자기 센서의 정밀도와 비슷하거나 더 높고 최소 오차와 편차는 더 낮은 것을 알 수 있다. 즉, 눈 응시 추적 시스템이 자기 센서만을 가지고 실험한 것보다 정확도는 높고 정밀도는 낮은 것을 알 수 있다. 이러한 실험 결과에서 정확도는 교정 과정을 통해 충분히 향상시킬 수 있다는 점을 감안하면 눈 응시 추적 시스템이 자기 센서만을 사용한 경우에 비해 안 좋은 결과를 보여주고 있다. 이는 눈의 움직임을 추적하는 과정에서 발생되는 오차와 눈 응시 추적 시스템의 기구부에 의해 발생되는 오차가 첨가되어 이러한 결과가 발생함을 알 수 있다.

4. 눈의 응시 방향을 이용한 인터페이스 활용 실험

본 논문에서 고안한 눈 응시 추적 시스템을 이용하여 사람과 컴퓨터간의 인터페이스 역할이 가능한지를 살펴보기 위해 다음과 같은 실험을 하였다. 우선 적절한 크기의 메뉴를 프로그램으로 제작하여 눈의 응시 방향을 이용한 메뉴 선택 실험을 해 보았다. 이 실험은 눈 응시 방향을 이용하여 컴퓨터의 마우스 포인터가

눈이 바라보는 방향으로 이동하고 마우스의 클릭 명령을 사용하지 않고 마우스의 위치만으로 그 메뉴를 선택하고 메뉴에서 선택되어진 명령이 수행되도록 풀다운 메뉴 방식의 프로그램을 사용하였다. 이는 눈 응시 방향을 이용하여 컴퓨터의 마우스 포인터를 이동시킴으로써 사람과 컴퓨터간의 인터페이스로의 활용 가능성이 충분히 있음을 보여주고 있다. 그림 15는 그 실행 프로그램 화면을 나타내고 있다.

또한, 이러한 메뉴 방식의 프로그램을 이용하여 인간과 기계간의 인터페이스로도 활용될 수 있는지를 검토하기 위해서 PC에서 직렬통신으로 명령을 내려 제어하는 pan-tilt 유닛을 제어해 보았다. 사람과 PC 간의 인터페이스는 눈 응시 방향과 pan-tilt를 제어하기 위한 프로그램으로 구성된다. 이를 이용하여 최종적으로 눈의 응시 방향을 이용하여 pan-tilt에 장착된 카메라에서 보여주는 화면을 사용자가 바라보면서 사용자가 보고자 하는 방향으로 pan-tilt를 제어함으로써 원하는 방향을 관측할 수 있는 실험을 하였다. 그림 16은 그 실험을 수행하는 모습을 보여주고 있다.

Ⅶ. 결 론

본 논문에서는 앞의 실험 결과에서 살펴본 바와 같이 실시간으로 눈의 응시 방향을 추적하는 시스템을 고안하였다. 이러한 시스템 개발을 위해 본 논문에서는 눈의 움직임을 연속적으로 추적하는 알고리즘을 제안하였고 그 알고리즘을 구현하였다. 그리고, 머리의 움직임을 측정하기 위해 자기 센서를 도입하여 그 센서를 사용함으로써 3차원의 위치와 회전에 대한 정보를 획득하여 그 정보를 가지고 머리의 움직임을 추적하였다. 고안한 시스템은 머리에 장착하는 형태이므로 자기 센서에서 제공하는 영역인 자기센서의 전송부를 중심으로 약 $8m^3$의 삼차원 정육면체 안에서 머리의 움직임에 제한 받지 않고 눈의 응시 방향을 찾아낼 수 있다.

또한, 본 연구에서 제안한 눈 응시 추적 시스템을 착용하고 클릭을 사용하지 않고 마우스의 위치만으로 메뉴를 선정하고 선택할 수 있는 인터페이스 화면을 이용하여 PC와 사용자간의 상호 작용 역할을 충분히 해낼 수 있음을 검증하였고, 이러한 사용자와 PC와의 인터페이스의 활용 범위를 넓혀 카메라를 장착한 pan-tilt 유닛을 눈의 응시 방향만으로 제어함으로써 인간과 기계간의 상호 작용에 까지 확장 가능함을 실험을 통해 살펴보았다.

그리고, 본 연구에서는 처리시간의 최소화와 변화하는 환경에서 강인성을 갖는 것에 주안점을 두었다. 그럼에도 이 연구의 선급과제는 처리 시간의 단축임을 언급하고자 한다. 추후에는 눈의 응시 방향을 추출하는데 있어서 정밀성과 정확성의 증가도 고려되어야만 한다. 정밀성이나 정확성을 증가시키기 위해서는 크게 두 가지를 고려해야 한다. 하나는 실험적으로 발생되는 오차를 감소시켜야 하고 나머지는 여기에서 제안된 알고리즘과 시스템 자체의 정확성, 정밀성을 증가시켜야 한다.

참 고 문 헌

[1] A. J. Glenstrup, T. Engell-Nielsen, "Eye Controlled Media: Present and Future State", B. S. Dissertation, Copenhagen University, 1995.

[2] K. F. Arrington, "Viewpoint Eye Tracker", Arrington Research, 1997.

[3] Flock of Birds : Installation and Operations Guide. Burlington, VT : Ascension Technology, 1999.

[4] T. Haslwanter, S. T. Moore, "A Theoreti cal Analysis of Three-Dimensional Eye Position Measurement using Polar Cross-Correlation", IEEE Trans. on Biomed. Eng., vol. 42, no. 11, pp. 1053~1061, November 1995.

[5] R. S. Allison, M. Eizenman and B. S. K. Cheung, "Combined Head and Eye Tracking System for Dynamic Testing of the Vestibular System", IEEE Trans. on Biomed. Eng., vol. 43, no. 11, pp. 1073~1082, 1996.

[6] K. T. Lee, "Eye-Controlled Human/Computer Interface using the Line-of-Sight and the Intentional Blink", Ph. D. Dissertation, Korea Advanced Institute of Science and Technology, 1995.

[7] D. H. Kim, "Development of an Eye-Gaze Tracking System using Video Image Sequence and Magnetic Sensory Information", M. S. Dissertation, Korea Advanced Institute of Science and Technology, 2000.

[8] J. Merchant, R. Morrissette and J. L.

Porterfield, "Remote measurement of Eye Direction Allowing Subject Motion Over One Cubic Foot of Space", IEEE Trans. on Biomed. Eng., vol. BME-21, no. 4, pp. 309~317, July 1974.

[9] X. Xie, R. Sudhakar and H. Zhuang, "Real-Time Eye Feature Tracking from a Video Image Sequence using Kalman Filter", IEEE Trans. on Syst., Man and Cy., vol. 25, no. 12, pp. 1568~1577, December 1995.

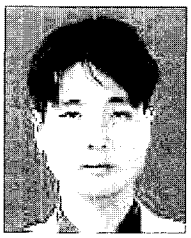
[10] A. L. Yuille, P. W. Hallinan and D. S. Cohen, "Feature Extraction from Faces using Deformable Templates", Int. J. Comp. Vision, vol. 8, no. 2, pp. 99~111, February 1992.

[11] X. Xie, R. Sudhakar and H. Zhuang, "On Improving Eye Feature Extraction using Deformable Templates", Pattern Recogn., vol. 27, no. 6, pp. 791~799, January 1994.

저 자 소 개

金 度 亨(正會員)

1998년 : 전북대학교 공과대학 제어계측공학과 학사. 2000년 : 한국과학기술원 전자전산학과 석사. 2000년 8월~현재 : 한국과학기술원 전자전산학과 박사과정



金 在 憲(正會員)

1997년 : 연세대학교 공과대학 전자공학과 학사. 1999년 : 한국과학기술원 전기 및 전자공학과 석사. 1999년~현재 : 한국과학기술원 전자전산학과 박사과정

鄭 明 振(正會員)

1973년 : 서울대학교 공과대학 전기공학과 학사. 1977년 : 미시간대학교 전기공학과 석사. 1983년 : 미시간대학교 제어공학과 박사. 1976년 : 국방과학연구소 연구원. 1981년 1월~1983년 8월 : 미시간대학교 CRIM 연구 조교. 1983년 10월~현재 : 한국과학기술원 전자전산학과 교수