
실시간 쌍안구 추적 마우스 시스템 구현에 관한 연구

류광렬* · 채덕현**

Real Time System Realization for Binocular Eyeball Tracking Mouse

Kwang Ryol Ryu* · Duck Hyun Chai**

이 논문은 2006년도 목원대학교 교수연구년 수혜로 이루어졌음

요 약

본 논문은 컴퓨터 모니터 거리 30-40cm에서 실시간 쌍안 눈동자 추적 마우스 시스템 구현에 관한 연구이다. 눈동자 탐색과 커서의 추적과정은 소형 CCD카메라에서 받은 안면영상은 2진영상으로 변환되고 눈 주위의 5영역 마스크 방법으로 눈을 찾고 측4점대각선위치 지정방법으로 각 홍채를 탐색한다. 커서추적은 홍채중심좌표의 이동위치를 측정하여 이루어진다. 화면커서 제어는 홍채의 최대 이동거리를 측정하여 화면커서최대 이동거리와 비교하고 홍채의 움직임에 비례해서 커서가 이동해야 하는 거리를 계산하여 화면커서를 화면의 바라보는 지점에 위치시킴으로써 화면커서가 제어된다. 실험결과 구현시스템은 비교적 간단하고 실시간으로 동작속도가 빠름을 보였다.

ABSTRACT

A real time system realization for binocular eyeball tracking mouse on the computer monitor being far from 30-40cm is presented in the paper. The processing for searching eyeball and tracking the cursor are that a facial image is acquired by the small CCD camera, convert it into binary image, search for the eye two using the five region mask method in the eye surroundings and the side four points diagonal positioning method is searched the each iris. The tracking cursor is matched by measuring the iris central moving position. The cursor controlling is achieved by comparing two related distances between the iris maximum moving and the cursor moving to calculate the moving distance from gazing position and screen. The experimental results show that the binocular eyeball mouse system is simple and fast to be real time.

키워드

Binocular Eyeball Tracking mouse, Irises Tracking, FRM, SFPDP, Screen Cursor Matching

I. 서 론

휴먼 인터페이스에서 기존의 손으로 사용하는 키보드

나 마우스 대신 눈의 움직임 추적으로 컴퓨터 및 자동화 기기를 사용하는 연구가 진행되고 있다. 이 연구는 모니터의 화면을 응시함으로써 화면상의 마우스 커서를 움직

* 목원대학교 IT공학과 정교수

접수일자 : 2006. 6. 28

** 목원대학교 IT공학과 석사

이고 아이콘을 선택하는 장애인용 컴퓨터 인터페이스, 자동차를 운전하는 사람의 눈을 관찰하여 졸음경보를 알리는 장치, 군사적 시선추적 제어시스템, 눈을 따라 움직이는 인명구조 로봇, 공장시설관리 등의 분야에서 사용될 수 있다.[1]. 이를 위해 카메라에서 안면을 인식하고 눈을 찾고 홍채의 움직임 추적해야 한다. 눈과 홍채의 중심좌표를 찾는 기법에는 눈이나 홍채의 테두리형태가 비슷한 물체를 추적하는 템플릿 매칭방법이 있고[2]. 물체의 전반적인 형태를 이용한 기하학적 모델을 이용한 방법이 있다[3,4]. 또한 유사한 형태를 신경회로망에 연결하거나, 눈의 생리적 특성을 이용하여 눈을 찾고[5,6]. 얼굴과 눈의 통계적 모델을 적용하는 방법, 눈에서 반사되는 빛을 이용하거나 눈에 적외선을 비춰 반사되는 빛을 추적할 수 있다.[7] 대부분의 기법은 단일 홍채 탐색기법이며 이미지센서가 눈과 밀착되어 불편한 점이 있다. 본 연구는 30-40cm 거리에 있는 CCD카메라로부터 안면영상을 획득한다. 영상에서 쌍안과 홍채를 찾고 홍채의 움직임을 추적하는 시선추적한다. 그 과정은 안면영상을 2진영상으로 변환하고 FRM(Five Region Mask)기법을 이용하여 눈을 찾는다[8,9]. 눈에서 홍채의 중심을 찾는 방법은 SFPDP(Side Four Points Diagonal Positioning)기법을 적용한다.[10,11] 또한 코의 위치를 탐색하여 오류를 제거한다. 그리고 홍채와 홍채 중앙점에 중심점을 지정하여 중심점의 이동정도에 따라 홍채의 수직 수평 시선방향을 추적한다. 동시에 실시간으로 컴퓨터 모니터 화면의 커서와 수직 수평 평면좌표 상에 대응시켜 쌍안추적 화면커서 제어시스템을 구현한다. 또한 샘플실험하고 고찰한다.

II. 쌍안 눈동자 추적 마우스 시스템

2.1. 쌍안 홍채 추적

제한환경에서 안면영상을 획득, 2진영상으로 변환하고 눈과 홍채를 추적한다. 환경에 따른 적절한 임계값을 눈과 홍채가 명확하게 보이도록 조정한다. 입력영상은 눈과 홍채는 분명하게 나타나야 한다. 머리카락으로 눈을 가리거나 모자의 그림자가 눈의 주변을 어둡게 하면 눈의 식별이 불가능하다. 눈과 홍채가 명확한 영상을 이진영상으로 변환할 때 임계값을 주변의 밝기에 따라 조정한다. 주위가 밝은 환경에서 임계값을 낮게 조정하면 화면은 모두 흰색으로 변한다. 반대로 어두운 환경에서 임계값을

높게 조정하면 화면은 모두 검게 변한다. 그래서 눈과 홍채를 나타내는 부분이 혼합되어 구분이 어렵게 된다.

2진영상은 컬러영상이나 그레이영상보다 영상을 나타내는 정보의 크기가 상대적으로 매우 작다. RGB컬러는 한 가지 색을 나타내기 위해 빨간 색의 강도와 녹색의 강도 그리고 파란 색의 강도 값들이 필요하다. HSI컬러는 색상, 채도, 그리고 명도의 값들이 필요하다. 그레이영상 역시 흑백의 강도가 단계별로 이루어져 있다. 그러나 이진영상은 2비트로 이루어져 데이터의 크기가 가장 작기 때문에 실시간처리를 가능하게 한다. 그리고 홍채는 검은 색이므로 이진영상으로 변환할 때 임계값의 영향을 적게 받는다. 피부색이 검은 색에 가깝지 않으면 이진영상으로 변환함으로써 안면의 눈과 코와 피부를 쉽게 구분할 수 있다.

눈과 홍채는 FRM과 SFPDP기법으로 추적된다. 눈과 홍채가 표시된 이진영상은 대부분 눈, 코, 머리카락, 귀 그리고 눈썹을 볼 수 있다. 그리고 눈 주위에 네 개의 공백영역이 있다. 눈과 눈 사이, 눈과 귀 사이, 눈썹과 눈 사이 그리고 눈과 코 사이에 공백이 존재한다. 그리고 입력영상에서 눈을 나타내는 부분은 이진영상으로 변환하였으므로 눈 크기정도의 검은 색 픽셀이 존재한다. 이 영역들을 하나로 묶어 마스크화한다. 마스크된 FRM의 구성은 그림 2-1과 같이 지정된다.

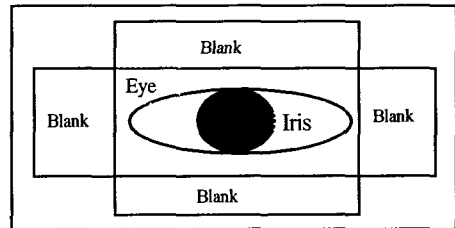


그림. 2-1 5영역 마스크
Fig. 2-1 Five Region Masks

상하좌우 방향에 네 개의 공백영역이 존재해야 하고 중심에 눈 크기정도의 검은색 픽셀들이 존재해야 하는 한 개의 영역을 가진 마스크이다. 그림 3-2는 FRM을 이진영상에 적용하는 방법을 나타내고 있다. FRM기법은 이진영상에서 4개의 공백영역과 1개의 눈 영역으로 이루어진 부분만이 눈으로 인식한다.

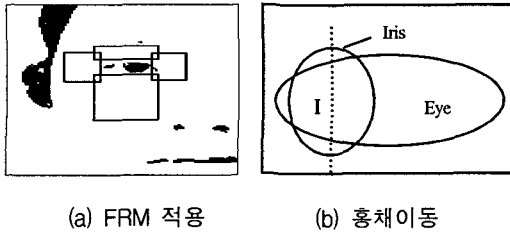


그림 2-2 눈과 눈동자의 정확한 위치
Fig. 2-2 Eye and eyeballs correct position

눈과 홍채의 구분은 시선추적을 위해 홍채의 움직임을 이용한다. 눈과 홍채는 서로 매우 근접해 있다. 홍채는 눈꺼풀에 의해 일부가 가려져 있다. 그리고 눈꺼풀의 그림자가 형성된다. 이와 같은 문제점을 해결 하고 눈 안에서 정확한 홍채의 위치를 찾기 위해 홍채가 중앙을 주시할 때 눈에서 가장 두꺼운 부분을 찾는다. 홍채는 검은색이므로 이진영상에서는 눈에서 홍채가 위치한 부분이 가장 두껍게 나타난다. 그러나 홍채가 좌우로 이동할 경우 눈꺼풀에 홍채가 가려져 홍채의 위치에 오차가 발생한다. 눈에서 홍채가 위치한 부분이 눈의 다른 부분보다 비교적 두껍게 나타난다. 따라서 오차를 줄이기 위해 그림 2-2 (b)와 같이 눈꺼풀에 가려진 홍채가 가장 많이 분포한 부분의 중앙 I 지점을 홍채의 위치로 지정한다.

FPDP기법과 SFPDP기법은 홍채의 테두리에 4개의 좌표를 이용하여 홍채의 중심위치를 추적한다. FPDP기법과 SFPDP기법은 4개의 좌표를 지정할 때 특정한 조건에서 오차를 보인다. FPDP기법은 그림 2-3의 홍채의 좌우 아래 테두리부분에 각각 2두개의 좌표를 지정한다. 눈의 좌우 아래 부분에서 4개의 픽셀이 화살표와 같이 수직으로 홍채의 테두리를 향해 진행한다. 그리고 홍채의 테두리와 접촉하여 좌표를 지정한다. 눈꺼풀에 의해 가려진 홍채의 불규칙한 테두리부분은 좌표지정

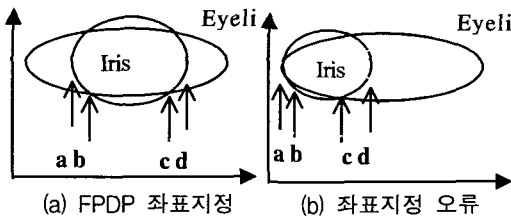


그림 2-3 FPDP 좌표지정
Fig. 2-3 FPDP coordinate designation

구간에서 제외된다. 그러나 홍채가 눈의 가운데 위치한 경우는 정상적인 좌표지정을 한다. 그러나 그림 2-3(b)와 같이 홍채가 눈의 한쪽 방향으로 이동한 경우 눈꺼풀은 홍채의 테두리를 가리게 되고 홍채의 테두리위에 지정되어야 하는 4개의 점들 중 일부는 눈꺼풀을 가리키는 오류가 발생한다. 이 문제는 SFPDP기법으로 해결한다. 이 기법은 홍채가 눈의 가운데 위치한 경우와 좌우방향으로 이동한 경우의 4개의 좌표지정 방법이 다르다. 그 과정은 눈의 좌우경계와 홍채의 위치를 지정하고 안면의 기울기를 측정한다. 눈안에서 홍채의 위치에 따라 홍채의 테두리에 좌표를 지정한다. 그림 2-4는 좌표 f1(Xf,Yf)와 f2(Xg,Yg)를 지나는 수직이등분선의 교점 h가 홍채의 중심위치이다. 홍채의 중심위치를 표현하는 좌표는 (I)식으로 구한다.

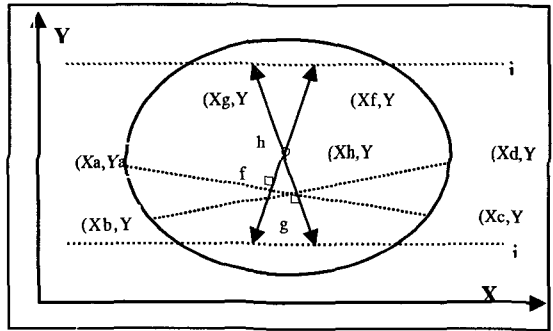


그림 2-4 눈동자 중심위치 추적
Fig. 2-4 Eyeball center position Tracking

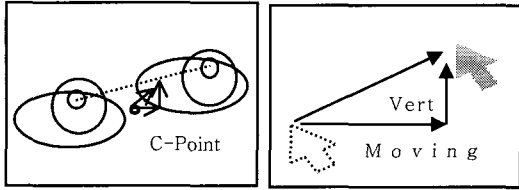
$$f_i(X_i, Y_i) = \left((-Y_f - \frac{X_c - X_a}{Y_c - Y_a} X_f + Y_g + \frac{X_d - X_b}{Y_d - Y_b} X_g), \right. \\ \left. (-\frac{X_c - X_a}{Y_c - Y_a} (-Y_f - \frac{X_c - X_a}{Y_c - Y_a} X_f + Y_g + \frac{X_d - X_b}{Y_d - Y_b} X_g) + X_f + \frac{X_c - X_a}{Y_c - Y_a} X_f) \right) \quad (1)$$

여기서 Xf, Yf는 a와 c의 중심, Yf, Yg는 b와 d의 중심값이다.

2.2. 홍채와 화면커서 추적과정

홍채의 움직임을 이용한 화면커서 제어과정은 두 눈 홍채의 중심좌표 사이의 중심점 이동을 이용하여 화면커서를 추적한다. 그림 2-5(a)와 같이 두 눈 홍채의 중심좌표 사이의 중심점 C가 이동했을 경우 시선이동을 중심점의 수평이동거리와 수직이동거리성분을 측정한다. 그리고

그림 2-5(b)와 같이 화면커서를 대응시킴으로써 화면커서를 동시에 이동 추적한다.



(a) 눈동자의 이동 (b) 화면커서 이동

그림 2-5 눈동자와 화면커서 매칭
Fig. 2-5 Eyeballs and cursor movement

(1) 홍채의 수평이동 측정

홍채가 수평으로 이동한 거리를 계산하기 위해 두 눈의 거리와 두 홍채의 거리를 측정한다. 그리고 두 눈의 거리와 두 홍채의 거리의 차를 이용하여 홍채의 좌우이동거리를 계산한다. 수평이동거리 측정과정은 쌍안 홍채의 위치 지정, 쌍안 홍채의 중심 위치 지정, 쌍안 위치 지정, 쌍안과 홍채의 중심위치 간의 거리에 의한 수평이동거리 지정 순이다. 그림 2-6는 홍채의 이동을 계산하기 위해 홍채와 눈의 위치사이의 거리를 나타낸다. IC는 좌측홍채의 위치 L과 우측홍채의 위치 R의 중심 점을 나타내고 EC는 두 눈의 안쪽 끝부분사이의 중심점을 나타낸다. 그리고 d는 IC와 EC사이의 거리를 나타낸다. 홍채가 화면의 정중앙을 바라볼 경우 d는 0이 되도록 초기 값을 지정한다. 홍채를 이동하면 IC와 EC 사이에 거리가 생겨 d의 값이 정해진다. 눈이 좌측으로 이동할 경우 -값을 갖고 우측으로 이동할 경우 +값을 갖는다. +와 -를 이용하여 눈의 수평 이동 방향을 알 수 있다. 그리고 d값의 크기를 측정하여 홍채의 수평이동거리를 알 수 있다.

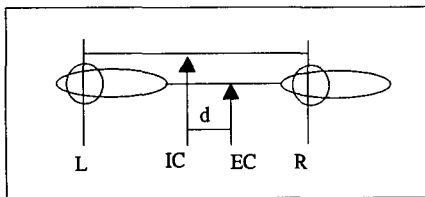


그림. 2-6 수평이동측정
Fig. 2-6 Measurement of horizontal moving

(2) 홍채의 수직이동 측정

홍채의 수직이동거리를 측정하기 위해 코의 위치를 찾

는다. 그리고 화면의 중앙을 응시할 때 홍채의 중심좌표와 코 사이의 거리와 화면의 중앙이 아닌 다른 부분을 응시할 때 홍채의 중심좌표와 코사이의 거리의 차를 이용하여 홍채의 수직이동거리를 계산한다. 그 과정은 눈의 위치지정, 두눈사이 아래 방향에 코의 위치 탐색, 홍채수평이동의 기준점지정, 기준점과 홍채의 수직이동위치간의 거리를 측정하여 수직이동거리를 산정한다. 코는 눈과 눈 사이의 아래 부분에 위치한다. 눈과 눈을 연결한 선분의 수직이동분선을 따라 이동하면서 코를 나타내는 픽셀들을 찾는다.

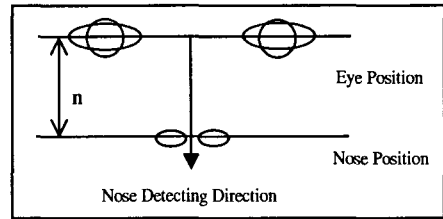


그림. 2-7 코의 위치 검색
Fig. 2-7 Tracking nose position

그림2-7은 코의 추적방법을 나타낸 것이다. 일반적으로 빛의 방향이 위에서 아래로 향할 경우 콧구멍에 의해 그림자가 발생한다. 따라서 코를 나타내는 그림자픽셀들은 수평으로 길게 분포해 있다. 이와 같은 특성을 이용하여 코의 그림자가 위치한 지점을 코의 위치로 지정한다. n은 홍채와 코사이의 거리를 나타낸다. 사용자가 얼굴을 고정된 상태에서 시선이 화면의 정중앙을 응시할 때 n값을 초기 값으로 지정한다. 홍채가 상하로 이동하면 그림 2-8의 m과 같이 초기 값과 홍채의 위치사이의 거리가 발생한다. 홍채위치에서 초기 값을 뺀 경우 m의 값이 '+'이면 홍채는 위로 이동한 것이고 '-'이면 아래로 이동한 것이다. m을 이용하여 홍채의 수직이동거리를 측정한다.

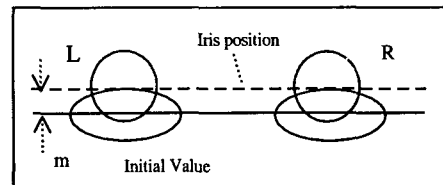


그림. 2-8 수직이동측정
Fig. 2-8 measurement of vertical moving

(3) 화면커서의 제어

수평이동거리를 이용하여 화면커서와 홍채의 이동거리가 대응해야 한다. 대응시키는 과정은 초기값 지정, 홍채 최대이동거리 측정, 최대시야각 측정, 최대값과 실제값의 비례식으로 시야각 계산, 시야각을 시선과 화면상의 시선위치관계식에 대입, 화면상의 시선위치에 화면커서 이동 등으로 이루어진다. 시선방향과 화면상의 시선지점의 관계는 그림 2-9와 같다. O는 화면의 중앙을 나타낸다. 그리고 홍채와 홍채사이의 중심점이 화면의 중앙을 응시하는 기준점이다. 홍채와 홍채사이의 중심점과 화면의 중앙에 위치한 O사이의 거리는 d다. 화면의 좌우부분을 응시할 경우 응시하는 지점과 O사이의 거리는 w다. 이때 화면의 중심점 O를 바라보는 시선방향과 화면의 끝부분을 응시하는 방향이 이루는 각은 θ 다. 사용자가 얼굴의 위치를 고정한 상태에서 화면의 중앙을 응시하여 그림 2-6와 2-8의 홍채이동거리가 0이 될 경우 눈과 홍채의 좌표들의 거리를 초기 값으로 지정한다. 홍채가 이동할 때 그림 2-6와 2-8의 홍채이동거리를 픽셀단위에서 그림 2-9의 θ 로 변환한다. 변환 방법은 다음과 같다. 화면의 좌측이나 우측의 끝부분을 응시하여 그림 2-6와 2-8의 홍채이동거리의 최대 변화 값을 측정한다. 그리고 그림 2-9의 θ 의 최대 변화 값을 측정한다. 실제 홍채의 수평이동거리 값과 최대 변화 값의 관계는 D는 홍채의 이동거리를 나타내고 D_{max} 는 홍채이동거리의 최대값을 나타내면 $D_{h_{max}} : \theta_{h_{max}} = D_h : \theta_h$ 와 같고 수직이동거리값과 최대변화값의 관계는 $D_{v_{max}} : \theta_{v_{max}} = D_v : \theta_v$ 으로 표현된다. θ_{max} 는 θ 의 최대 변화 값을 나타낸다. 실제 시야각 θ 의 값을 계산하고 관계를 식으로 표현하면 수평과 수직은 그림 2-9(a)에서 $w_h = d \times \tan \theta_h$ 와 (b)에서 $w_v = d \times \tan \theta_v$ 로 각각 구한다.

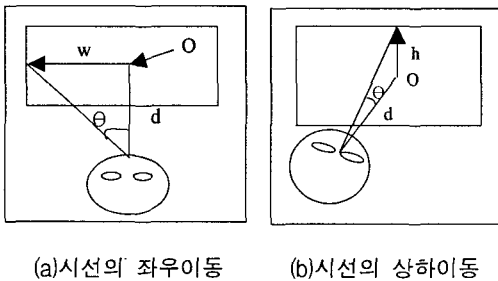
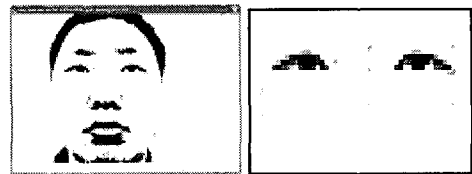


그림. 2-9 화면상의 시선위치
Fig. 2-9 Gazing position on screen

이 관계식에서 화면위의 시선위치를 찾아 화면커서를 시선위치에 위치시킨다.

III. 실험 및 고찰

실험은 안면영상에서 두 눈과 홍채를 찾고 홍채의 움직임을 추적하여 시선방향에 화면커서를 이동시킨다. 실제 실험을 위한 배치는 화면과 카메라 그리고 얼굴을 위치시켰다. 빛의 방향은 위에서 아래로 향하게 하고 카메라는 얼굴의 아래쪽에 위치하였다. 홍채가 보다 잘 보이도록 하기 위해 주변의 밝기를 밝게 하였다. 모니터는 18.1 inch LCD모니터를 사용하였고 해상도는 1280*1024 픽셀로 지정했다. 실제로 모니터에서 영상을 나타내는 부분의 크기는 36cm * 28.5cm다. 눈과 화면사이의 시선거리는 43cm를 유지하였다. 그리고 실험에 사용된 카메라는 30만화소의 컴퓨터용 CCD화상카메라를 사용하였다. 카메라 렌즈의 높이는 13.2cm이고 눈과의 거리는 33cm를 유지한다. 먼저 640*480 픽셀의 컬러영상을 실시간으로 획득한다. 획득한 영상을 2진영상으로 변환한다. 그리고 FRM기법을 적용하여 그림 3-1(a)와 같이 안면에서 눈의 위치를 추적한다.



(a) FRM 쌍안 영상 (b) 홍채의 중심위치

그림. 3-1 눈동자 추적 결과
Fig. 3-1 Eyeball tracking result

눈의 위치에서 그림 3-1 (b)와 같이 눈동자의 위치와 눈의 양쪽 가장자리에 눈의 경계를 지정하여 눈안에서 홍채 위치를 추적한다. FPDP기법과 SFPDP기법을 응용하여 홍채의 중심 좌표를 추적한다. 홍채의 중심에 '+'표시한다. 이진영상에 FRM기법을 사용하여 눈의 위치를 네모 모양의 선으로 나타내고 SFPDP기법의 홍채의 중심위치 추적을 위한 홍채테두리 위의 4개의 좌표들을 십자가 모양으로 좌측 눈에 표시한다. 그리고 창의 상단부분에 화면(x,y) 위치를 나타내는 좌표값을 표시한다. 두 홍채사이

의 중앙좌표의 최대 이동거리를 측정하기 위해 수평방향과 수직방향의 모니터끝부분을 바라볼 때의 X, Y좌표를 측정한다. 그림 3-2의 (a)와 (b)는 프레임마다 두 홍채의 중심좌표사이의 중심점을 나타내는 X, Y좌표값의 변화를 각각 나타낸 것이다.

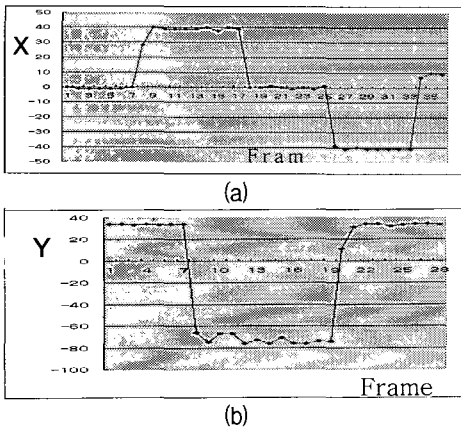
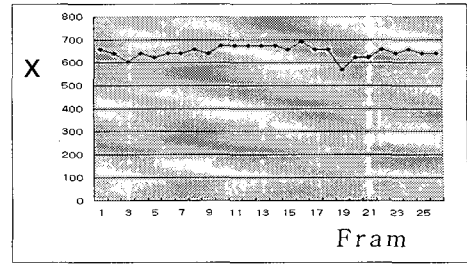


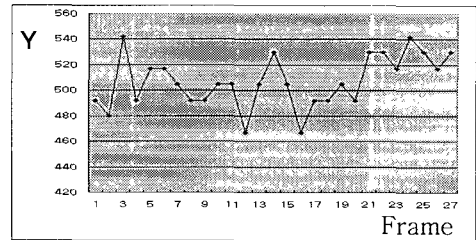
그림. 3-2 홍채의 최대이동범위
Fig. 3-2 Iris's maximum moving range

실험결과 X좌표는 화면의 우측 끝부분을 바라볼 경우 약 40픽셀 이동하고 좌측끝부분을 바라볼 경우 약 -40픽셀 이동했다. 그리고 Y좌표는 화면의 가장 윗부분을 바라볼 경우 약 34픽셀값을 보이고 아랫부분을 바라볼 경우 약 -77과 -68사이의 픽셀값을 보인다. 화면의 정중앙을 바라볼 경우 X좌표는 약 0픽셀의 값을 보였고 Y좌표의 경우 -8과 3사이의 값을 보였다. X값은 안정적이지만 Y값은 연속해서 오차를 발생시키는 것을 알 수 있다. 눈과 눈 사이를 검색하여 좌우로 넓게 분포한 코의 위치를 찾는다. 콧구멍의 그림자 사이에 코의 위치를 지정한다. 화면의 중앙을 기준으로 수평방향과 수직방향을 바라보며 약 2초 동안 화면의 끝에서 끝으로 시선위치를 이동시킨다. 그리고 3회 왕복한다. 그 결과, 화면커서의 수평이동시 Y좌표변화와 수직이동시 X좌표변화는 그림 3-3과 같다.

임의로 남자 10명을 대상으로 위치를 임의로 옮겨가며 앞의 과정들을 5회씩 테스트 하였다. 실험에 사용된 모니터의 해상도는 1280*1024 픽셀이고 가로와 세로의 길이는 36cm*28.5cm이다. 이 관계를 이용하면 1픽셀은 약 0.028 cm의 길이를 갖는다. 화면커서이동시 평균오차 픽셀은 홍채의 X좌표 최대이동거리 75, Y좌표 중앙 상단과 하단을 주시할 때 홍채와 코사이의 거리는 166과 60, 화면



(a) 수직이동시 X좌표변화



(b) 수평이동시 Y좌표변화

그림. 3-3 화면커서의 수평 및 수직방향 이동시 좌표변화

Fig. 3-3 Variations on Hor. and Vert. movement of screen cursor

중앙의 X과 Y좌표는 0와 132, 수직이동시 X좌표 (642,3), 수평이동시 Y좌표 (510,17)이다. 결과를 이용하여 실제 바라보는 지점과 화면상의 위치사이의 거리를 계산한 후 오차로 지정한다. 그리고 기존의 방법과 5명의 평균적인 정확도를 비교결과 평균오차는 구현시스템 0.47, 다중신경망 0.47로 비교적 유사하다.

초당 프레임변화는 눈을 찾지 못한 경우와 눈을 찾은 경우로 나누어 측정한다. 그 결과는 그림 2-9와 같이 눈을 찾지 못한 경우는 평균적으로 초당 프레임의 속도를 보이고 눈을 찾은 경우는 약 3.75 프레임의 속도를 나타내고 있다.

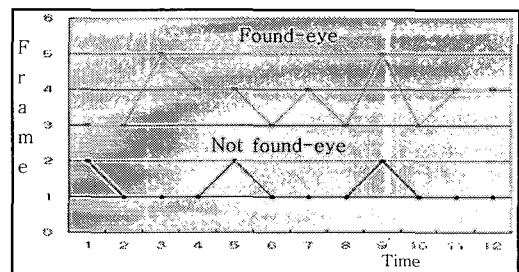


그림. 2-9 처리속도 측정결과
Fig. 2-9 Processing speed measurement

표 3-1은 기존의 방법들의 초당 프레임 속도를 나타낸 것이다. 수치가 높을수록 속도가 빠르다. 눈을 찾은 경우의 속도는 다중 신경망보다 0.6프레임 빠르고 선형보간법보다 0.1프레임 빠르다. 따라서 다중신경망보다 2.4배 빠른 속도를 보였고 선형보간법 보다 1.5배 빠른 속도를 보였다.

표. 3-1 처리속도 및 평균오차 비교
Table. 3-1 Comparison with processing time

방법	다중 신경망	이동량 추정 방법	선형 보간법	구현시스템
처리시간 (fps)	1.11	0.91	2.5	3.75
평균오차 (cm)	0.47	0.42	0.47	0.47

IV. 결 론

본 연구는 쌍안 입력영상을 이진영상으로 변환하고 FRM기법을 사용하여 눈의 위치를 찾아, 눈의 위치에서 홍채의 중심위치추적을 위해 FPDP기법과 SFPDP기법을 응용하였다. 그리고 홍채의 중심좌표의 움직임을 측정하여 화면커서를 제어했다. 화면커서 제어는 홍채의 최대 이동거리를 측정하여 화면커서 최대 이동거리와 비교하였다. 그리고 홍채의 움직임에 비례해서 커서가 이동해야 하는 거리를 계산하여 화면커서를 화면의 바라보는 지점에 위치시킴으로써 화면커서를 제어했다. 실험결과 이동 거리를 측정 20세와 30세 사이의 남자 10명을 5회씩 테스트한 결과 화면커서의 수직이동시 X좌표의 평균오차는 7 픽셀이고 수평이동시 Y좌표의 평균오차는 17.4 픽셀이다. 실제 바라보는 지점과 화면커서위치의 평균오차거리는 0.47cm로서 다중신경망과 선형보간법의 경우 오차의 큰 차이는 없지만 단일 신경망 및 얼굴의 3차원 이동량 추정방법 보다 0.05픽셀정도 오차가 크다. 처리속도는 눈을 찾은 경우만을 고려하면 약 3.75 프레임의 속도를 나타내고 있다. 이는 다중 신경망보다 0.6프레임 빠르고 선형보간법보다 0.1프레임 빠르다. 따라서 다중신경망보다 약 2.4배 빠른 속도를 보였고 선형보간법 보다 약 1.5배 빠른 속도를 보였다. 전체적으로 30-40cm 거리에서 쌍안 눈동자를 추적하여 동시에 화면 커서와 매칭시킨 결과 실시간으로 동작하였으며 시스템 구현도 간단하고 경제적이다. 이 연구는 손을 움직일 수 없는 장애인이 눈으로 화면을 바

라보는 것만으로 컴퓨터를 제어할 수 있는 가능성을 제시하였다. 한편, 주변의 밝기에 영향을 받지 않고 홍채에 빛이 반사되어도 정확한 홍채추적이 가능한 연구가 요구된다.

참고문헌

- [1] Hideo Kawai, Shinichi Tamura, "Eye Movement Analysis System Using Fundus Images." Pattern Recognition. Vol. 19, No.1, pp. 77-84, 1986
- [2] Hori, Y. Shimizu, K. Nakamura, Y. Kuroda, T."A real-time multi face detection technique using positive-negative lines-of-face template" Pattern Recognition, vol.1, pp. 765-768, 2004
- [3] 김재희, "A study on the Development of Data Base of Human Sensibility and Its Support for the Commercial Product", 연세대학교, 1999
- [4] S.-H.Jeng, et al., "Facial feature detection using geometrical face model: an efficient approach", Pattern Recognition, vol.30, pp.273-281, 1997
- [5] S. Baluja & Pomerleau "Non-intrusive Gaze Tracking Using Artificial Neural Networks" Technical Report CMU-CS-94-102, Carnegie Mellon University.
- [6] 김재희, "A Visual Computer Interface for Ergonomics", 연세대학교, 2000
- [7] A. Haro, M. Flickner and I. Essa, "Detection and tracking eyes by using their physiological properties, dynamics and appearance", Proceedings of IEEE CVPR 2002
- [8] D. Maio, D. Maltoni, "Real-time face location on gray-level static images", Pattern Recognition, Vol.33, pp.1525-1539, 2000
- [9] 채덕현, 강희조, 김윤호, 류광렬, "안면영상에서 눈과 홍채의 중심위치 추적능력 향상", 『한국해양정보통신학회』 추계종합학술대회 논문집, Vol.8, No.2, pp.323-326, 2004
- [10] Duck Hyun, Chai and Kwang Ryol, Ryu, "A Study on the Enhancement of Tracking Capability for Iris Image", Proceedings of ISMICS'2004, pp. 24-27, 2004
- [11] 채덕현, 류광렬, "S-FPDP기법을 적용한 동영상 홍채의 위치추적 향상", 『한국해양정보통신학회』, 9권 5호, 2005.

저자소개



류 광 열(Kwang Ryol Ryu)

1988년 경희대학교 공학박사
1996년 University of Pittsburgh School
of Engineering 초빙교수

현재 : 목원대학교 IT공학부 정교수, University of Pittsburgh
Medical Center 초빙교수

※ 관심분야 : Video and Image Signal Processing, Medical
Engineering, Speech Recognition



채 덕 현(Duck Hyun Chai)

2002년 목원대학교 전자공학과 공학사
현재 목원대학교 전자공학과 공학석사

※ 관심분야 : 디지털영상처리