제 26 회 한국정보처리학회 추계학술발표대회 논문집 제 13 권 제 2 호 (2006. 11)

응시 위치 추적을 이용한 스크린 커서 제어

장동현, 김정규
시립인천대학교 컴퓨터공학과

A Study on The Screen Cursor Control using Gaze Tracking

Dong-hyun Jang , Chung-kyue Kim
Department of Computer Science and Engineering, University of Incheon

요 약

컴퓨터의 급속한 발전 속도와 맞물려 사용자의 의지나 몸짓, 감성을 인지하는 보다 편리하고 자연스러운 휴먼 인터페이스에 대한 요구가 늘어나고 있다. 휴먼 인터페이스 가운데 응시 위치 추적은 현재 사용자가 쳐다보고 있는 위치를 컴퓨터 시각 인식 방법을 통하여 파악하는 연구이다. 본 논문에서는 여러 감성 컴퓨터 인터페이스 중 눈동자를 통해 컴퓨터의 입력장치를 간접적으로 제어하는 방법에 대해 기술한다. 웹 카메라를 통해 입력 받은 영상을 이용하여 눈동자의 위치 이동을 탐색하여 마우스를 제어한다.

1. 서론

최근 컴퓨터의 급속한 발전 속도와 함께, 사용자의 의지나 감성을 인지하는 보다 편리하고 자연스러운 휴먼 인터페이스에 대한 요구가 나날이 증대되고 있다. [5]

이에 따라 기존의 키보드나 마우스 대신 음성 인식, 문자 인식, 얼굴 인식 및 동작 인식 등을 이용하여 보다 인간 친화적인 감성 컴퓨터 인터페이스를 구축하고자 하는 방법들은 연구를 거듭하며 발전해 오고 있다.[7]

이 중에서 얼굴과 눈동자의 자연스러운 움직임을 이용하여 모니터의 화면을 응시함으로써 일정 위치를 지시하거나 선택하며, 또 화면상의 마우스 커서를 움직이고 클릭하는 인터페이스 개발 연구는 단지 공학적인 관점을 떠나 사용자의 관심 위치 및 감성까지 파악 할 수 있는 중요한 수단으로 인식되어, 미국, 유럽 및 일본 등지에서 연구가 진행되고 있으며, 국내에서도 몇몇 대학과 기업을 중심으로 연구가 추진되고 있다.

또한 손과 같은 신체의 일부를 사용하는 입력장치들에 비해 컴퓨터를 동작시키는 속도를 매우 빠르게 할 수 있을 것이라 기대되고 있다. Yamato 의 연구에서 21 인치 모니터 좌측 상단 끝에서 우측 하단을 바라볼 때 걸리는 시간이 150ms 라고 나타낸바 있다.[2]

응시위치 추적에 관한 연구는 많은 활용분야가 있다. 손을 사용하지 못하는 상황 및 손을 쓸 수 없는 장애인들을 위한 컴퓨터 인터페이스가 대표적은 예이다.[4]

응시위치 추적 기술을 활용함으로써 컴퓨터에 사용되는 마우스 커서 제어를 대신할 수도 있다. 뿐만 아니라 손을 쓰기 힘든 공장과 같은 환경에서 눈동자의 움직임에 의해서 시스템을 제어할 수 있다.

본 연구에서는 웹 카메라를 통한 응시 위치 추적기술을 사용하여 간접적으로 마우스를 제어 할 수 있는 방법을 제안한다.

응시 추적을 위해서는 우선 눈동자의 위치를 알아내고 상하좌우 움직임을 파악 할 필요가 있다. 본 논문에서는 원형으로 배열된 직사각형의 윈도우를 이용한 눈동자 추적 알고리즘을 설명한다.

본 연구는 인천대학교 동북아물류센터(e-LRC) 지원에 의한 연구결과임.

2. 처리 알고리즘

영상의 이치화를 통해 눈동자의 위치를 파악하고 눈동자의 움직임을 통해 키보드와 마우스 같은 인터페이스 장치를 제어하는 알고리즘을 구현 하는 것이다.

2.1 눈동자 탐색을 위한 이치화

카메라에서 직접적으로 얻은 칼라 영상보다 효율적인 영상 처리를 위해 영상을 흑백으로 분류하는 이치화를 사용하였다.

일단은 원 영상을 회색조의 영상으로 [식 2.1]로 변환한다.[6]

$$GrayImage = 0.299 * R + 0.587 * G + 0.144 * B \ldots [식\ 2.1]$$

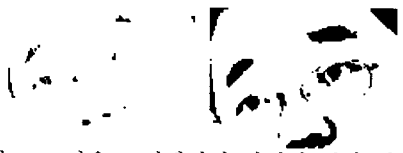
회색조로 바뀐 영상을 다시 밝기 값이 0 아니면 255 의 값들 중 하나의 값을 가지도록 이치 영상을 만든다. 이치 영상을 만드는 방법은 [식 2.2]로 변환한다

$$GrayImage > 임계값 = 255$$
$$GrayImage < 임계값 = 0 \ldots . [식\ 2.2]$$

이와 같이 변환된 영상은 아래 [그림 2.1]과 [그림 2.2] [그림 2.3]을 통해 확인 할 수 있다.

[그림 2.1] 원본 [그림 2.2] Gray [그림 2.3] Binary

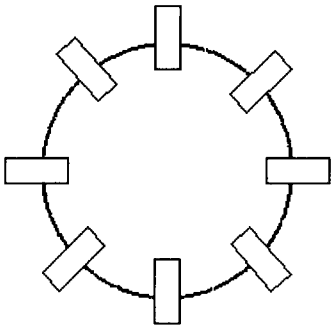
영상의 이치화는 필요 없는 정보를 삭제하고 눈동자 정보를 부각시켜주는 장점이 있다. 하지만 [그림 2.4]와 같이 조명에 변화에 따라 배경과 대상을 분리할 수 있는 임계치가 변하므로 이치화 임계값을 조명에 맞춰 동적으로 설정을 해 줘야 한다.



[그림 2.4] 같은 조명에서의 임계치 변화 차이

2.2 눈동자의 크기 및 위치 보정 마스크

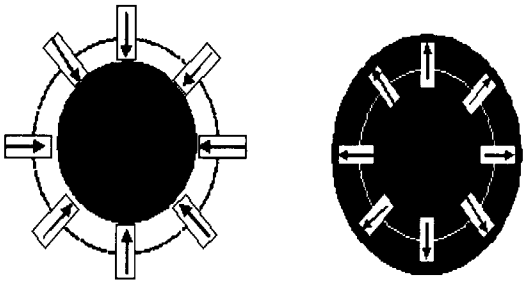
눈동자의 기본적인 위치와 크기를 찾는데 사용하는 객체의 기본적인 개념은 [그림 2.5]와 같다[3][4][8].



[그림 2.5] 눈동자 위치 및 크기 파악 기본 윈도우

객체는 중심점의 위치에 대한 정보와 가상의 원 크기에 대한 반지름 값을 가지다. 또한 객체는 8 개의 윈도우를 가지는데 이 윈도우는 영상의 흑백부분의 값을 계산 하기 위하여 사용한다. 이 윈도우를 통해 눈동자의 상하좌우 및 크기를 조절 하게 된다.

객체의 크기 조절은 다음 [식 2.3]을 통해 이루어진다.



[그림 2.6] 눈동자의 크기 조절

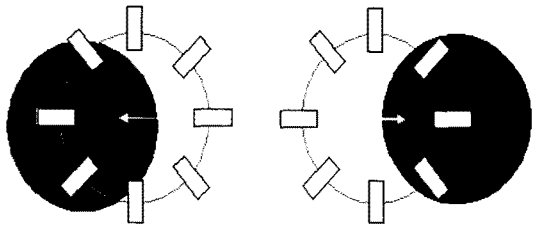
$$f_{resize} = \int_0^{2\pi} f_w(\theta)\, d\theta$$

..[식 2.3]

$f_w(\theta)$ = 객체의 중심을 지나는 수평선과 θ 각을 이루는 윈도우에너지

윈도우 안에서 들어 있는 모든 값이 검은색 값을 가지고 있다면 눈동자의 크기가 크다는 것을 알 수 있고 반대로 모든 값이 흰색의 값을 가지고 있다면 눈동자가 작음을 판단 할 수 있다. 즉 [식 2.3]에서의 값이 0 보다 크면 객체의 크기를 줄이고 0 보다 작으면 객체의 크기를 키워주어 객체의 크기를 조절한다.

객체의 좌우 조절방법은 크기 조절과 마찬가지로 윈도우를 이용하여 계산하여 위치를 보정한다. 좌우의 위치를 파악하는 방법은 코사인 함수를 이용한다. 코사인 함수는 90~270 도 사이에서 음의 값을 가지기 때문에 이를 이용하여 좌우 방향을 결정한다. [식 2.4]는 위와 같은 과정을 보여준다.



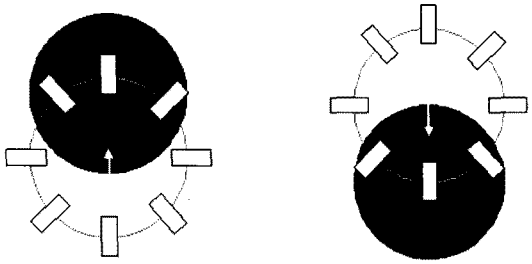
[그림 2.7] 눈동자의 좌우 위치 조절

$$f_{move_x} = \int_0^{2\pi} f_w(\theta)\cos\theta d\theta$$

.....[식 2.4]

눈동자가 오른쪽에 있으면 [식 2.4]는 양의 값을 가지게 되고 반대로 왼쪽에 있으면 음의 값을 가지게 된다. 이를 통해 눈동자의 좌우 위치를 보정한다.

눈동자의 상하의 위치를 보정하는 방법은 좌우 위치조절과 같은 방법으로 보정한다. 다른 점이 있다면 상하 위치 보정은 코사인 값이 아닌 사인 값을 이용한다는 것이다. 사인 값은 0~180 도에서 양의 값을 가지고 나머지는 음의 값을 가지기 때문에 상하 방향을 결정 할 수 있다.



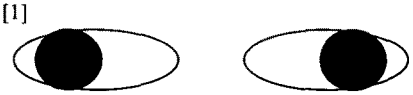
[그림 2.8] 눈동자의 상하 위치 조절

$$f_{move_y} = \int_0^{2\pi} f_w(\theta)\sin\theta d\theta$$

...[식 2.5]

2.3 눈동자 움직임을 통한 마우스 커서 제어

눈동자의 움직임을 감지하여 마우스 커서를 제어하는 방법은 일반 마우스의 움직임과 동일 하게 하였다. 즉 맨 처음 위치에서 어느 한쪽으로 눈동자가 움직이면 보고 있는 눈동자에 좌표가 변하면서 변화된 값만큼 마우스 커서가 계속 움직이고 눈동자가 원래 위치로 다시 돌아오면 멈추게 하는 형식으로 구현 하였다. [1]



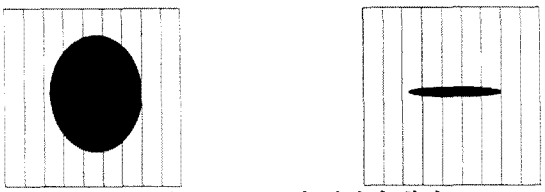
[그림 2.9] 눈동자의 좌우 움직임 파악

좌, 우 움직임은 눈동자의 움직임으로 충분히 파악 할 수 있으나 상, 하의 움직임은 위치로 파악 하는 것으로는 좌, 우의 눈동자의 움직임보다 움직임이 적다는 문제점 때문에 눈의 크기로 파악하는 방법을 사용하였다. 사람이 위쪽을 볼 때 눈을 크게 뜨게 되고 아래쪽을 볼 때는 눈을 반쯤 감겨 상대적으로 작게 보이는 현상을 이용하여 눈동자의 커서를 제어 하였다.

[그림 2.10] 눈동자의 상하 움직임 파악

2.4 눈 깜빡임을 이용한 마우스 클릭 이벤트 제어

마우스의 클릭 이벤트는 눈을 깜빡이는 것으로 처리 하였다. 즉 사람이 눈을 감게 되면 현재 마우스 커서가 있는 곳에서 클릭을 하게 된다. 눈을 감는 부분에 대한 연산 과정은 눈동자의 일정 범위에 세로로 마스크를 씌우고 검은색 픽셀과 흰색 픽셀을 이용하여 계산 하였다.



[그림 2.11] 눈동자 깜빡임 확인

즉 마스크에 검은색 픽셀과 흰색 픽셀을 비교하여 검은색 픽셀이 흰색 픽셀보다 월등히 적은 경우를 눈을 감았다고 판단한다.

좌측 눈에 깜빡임은 마우스의 좌 클릭으로 구현 했고 우측 눈에 깜빡임은 마우스의 우 클릭으로 구현 했다.

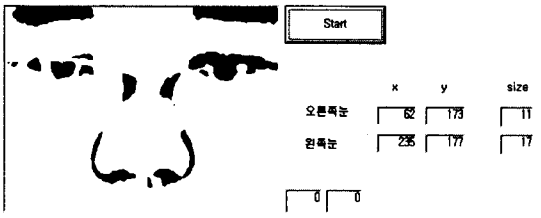
통상적인 인간의 눈 깜빡임에 대한 예외 처리는 두 눈을 다 감지 않고 한쪽 눈만 감을 경우에만 클릭 이벤트를 처리 하는 방식으로 구현 했다.

3. 실험 및 고찰

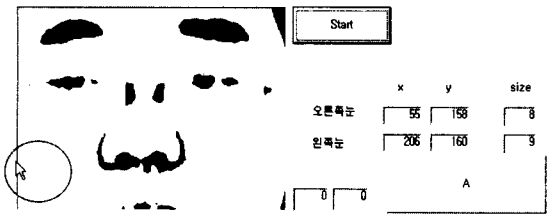
본 논문의 실험은 vfw32 라이브러리를 이용하여 320 * 240 화소 크기의 카메라 영상으로 실험 하였고, 개발 도구는 Visual Studio 6.0(MFC)을 사용하였다. 화면 캡춰는 Snagit 을 사용 하였다.

다음 그림은 본 논문에서 소개된 알고리즘을 사용하여 구현한 화면이다.

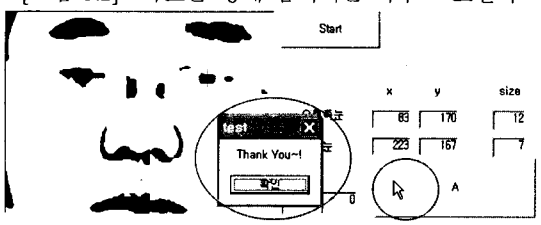
[그림 3.1]은 카메라에서 입력 받은 영상을 이치화한 후 눈동자 위치 및 크기 보정 마스크를 사용하여 눈동자를 중심 위치를 추출한 영상이다. [그림 3.2]는 실제로 마우스 커서가 움직이는지를 확인 하기 위하여 마우스 포인터를 실제 인터페이스 위에 올려 놓고 버튼 A 라는 목표를 향해서 가기 전의 모습이다. [그림 3.3]은 마우스 포인터가 버튼 A 에 도착해서 클릭하여 메시지 박스를 띄우는 이벤트를 처리한 결과이다.



[그림 3.1] 눈동자 추적



[그림 3.2] 목표를 향해 움직이는 마우스 포인터



[그림 3.3] 목표를 클릭하고 성공의 메시지가 뜬 모습

실제적으로 마우스의 움직임이나 정확도에 많은 미흡한 점을 보였다. 버튼이 작으면 커서가 움직이는 도중에 버튼을 지나가 버리기 때문에 클릭 이벤트를 못하고 지나가 버리는 문제점이 있다. 또한 현재 눈동자를 찾을 때 자동적으로 바로 눈동자를 찾는 것이 아니라 위치 크기 보정 마스크에다 눈동자를 50% 이상 맞추고 찾는 방법등에도 많은 개선이 요구된다.

4. 결론

본 논문에서는 시선 추적을 통한 입력장치에 대해서 제안 하였다. 실험결과 현재 시스템으로는 눈동자로 몇 픽셀 단위까지 세세하게 움직임을 구현 해 낼 수가 없었다. 인간이 눈을 픽셀 단위로 자유자재로 움직인다는 것이 워낙 어렵기 때문에 원하는 대로 움직이고자 한다면 약간의 훈련이 필요 할 것이라 예상된다.

눈동자를 이용해서 부드럽고 세세한 마우스의 움직임과 기타 다른 입력장치에 적용하는 방법을 만들기 위해서는 더 많은 연구가 필요 할 것이다.

참고문헌

[1] L.E. Sibert and R. J. K Jacob, "Evaluation of Eye Gaze Interaction," Proc. Of the CHI, ACM in New York, pp 281-288, 2000.

[2] Yamoto, M., Monden, A., Matsumoto, K., Inoue, K., and Torii, K, "Quick Button election with Eye Gazing for General GUI Environments," International Conference on Software: Theory and Practice, August 2000.

[3] 김재희, "감성 컴퓨터 인터페이스 기술 개발(얼굴/눈의 움직임에 의한 응시 위치 추적)," 과학기술부 보고서, 연세대학교, 2000.

[4] 이정준, 박강령, 김재희, "응시 위치 추적 기술을 이용한 인터페이스 시스템 개발," 1999 년 대한 전자공학회 하계 종합학술대회 논문집, 연세대학교, pp. 516-519, 1999.

[5] 양현승, "컴퓨터의 감성 인터페이스 기술 개발," 과학기술부 과제 최종 보고서, 한국과학기술원, 1998.

[6] 남시욱, 박강령, 정진영, 김재희 "얼굴의 칼라정보와 움직임 정보를 이용한 얼굴 영역 추출," 1997 년 대한 전자공학회 하계 종합학술대회 논문집, pp. 905-908, 연세대학교, 1997.

[7] Jacob, R and J. K, "Eye Movement-Based Human Computer Interaction Technique : Toward Non Command Interface," Advances in Human Computer Interaction, H. E. Hartson and D. Hix, Editors, Ablex Publishing Co., pp. 151-190, 1993.

[8] 정재헌, "얼굴 구성 성분 추출 및 특징 분석에 관한 연구," 석사 학위 논문, 한국과학기술원, 1991.