碩士學位論文

눈동자 추적에 의한 마우스 커서 제어

嶺南大學校 大學院

컴퓨터工學科

컴퓨터工學專攻

韓 濼 明

指導敎授 尹 英 雨

2007年 6月

碩士學位論文

눈동자 추적에 의한 마우스 커서 제어

指導敎授　尹　英　雨

이 論文을 碩士學位論文으로 提出함

2007年　6月

嶺南大學校 大學院

컴퓨터工學科 컴퓨터工學專攻

韓　濼　明

韓潔明의 碩士學位論文을 認准함

審査委員 ㊞

審査委員 ㊞

審査委員 ㊞

2007年 6月

嶺南大學校 大學院

感謝의 글

많은 인생의 시행착오를 거치며 늦은 나이에 다시 뛰어든 학업의 길에서 이렇게 석사과정을 마감하는 논문을 내게 되어 감회가 새롭습니다. 단순하고 지루한 일상으로부터 힘들고 어렵지만 도전과 열정으로 인생을 채워갈 수 있도록 저를 받아주신 윤영우 교수님께 큰 감사를 드립니다.

본 논문이 완성되기까지 끊임없는 지도와 자상하신 배려로 돌보아주신 윤영우 교수님 그리고 바쁘신 와중에도 논문심사를 맡아주시고 지도편달에 애써주신 강병욱 교수님, 손영호 교수님께 깊은 감사를 드립니다. 그리고 많은 관심과 도움을 주신 박준호 선생님, 장종원 선생님, 구자효 선생님께 감사를 드리고, 논문이 잘 진행될 수 있도록 조언을 주신 김미애 선생님께도 감사드리며, 이 외 늘 관심을 가져주신 영상기술 연구실의 선후배님들께 감사를 드립니다.

늦은 나이에 학업에 다시 복귀하였지만 저를 믿고 학업에 열중하도록 배려해 주신 부모님께 감사드리며 무엇보다 저를 끝까지 지지해 주신 바로 그 교회 지체들에게 고마움을 전하며 제가 이곳에 설 수 있도록 인도하시고 지켜주신 하나님께 이 모든 영광을 돌립니다.

목 차

그 림 목 차

수식 목차

표 목차

1. 서 론

사람들은 컴퓨터를 이용하는 자연스러운 방법의 가능성에 대해서 생각해 왔다. 키보드나 마우스처럼 손으로 제어되는 입력장치 대신에 음성이나 표정으로 컴퓨터와 자연스럽게 대화하는 방법에 대하여 관심을 가져왔다. 또 키보드나 마우스를 동시에 빈번히 사용하는 작업의 경우 마우스와 키보드를 빈번히 이동하는 것이 불편하기 때문에 사용자들은 한번쯤 작업의 효율성을 위해서 마우스를 대체할 만한 장치의 필요성에 대해서 생각해 봤을 것이다. 거동이 불편한 장애인의 경우 손을 사용하지 않고 컴퓨터와 대화 할 수 있는 방법의 필요성에 대해서 아주 절실히 깨닫고 있을 것이다.

컴퓨터를 사용할 때 사람이 어떻게 하면 편안함을 느낄까? 그것은 사람이 사람과의 의사소통을 어떻게 하고 있는지를 살펴보면 될 것이다. 사람은 대부분 언어와 표정으로 대화한다. 사람이 컴퓨터를 사용할 때 언어와 표정으로 대화하기 위해서는 컴퓨터가 음성이나 표정 정보를 획득하고 분석하여 인지할 수 있는 능력을 가져야 한다. 이러한 컴퓨터와의 자연스러운 대화는 고도로 발달된 하드웨어의 성능을 자랑하는 오늘날에도 쉬운 과제가 아니며 끊임없이 연구되어지고 있지만 잘 해결되지 않는 과제이다. 하지만 이러한 것들 중 가장 쉽게 구현 해 볼 수 있는 것이 비디오 입력 장치로 획득되는 영상에서 인간 신체의 움직임을 추적하는 것이다. 컴퓨터와 연결 되어 있는 웹 카메라로부터 실시간으로 획득된 영상에서 인간신체의 파라미터들을 구하고 그 파라미터들로 컴퓨터를 제어할 수 있다.

인간의 신체부분 중에서 얼굴은 컴퓨터와 인간의 인터페이스를 위하여 가장 많이 연구 되어져온 대상이다. 왜냐하면 얼굴은 통계적으로 색깔, 모양, 질감 정보에서 일관성을 보이며 이러한 사실은 컴퓨터가 강건하게 그리고 정확하게 얼굴을 검출하고 추적할 수 있는 특성이 된다. 또한 얼굴은 이동과 회전이 가능하므로 다양한 파라미터를 생성할 수 있다. 얼굴은 눈, 눈썹, 코, 입 등의 구성요소를 가지고 있으며 각각 이들을 검출하고 추적하는 방법들이 연구되어져 왔다. 하지만 이들 구성요소 중에서 눈동자는 가장 두드러지는 기하학적 특성을 가졌을 뿐만 아니라 색상 정보에서도 확연한 특성을 보이며 빛이나 적외선을 반사하는 물리적 특성을 나타내는 등 여러 가지 다양한 특징을 소유하고 있고 검출과 추적이 얼굴의 다른 구성 요소보다 쉽다고 인정되기 때문에 가장 활발하게 연구되어 지고 있다. 눈은 또한 얼굴 인식 분야에서도 중요한 화두이다. 얼굴 인식률을 높이기 위하여 얼굴 위치를 보정하고 얼굴의 이미지 크기를 정규화는 과정이 필요한데 이러한 것들을 수행하기 위한 기준이 바로 눈이 되기 때문이다.

현재까지 눈동자의 위치를 검출하는 여러 가지 논문들이 게재되었다. 이 논문들은 주로 알고리즘의 성능을 정확도라는 측면에서 바라보고 있다. 몇 년 전만 하더라도 하드웨어의 성능이 좋지 않았기 때문에 계산량이 많은 영상 처리 알고리즘에서 정확한 눈동자의 위치를 찾는 것만으로도 학문적 가치가 있었고 눈동자 검출을 실시간으로 할 필요성이 없었다. 하지만 눈동자의 위치를 검출하고 실시간으로 마우스 포인터를 이동하는 본 논문의 경우 눈동자 검출의 정확도뿐만이 아니라 눈동자 검출의 속도 또한 고려해야할 중요한 측면이 되었다. 불행하게도 속도와 정확도는 trade-off의 관계에 있어서 정확도를 높이고자 하면 속도가 저하되고 속도를 높이고자 하면 정확도가 떨어지게 된다.

본 논문에서는 눈동자가 기하학적 원에 근접한다는 점에 착안하여 눈동자를 추출하고 이를 수학적 모델과 비교하여 오차율을 계산하여 검출하는 새로운 눈동자 검출 알고리즘을 제안한다. 두 눈동자의 위치를 검출한 후에는 두 눈동자의 중간 지점을 계산하여 이를 이용하여 마우스 포인터를 이동시키는 방법을 고안하였다.

1963년 스탠포드 연구소의 Douglas Engelbart가 마우스를 발명한지 많은 세월이 흘렀지만 마우스에 있어서 획기적인 변화를 거친 적은 한 번도 없다. 다양한 버튼의 추가와 휠의 사용으로 마우스의 사용이 편리해지긴 했지만 여전히 사람들은 더 자연스러운 컴퓨터와의 의사소통을 원하고 있다. 또한 손을 이용해야하는 입력 장치는 손의 움직임이 자유롭지 못한 많은 장애인에게는 커다란 장벽이며 하루 빨리 극복되어져 할 과제이다. 정확도가 높고 강건한 마우스를 대체할 수 있는 장치의 발명은 컴퓨터와 인간의 의사소통에 있어서 큰 변화를 가져 올 것이다. 또한 많은 장애인들도 큰 혜택을 누릴 수 있게 되므로 유용한 연구가 될 것이다.

논 논문의 구성은 다음과 같다. 2장에서는 기존 연구 및 문제점으로 여러 가지 눈동자 검출 기법에 대해서 소개하고 장점과 단점에 대해서 논의한다. 3장에서는 마우스 포인터를 이동시키기 위한 신속하고 빠른 새로운 눈동자 검출 알고리즘을 소개하고 4장에서는 전체적인 알고리즘을 세부적으로 기술한다. 마지막으로 5장에서는 실험 및 결과 분석에 대하여 기술하며 6장에서는 결론에 관해 기술한다.

2. 기존 연구 및 문제점

웹 카메라로 얼굴 영상을 획득하고 얼굴에서 눈동자를 검출하여 마우스 포인터를 이동시켜야 하므로 눈동자 검출 기법들을 우선 살펴보도록 한다. 기존 눈동자 검출 기법들을 분류하여 특징들을 살펴본다. 마우스를 실시간으로 이동하여야 하므로 가장 중요한 요소인 정확도와 속도라는 두 가지 측면에서 논문들의 문제점을 파악해 보도록 한다.

2.1. 눈동자 검출 기법의 분류

눈동자 검출 기법은 크게 두 가지로 분류할 수 있으며 눈동자 검출에 능동적인 입장을 취하고 있는가 아니면 수동정인 입장을 취하고 있는가에 따라 능동적 눈동자 검출과 수동적 눈동자 검출의 두 가지로 분류할 수 있다.

1) 능동적 눈동자 검출

특수한 기계 장치나 조명을 추가하여 능동적으로 눈동자를 검출하는 기법들의 총칭이다. 부가적인 장치가 추가되므로 구현의 어려움과 비용적인 부담이 있지만 일단 이러한 기법으로 획득된 영상들은 아주 간단하게 처리되고 눈동자의 위치를 손쉽게 검출할 수 있다. 가장 대표적인 예가 카메라에 적외선 LED를 추가하여 적외선이 눈의 동공에서 반사되는 특징을 이용한 방법이 있다.

2) 수동적 눈동자 검출

일반 조명하에서 획득한 영상 자료만을 자지고 눈동자를 검출하는 기법들의 총칭이다. 부가적인 장치가 필요 없으므로 간단하지만 획득한 영상을 처리하는데 시간과 비용이 많이 든다. 일반적으로 가장 많이 연구되어 지고 있는 분야이다. 또 세부적으로 특징 기반 검출 방법, Template 기반 검출 방법, 신경회로망 검출 방법 등이 있다.

2.2. 눈동자 검출 알고리즘

눈동자는 얼굴에서 가장 두드러지는 물리적 수학적 기하학적 특성을 가지는 부분이며 이러한 여러 가지 특성을 이용하여 눈동자를 검출하는 방법들이 많이 연구되었다.

2.1.1. 적외선을 이용하는 방법

적외선을 이용하는 방법은 Active Eye Detection의 가장 대표적인 예이다. 적외선 LED를 카메라 축에 가까운 곳에 둘 경우 동공은 보통 혈액이 풍부한 망막에서 빛을 반사시키기 때문에 평소보다 더욱 밝게 보인다. 이런 현상은 “bright pupil effect”라고 불리는데 눈동자를 검출하고 추적하는데 아주 유용하게 이용될 수 있다[1].

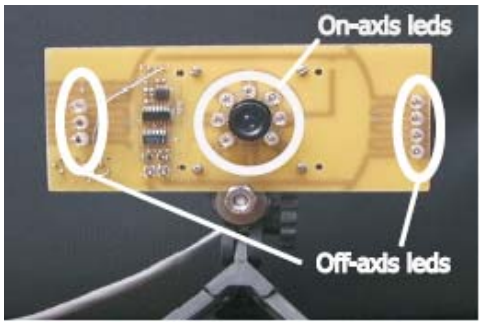


그림 2-1 적외선 조명이 부가된 카메라

그림은 2-1에서 "bright pupil effect를 발생시키기 위하여 카메라 렌즈의 축 주위로 적외선 LED를 설치한 것을 볼 수 있다. 카메라에 근접하게 설치된 적외선 LED(on axls leds)가 켜지면 눈동자가 밝게 빛나는 영상을 얻게 되고 카메라에서 멀리 떨어진 적외선 LED(off-axls leds)가 켜지면 눈동자가 빛나지 않는 영상을 얻을 수 있다. 이렇게 얻어진 두 영상에서 밝게 빛나는 눈의 동공 부분을 검출해 낼 수 있도록 먼저 영상을 흑백 영상으로 변환하고 두 영상의 차 영상을 구함으로서 눈동자를 검출할 수 있는 방법이다.

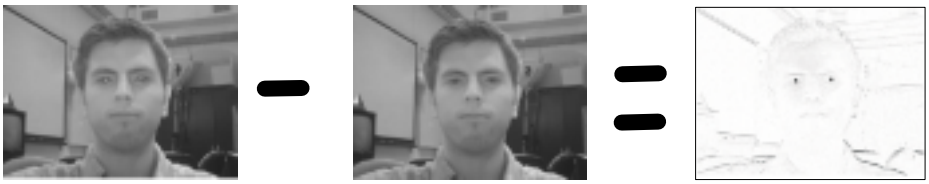


그림 2-2 적외선 조명을 이용한 눈동자 검출

이 방법을 이용하면 아주 강건하게 그리고 신속하게 눈동자의 위치를 검출하고 추적할 수 있다. 단점이라고 한다면 카메라 이외에 추가적인 적외선 장치를 구현해야 하고 동공이 빛나는 영상과 그렇지 않은 영상을 얻기 위하여 적외선 LED와 카메라를 동기화하는 것이 필요하다. 그리고 적외선은 태양 광선에도 포함되어 있으므로 실외나 태양광이 많이 들어오는 실내에서는 오동작할 소지가 있다. 그리고 유리컵이나 안경도 적외선을 반사하는 성질이 있으므로 안경을 착용한 사람이나 주위에 유리 제품이 있을 경우 눈동자 검출의 정확도가 떨어지게 된다.

2.1.2. 이미지 기울기를 이용한 방법

흑백 이미지에서 눈동자의 가장 두드러진 특징은 홍채의 타원 모양이며 홍채의 주변 영역이 극명한 명암 대비를 보인다는 것이다. 홍채부분에서 기울기의 방향은 바깥쪽으로 향하는 특징이 있다. 그림 2-3은 눈동자의 기울기 방향을 보여주고 있다.

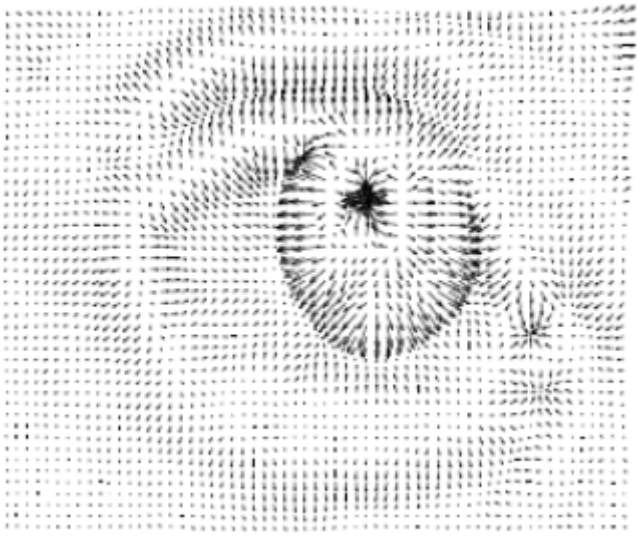


그림 2-3 눈동자의 기울기 방향

그림 2-3에서 눈동자의 중앙에 검게 나타나는 동공을 중심으로 기울기들이 바깥쪽으로 향하는 특징을 보여주고 있다. 이것은 바꾸어 말하면 기울기의 방향을 반대 방향으로 했을 때 동공에 수렴한다는 것이 되고 이를 이용하여 눈동자를 구분해 낼 수 있다[2]. 이 알고리즘에서는 2차원 배열의 bin을 사용한다.

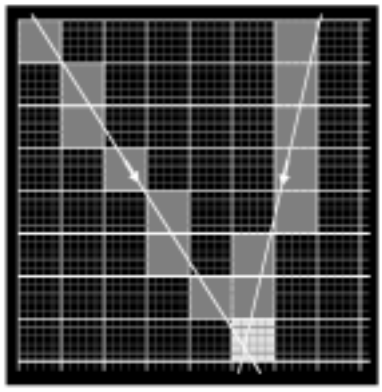


그림 2-4 bin을 이용한 기울기 수렴점 구하기

그림 2-4에서 굵게 표시된 사각형 영역은 bin영역이고 가는 실선으로 표시된 작은 사각형 영역은 픽셀이다. 각각의 픽셀에서 시작하는 기울기에 대하여 직선을 그린다. 직선이 관통하는 bin은 1씩 값들이 증가시킨다. 이런 방법으로 모든 기울기들로부터 직선을 표시하고 관통하는 빈들에 값들을 증가시키면 눈동자 영상에서 직선들이 교차하는 bin은 가장 큰 값이 쌓이게 된다. 가장 많은 축적값을 가지는 부분을 눈동자의 영역으로 검출하면 된다.

이 방법은 알고리즘이 간단하고 눈동자의 모양이나 얼굴의 방향에 크게 영향을 받지 않는다는 장점이 있다. 하지만 눈동자 영상의 크기가 올바른 결과를 낼 수 있도록 비교적 큰 사이즈여야 하며 눈동자가 빛을 전반사시키거나 사물이 눈동자에 투영될 경우 정확한 결과를 얻을 수 없으며 많은 연산과 수행시간이 필요한 것이 단점이다.

2.1.3. 투영을 이용하는 방법

투영을 이용하는 방법은 이미지의 서브 이미지 윈도우 영역에서 픽셀들의 밝기 수직합이나 수평합을 구하거나 투영함수의 도함수 등을 이용

하여 이미지가 특정한 패턴을 나타내는가를 관찰하는 방법이다. 특정한 모양과 색상을 가진 물체에서 동일한 패턴이 발견되므로 눈동자를 검출하는 데에도 이용할 수 있다.

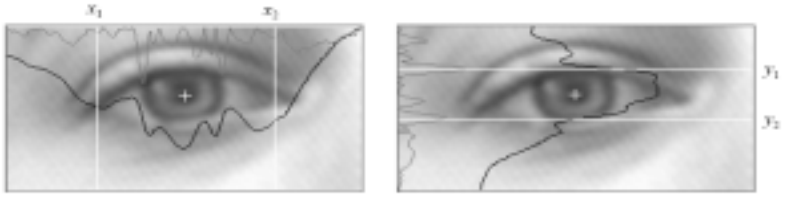


그림 2-5 눈동자의 위치를 검출하기 위하여 프로젝션 함수 사용

$$IPF_v(x) = \int_{y_1}^{y_2} I(x,y)\,dy \qquad (3\text{-}1)$$

$$IPF_h(y) = \int_{x_1}^{x_2} I(x,y)\,dx \qquad (3\text{-}2)$$

그림 2-5에서 검은 색으로 보이는 선은 식 (3-1)과 (3-2)를 이용하여 구한 것이고 회색의 선은 이들 프로젝션 함수의 도함수 값이다.
이미지의 영역들의 경계에서 투영함수의 도함수 값들이 증가하므로 투영함수를 이용하면 쉽게 이미지 영역의 경계를 검출할 수 있으며 투영함수는 특정한 대상에 대하여 동일한 패턴의 유형을 보이므로 눈동자를 검출하는 데에 이용될 수 있다[3].

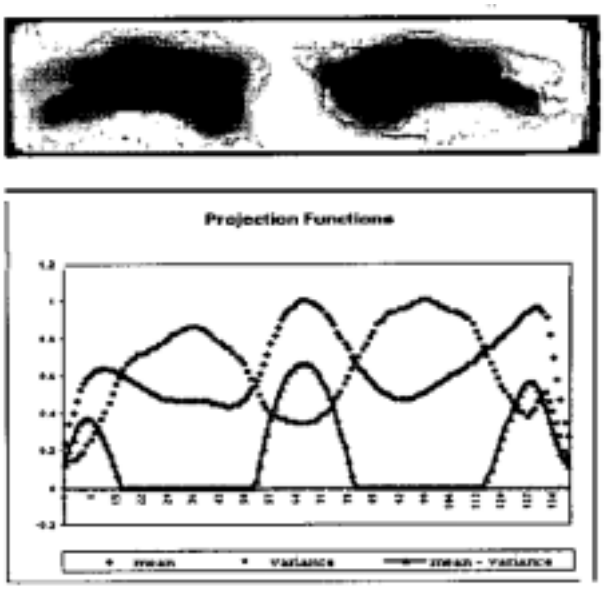


그림 2-6 수직투영 값의 평균과 분산값을 이용한 눈동자 검출

그림 2-6은 먼저 영상에 Threshold 값을 적용하여 이진화 시키고 눈의 후보 영역을 찾은 후에 두 눈이 포함된 윈도우에서 수직으로 값들을 투영시키고 그 값들의 평균과 분산을 이용하여 투영의 패턴을 관찰한 경우이다[4]. 이 투영 방법의 장점은 다양한 크기의 이미지에 적용할 수 있다는 점이다. 단점으로는 눈동자와 눈꺼풀이 움직이므로 다양한 패턴을 고려해야 하고 얼굴의 각도가 정면을 향하고 있을 때 가장 잘 검출되며 계산량이 다소 많고 얼굴의 크기가 큰 이미지가 필요하다는 점이다.

2.1.4. Template를 이용하는 방법

Template를 이용하는 방법은 경계선을 검출한 영상에서 Circle Hough Transform을 이용하여 원을 구성하는 픽셀들만 골라내고 여기에 아래의 그림과 같은 원형의 틀을 컨볼루션하여 눈동자를 검출해 내는 방법이다[5].

그림 2-7 얼굴 영상에서 경계선을 검출

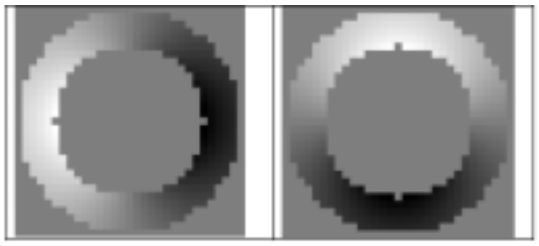


그림 2-8 눈동자를 검출하기 위한 Template

이 방법은 아주 정확하고 빠른 수행 시간을 보여준다. 하지만 사람마다 눈동자의 크기가 다양하며 카메라와 사람의 거리에 따라 눈동자의 크기가 달라진다는 사실을 고려하고 있지 않다. 또 눈이 작은 사람은 눈동자가 원형이 아니기 때문에 이 방법으로는 잘 검출할 수가 없게 된다.

2.3. 대체 마우스 사례

2.3.1. 얼굴의 3D 모델을 이용하는 방법

이 방법은 그림 2-7과 같이 얼굴을 3D로 모델링하여 얼굴의 회전, 이동뿐만 아니라 얼굴의 표정 정보를 획득하여 마우스를 움직이는 방법이

다. 얼굴의 이동 회전 정보로 마우스를 움직이며 표정 혹은 눈이나 입의 개폐 상태를 추적하여 마우스 버튼의 클릭 기능을 구현한다[6]. 이 방법은 아주 복잡한 3D 모델링 기법을 사용하고 있기 때문에 상당히 큰 연산시간을 요구하며 강건하게 얼굴을 추적하는 것이 불가능하여 논문으로 제시되었지만 실용화에는 많은 개선의 여지가 있다.

그림 2-9 얼굴 3D 모델링

2.3.2. 다층 신경망을 이용한 방법

이 방법은 그림 2-10과 같이 57개의 입력노드 33개의 은닉층 노드 1개의 출력 노드로 구성된 신경망 회로에 전체 영상에서 작은 윈도우 영역의 영상을 입력하여 눈동자의 위치를 검출하는 방법이다[7]. 결과는 [0, 1] 사이의 확률값으로 출력되며 만약 결과 값이 0.5보다 크면 눈동자로 고려되어진다. 다층 신경망의 입력은 그림 2-11에서 보는 것과 같이 작은 윈도우의 모든 값을 입력하는 것이 아니라 자동회귀 특징들 때문에 회색으로 표시된 부분만을 입력한다.

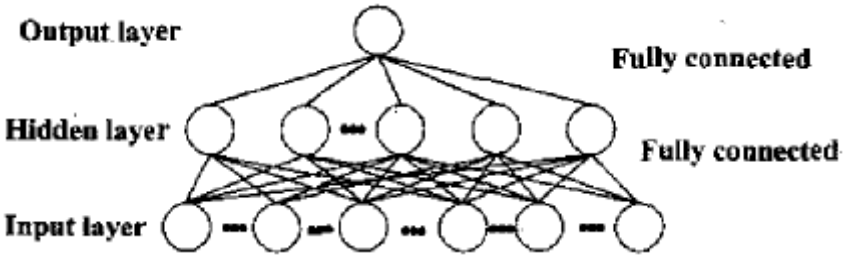


그림 2-10 눈동자 인식을 위한 다층 신경망

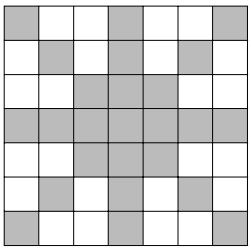


그림 2-11 입력 구성 : 회색 부분이 신경망의 입력으로 이용

일단 눈동자가 검출되면 다음 프레임에서는 mean shift algorithm으로 눈동자를 추적하여 눈동자 검출 시간을 줄인다. 마우스 컨트롤은 두 눈동자의 중심을 계산하여 윈도우 시스템으로 데이터를 전송하고 만약 1초 이상 두 눈동자의 중심 좌표에 변화가 없으면 마우스 버튼을 클릭한 것으로 간주 한다. 이 방법은 빠르게 실시간으로 눈동자를 검출 할 수 있으나 신경망 회로를 학습시키는 것이 가장 큰 난제이다. 모든 사람에게서 동일한 성능을 보이도록 신경망 회로를 학습시키는 것이 상당한 데이터와 노력을 요구하기 때문이다. 그리고 그림 2-11에서 보듯이 입력 윈도우의 크기가 고정되어 있어 다양한 눈동자의 크기에 대해서 검출할 수 없다는 단점도 있다.

2.3.3. 전기 안구도(electrooculogram)를 이용하는 방법

전기 안구도를 이용하는 방법은 눈 주위에 전극을 부착하고 눈동자가 움직일 때 발생하는 미세한 전압을 감지하여 마우스를 움직여 주는 기법이다[9]. 그림 2-12와 같이 마우스의 좌우 움직임을 위해 양쪽 관자놀이 부분에 전극을 부착하며 상하 마우스 움직임을 위해서는 미간과 턱 중앙에 전극을 부착하여 미세 전압을 증폭하여 마우스를 움직이는 신호로 사용한다. 이 방법은 단순히 눈동자의 움직임으로서 마우스를 이동시킬 수 있다는 점에서 유리하나 전극을 피부에 직접 붙여야한다는 점에서 상당히 불편한 점이 있다. 특히 여름과 날씨가 무더운 계절엔 땀으로 인해 전극이 쉽게 떨어지며 땀에 의해 미세 전압 감지가 어려울 수 있고 마우스를 장시간 사용 할 경우 피부에 부착한 전극 때문에 피부 트러블이 일어날 수 있는 단점이 있다.

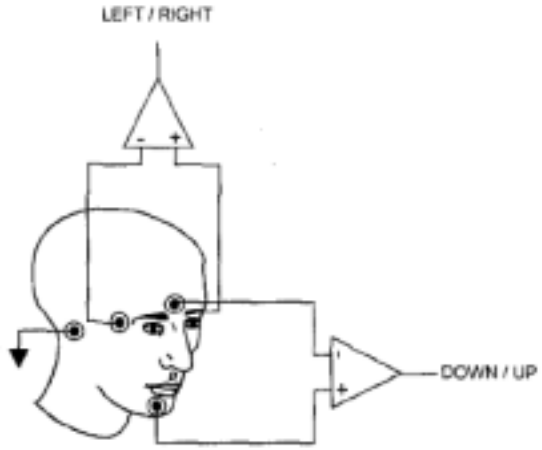


그림 2-12 전기 안구도를 이용한 마우스 이동

2.4. 눈동자 검출을 이용한 마우스 포인터 이동 시스템의 고려사항

카메라로 얼굴 영상을 입력하여 마우스 포인터를 이동시키는 시스템을 구현할 때 여러 가지 사항을 고려할 필요가 있다. 우선 이 시스템이 영상 처리를 기반으로 하는 시스템이란 점을 고려해 보자. 영상 처리는 엄청나게 다양한 입력 조건하에서 완벽한 결과가 아니라 최적의 결과를 얻어 내는 학문이다. 따라서 정확도가 100%인 결과는 영상처리에서 드문 경우가 된다. 보통은 동작 조건을 명시적으로 규정하고 그 조건하에서 최적의 성능을 나타냄을 증명한다. 마우스 포인터 이동 시스템도 마찬가지다. 여러 가지 환경적 요소가 시스템의 성능에 영향을 미칠 수 있다. 여기서 시스템의 성능에 영향을 미칠 수 있는 요소를 논의해 보면 다음과 같다.

(1) 주변 밝기

영상 처리 어플리케이션에 있어서 밝기는 아주 민감한 요소이다. 주로 색상을 이용하여 데이터를 처리하는 경우 반드시 고려해 주어야할 사항이다. 밝기의 영향을 제거하기 위해 여러 가지 다른 컬러 모델을 이용하기도 한다. 하지만 기본적으로 주변 밝기가 상당히 어두워지면 물체가 본래의 색을 잃어버리기 때문에 적당한 조명상태를 유지해 주어야 한다. 눈동자 검출에서도 눈동자 검출 이전 단계에서 얼굴 검출을 하는데 피부색의 통계적 특성을 이용하여 얼굴을 검출하는 방법이 주로 쓰이기 때문에 양호한 조명을 유지해 주는 것이 필요하다.

(2)카메라와 얼굴의 거리

카메라와 얼굴의 거리에 따라 캡처되는 물체의 크기가 달라진다. 물

체의 크기는 영상처리에서 반드시 고려되어져야 하는 요소이다. 특정 알고리즘에 있어서 영상의 크기가 너무 작거나 너무 크다면 알고리즘이 제대로 동작하지 않을 가능성이 높으므로 알고리즘이 동작하는 영상의 크기를 명시하고 제한할 필요가 있다. 눈동자 검출에서도 적당한 눈동자 크기의 확보가 아주 중요함으로 카메라와 얼굴의 거리를 명시함으로써 알고리즘이 가장 잘 처리해 낼 수 있는 눈동자의 크기를 연구해 두는 것이 필요하다.

(3)카메라와 얼굴의 각도

얼굴의 각도가 달라지면 얼굴의 눈, 코, 입의 모양과 크기가 달라지게 된다. 눈동자의 검출에서 있어서 눈의 모양은 아주 중요한 요소가 된다. 특히 템플릿을 이용하는 방법에서 눈의 모양의 변화는 성능의 저하로 결부되기 때문에 얼굴의 각도에 대한 고찰이 필요하다. 모든 각도에서 좋은 눈동자 검출 성능을 보이는 알고리즘은 템플릿이나 투영의 기법을 이용하지 않은 방법들 일 것이다.

3. 제안하는 눈동자 검출과 마우스 컨트롤 방법

기존에 눈동자 검출 알고리즘의 단점은 속도가 늦거나 다양한 눈동자의 크기와 모양에 취약하다는 것이었다. 속도와 정확도의 두 가지 측면이 모두 잘 고려되어질 때 눈동자 검출을 통한 마우스 포인터 이동 시스템이 신뢰할 수 있는 성능을 보이게 된다. 이 장에서는 투영기법을 응용한 신속하고 정확한 새로운 눈동자 검출 기법을 제안하고 이 기법 속도와 정화도 측면에서 어떤 이점을 가졌는지 소개한다.

3.1. 제안하는 눈동자 검출 방법

신체의 모든 기관 중 눈동자는 가장 기하학적 원에 근접하는 기관이다. 이는 눈동자의 원형적 모양을 잘 추출한 후 수학적 원의 형태와 비교하면 눈동자의 위치를 손쉽게 검출할 수 있는 가능성을 제시해 준다. 하지만 눈의 개방 상태에 따라 눈동자는 완전한 원형의 형태로 나타나지 않고 반원이나 혹은 상하가 잘린 형태로 나타나는 경우가 많다. 반원의 경우 원의 수직 중심을 검출하고 잘린 부분을 제외하고 나머지 부분만을 비교하고 상하가 잘린 눈동자의 경우 눈동자의 중심에서 잘린 부분까지만 비교함으로서 완전하지 않은 눈동자의 형태에 대해서도 검출 효율을 높일 수 있다는 장점이 있다. 기존에 제시된 논문들에서는 눈동자를 완전한 원으로만 모델링하여 문제 해결에 접근 했지만 본 논문에서는 눈동자의 이런 특성을 고려한 것이 기존의 논문과 가장 차별된 부분이라 하겠다.

하지만 검출된 눈동자와 수학적 원의 비교는 상당히 복잡한 과정이 필요하게 된다. 왜냐하면 수학적 원의 매칭은 2차원 공간상에서 이루어지기 때문이다. 하지만 원의 투영함수를 도입하면 비교 공간이 2차원에서 1차원으로 떨어지게 되므로 비교가 용이하게 된다. 다음절에서는 원의 투영함수의 수학적 모델을 제시한다.

2.4.1. 눈동자의 수평 투영함수의 수학적 모델링

눈동자의 투영함수는 수평 투영함수와 수직 투영함수 두 가지가 있다. 눈동자를 검출하기 위해서 원의 수직 투영함수가 아니라 원의 수평 투영함수를 사용한 것은 눈꺼풀과 눈두덩에 의해서 눈동자가 상하로 잘렸을 때 비교의 범위를 제한하기가 원의 수평 투영함수가 용이하기 때문이다.

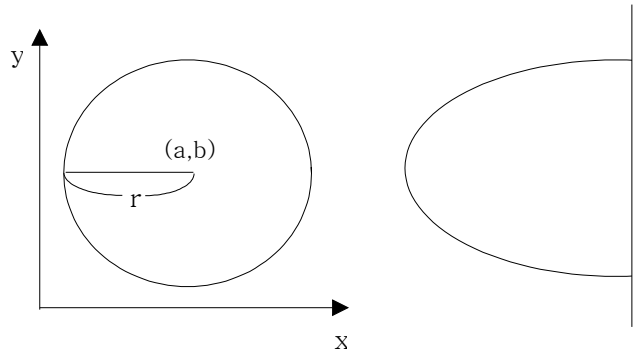


그림 3-1 원의 수평투영

눈동자는 그림 3-1과 같이 중심의 좌표가 (a,b)인 원으로 나타낼 수 있다.

$$(x-a)^2+(y-b)^2=r^2 \qquad (3-1)$$

위의 원을 수평으로 투영하면 그림 3-1의 오른쪽과 같은 그래프가 그려진다.

원으로부터 수평 투영함수의 식을 얻기 위해서는 그림 3-2와 같이 y=c인 직선을 원에 교차시키고 원의 외곽선과 교차하는 두 점 P1, P2 사이의 거리를 구하면 된다.

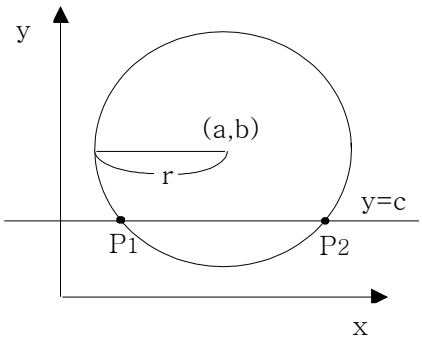


그림 3-2 원의 수평 투영 함수 구하기

따라서 y에 c를 대입하면 식 3-4와 3-5 같이 두 점 P1, P2를 구할 수 있다.

$$(x-a)^2+(c-b)^2=r^2 \qquad (3-2)$$

$$x=\pm\sqrt{r^2-(c-b)^2}+a \qquad (3-3)$$

$$P_1=(\sqrt{r^2-(c-b)^2}+a,\ c) \qquad (3-4)$$

$$P_2=(-\sqrt{r^2-(c-b)^2}+a,\ c) \qquad (3-5)$$

두 점 P1, P2 사이의 거리를 구하면 원을 수직으로 투영한 함수 PF가 된다.

$$PF = \sqrt{(\sqrt{r^2-(c-b)^2}+a+\sqrt{r^2-(c-b)^2}-a)^2} \qquad (3-6)$$

$$PF = 2\sqrt{r^2-(c-b)^2} \qquad (b-r \le c \le b+r) \qquad (3-7)$$

3.1.1. 눈동자의 수직 중심 찾기와 비교 범위

수직 중심을 찾아야 하는 이유는 눈동자의 수평 투영함수에서 수학적 투영값을 구해내기 위해서 눈동자의 반지름이 필요하기 때문이다. 이미지에서 눈동자는 완전한 기하학적 원이 아니다. 따라서 눈동자의 기학학적 중심이 항상 눈동자의 중심은 아닌 것이다. 그림 3-3에서 보듯이 눈이 작은 사람은 눈꺼풀에 의해서 눈동자가 상하로 절단될 수도 있다. 따라서 반지름을 구해 내는 것은 쉬운 일이 아니다. 직관적으로 볼 때 눈동자의 상부와 하부가 눈꺼풀이나 눈두덩에 의해서 잘릴 수 있기 때문에 수평투영 값 h(r)이 가장 클 때 이것을 원의 지름으로 인정하고 반지름을 구할 수 있으나 실제 눈동자의 모양은 기하학적 원이 아닐 수도 있기 때문에 여분의 검증을 통해서 수직 중심을 찾아내고 이것으로부터 반지름을 계산해내는 과정이 필요하다. 아래는 눈동자의 수직 중심 찾기 알고리즘을 제안하였다.

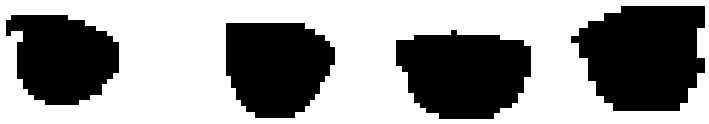


그림 3-3 여러 가지 눈동자의 종류

●눈동자의 수직 중심 찾기 알고리즘

1. 모든 r에 대하여 수평 투영값 h(r)을 구한다.
2. h(r)의 값 중 최대값을 갖는 Rmax을 일단 눈동자의 수직중심이라고 가정한다.

3. 수직 중심에서 상단까지 길이Lh와 반지름 h(Rmax)/2를 비교하여 Lh가 크면 Rmax 값을 1만큼 감소시킨다.

4.수직중심에서 하단까지 길이 Ll과 반지름h(Rmax)/2을 비교하여 Ll가 크면 Rmax 값을 1만큼 증가시킨다.

5. Lh가 h(Rmax)/2보다 크거나 Ll가 h(Rmax)/2보다 크면 3과 4를 반복한다.

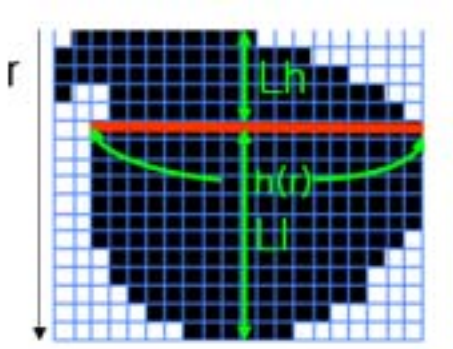


그림 3-4 눈동자의 수직 중심 찾기

이렇게 눈동자의 수직 중심을 찾은 후 한 가지 더 고려해야 할 사항은 눈동자가 반원이나 상하로 눈동자가 잘린 경우 비교 범위를 어떻게 할 것인가 하는 것이다. 눈동자가 기하학적 원에 근접하게 추출되었다면 비교의 범위를 전체 즉 눈동자의 지름만큼 하면 되지만 눈꺼풀에 의해서 반쯤 가려진 눈동자나 상하로 눈동자가 잘린 경우 눈동자의 전체에 대해서 오차를 계산하면 오차가 상당히 커지게 되어 실제로 눈동자임에도 불구하고 눈동자가 아니라고 판명될 가능성이 높아진다. 따라서 본 논문에서는 눈동자의 오차 비교 범위를 눈동자의 수직 중심으로부터 눈동자가 실제 존재하는 부분까지로 제한한다. 이렇게 하면 부분 눈동자를 계산하게 되므로 눈동자의 검출 확률을 높이게 된다.

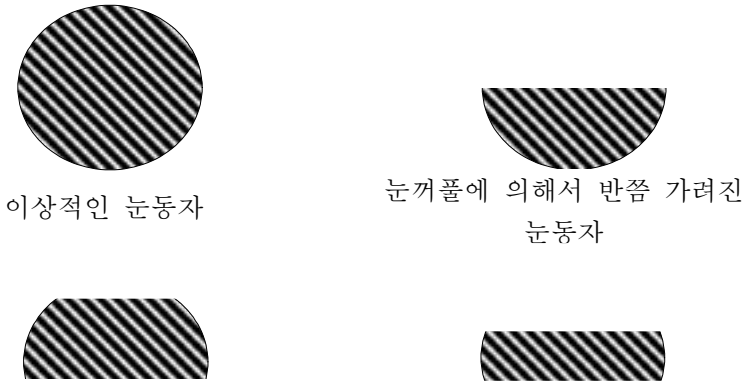


그림 3-5 여러 가지 눈동자의 종류

3.1.2. 눈동자 주변부 정리하기

영상에서 눈동자가 검은 색에 가깝다는 것을 이용하여 눈동자를 추출해 보면 그림 3-6과 같이 눈동자와 눈꺼풀의 경계부분이 검출된다. 눈꺼풀과 눈동자의 높이 차이가 존재하기 때문에 조명에 의해서 그림자가 생기게 되는 것이다. 본 논문에서 제시한 방법은 눈동자 영역만을 필요로 하기 때문에 불필요하게 검출되는 이런 눈동자 주변부분을 제거하는 과정이 필요하게 된다.

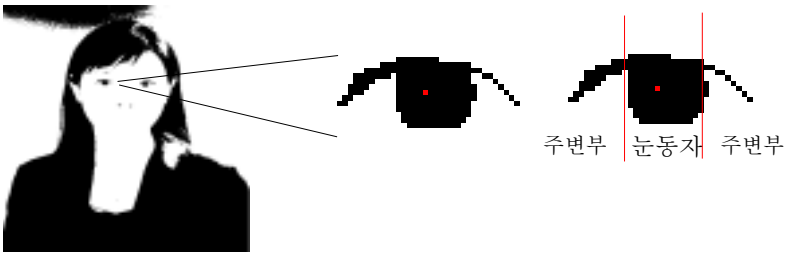
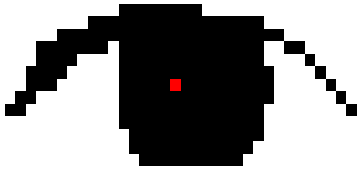
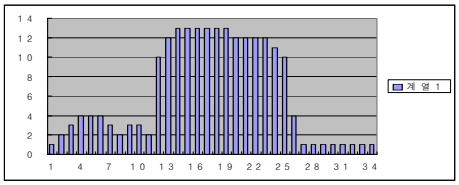


그림 3-6 눈동자 주변부

눈동자의 주변부의 제거 방법은 눈동자를 수직으로 투영한 v(c) 값을 구해 보면 그림 3-7의 하단 그래프와 같이 주변부와 눈동자의 경계에서 이웃한 두 투영값 v(c)가 큰 차이를 보인다는 점을 이용하면 된다. 눈동자 수평 중심을 기준으로 왼쪽과 오른쪽에서 두 투영값 v(c)의 차이가 가장 큰 부분을 검출하면 주변부를 절단해 낼 수 있게 된다. 아래는 눈동자 주변부 알고리즘을 기술하였다.



눈동자



눈동자를 수직 투영한 그래프

그림 3-7 눈동자 주변부 정리 원리

●눈동자 주변부 정리 알고리즘

1) 눈동자의 x축 중심 구하기
   (1) x축의 원점부터 검색하여 수직 투영값의 최대값을 구한다.
   (2) 최대값의 위치가 전체 영역의 중앙 영역에 위치하는지 점검한다.
2) 눈동자의 x축 중심의 양쪽 방향으로 기울기가 가장 큰 지점을 찾는다.
   (1) 기울기(인접값의 차)를 구한다.
   (2) 눈동자 x축 중심에서 x축 왼쪽 방향으로 기울기 양의 값 중

가장 큰 값을 구한다.

(3) 눈동자 x축 중심에서 x축 오른쪽 방향으로 기울기 음의 값 중 가장 큰 값을 구한다.

3) 1)과 2)에서 구한 값을 기준으로 눈동자 주변부를 제거한다.

3.2. 처리 과정

3.2.1. Eye Mouse System 전체 순서도

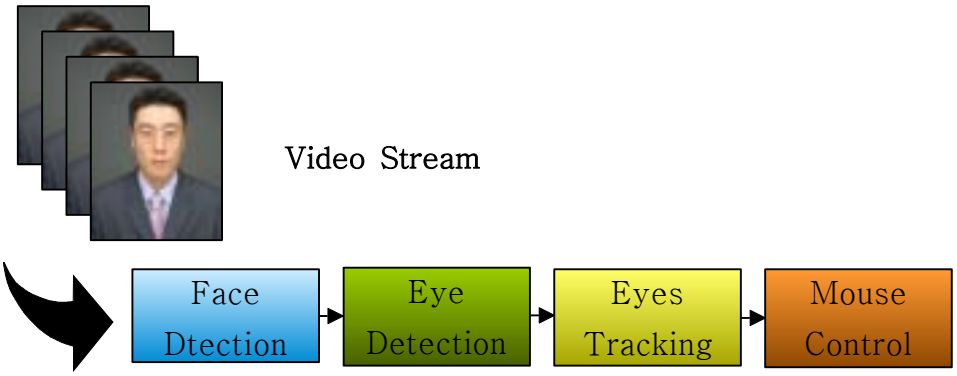


그림 3-8 Eye Mouse System

우선 마우스 포인터를 제어하는 이 시스템을 Eye Mouse라고 부르자. 웹 카메라에서 얼굴 영상을 획득하여 마우스 포인터를 이동시키므로 Face Mouse라고 이름을 지는 것도 가능하겠지만 엄밀히 눈동자의 위치를 검출하고 그것으로 마우스 포인터를 이동하는 것이므로 Eye Mouse라고 호칭하는 것이 더 적당할 것이다. Eye Mouse의 첫 번째 단계는 웹 카메라에서 비디오 스트림을 입력받는 것이다. 입력 영상의 크기는 클수록 좋지만 본 논문에서는 352x288 영상을 사용하였다. 두 번째 단계는 Face Detection인데 전체 영상에서 얼굴 영역만을 추출하여 눈동자의 검색 범위를 축소시켜 눈동자 검출을 빠르게 한다. 그리고 Eye Detection 단계를 거치면서 얼굴 영역에서 한쪽 눈동자를 검출하고 Eyes Tracking 단계에서 나머지 다른 한눈동자를 검출하고 두 눈동자 사이의 거리의 중심을 구하여 이를 이용하여 Mouse Control 단계에서 마우스를 제어하게 된다.

3.2.2. Face Detection

Face Detection 단계는 그림 3-9와 같이 전체 입력 영상에 대하여 얼굴이 포함된 영역만을 추출해 내는 과정이다. 이렇게 하는 이유는 눈동자를 검출하는데 걸리는 시간을 줄이기 위한 목적이 있다. 그리고 배경 부분을 제거함으로서 눈동자를 검출하는 과정을 단순화하여 눈동자의 검출 정확도를 높이기한 목적도 있다.

그림 3-9 얼굴 영역 추출

Face Detection 단계는 그림 3-10과 같이 세 단계로 나누어 질 수 있다.

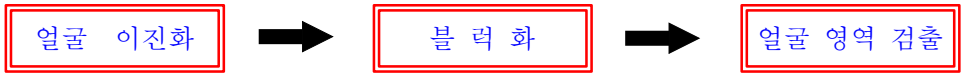


그림 3-10 Face Detection의 세부 단계

1) 얼굴 이진화

얼굴 이진화라고 하는 것은 영상의 각 픽셀들이 피부 색깔 범위에 속하는지 검사하여 피부 색깔에 속하면 1로 그렇지 않으면 0으로 이진화하는 것이다. 피부 색깔을 검출하는 데는 RGB Color Model을 쓰는 것이 아니라 A-B 색상 공간 모델을 이용한다[8]. A-B 색상 공간 모델은 식 3-12와 같다.

$$r = \frac{R}{R+G+B} \qquad (3-8)$$

$$g = \frac{G}{R+G+B} \qquad (3-9)$$

$$a = r + \frac{g}{2} \qquad (3-10)$$

$$b = \frac{\sqrt{3}}{2} g \qquad (3-11)$$

피부 색깔은 인종마다 분포의 차이가 있다. 따라서 모든 인종에 대해서 적용될 수 있는 피부 색깔의 분포는 값으로 결정해 놓지 않고 여러 논문에서는 임의의 실험 영상들에서 A-B 모델 상의 피부 색깔의 통계적 분포를 구하고 그 값으로 얼굴 이진화를 수행한다. 본 논문에서도 실험 영상으로부터 A-B 값의 실험적 평균과 표준 편차를 구하였고 얼굴 이진화의 범위를 아래와 같이 정의하여 이진화 하였다.

얼굴 이진화 Threshold 값의 분포
ave_a − δ_a ≤ a ≤ ave_a + δ_a ave_b − δ_b ≤ b ≤ ave_b + δ_b ave_a, ave_b : 피부색깔의 평균 δ_a, δ_b : 피부 색깔의 표준편차

표 3-1 얼굴 이진화 범위

얼굴을 이진화 하면 그림 3-11과 같은 결과를 얻을 수 있다. 얼굴 영역은 눈이나 눈썹 입술 등은 피부 색깔과 다르고 또 얼굴에서 그늘이 진 부분은 어둡게 나타나므로 얼굴 영역 안에서도 피부가 아니라고

판명된 무수한 많은 픽셀들이 있지만 얼굴 영역이 뚜렷하게 검출된 것을 볼 수가 있다.

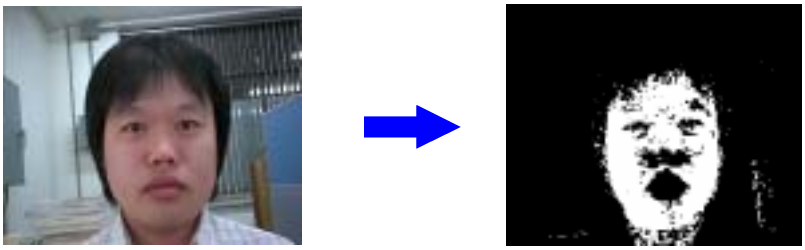


그림 3-11 얼굴 이진화

2) 블록화

얼굴 이진화를 거치고 난 영상을 살펴보면 얼굴 내부에 얼굴영역임에도 불구하고 피부로 분류되지 않은 많은 픽셀들이 존재한다. 특히 눈이나 코 구멍, 얼굴의 반점 부분은 얼굴 영역 내에 존재하지만 피부 색깔의 색상 범위에 들지 않으므로 피부로 인정되지 않는다. 특히 얼굴의 경계 부위에서는 피부로 인정되는 픽셀과 그렇지 않은 픽셀들이 혼재되어 있다. 따라서 임의의 크기 NxN 픽셀 블록을 설정하고 그 블록에서 피부로 인정된 픽셀이 과반수이상이면 그 블록을 얼굴 영역으로 분류하는 방법을 사용한다.

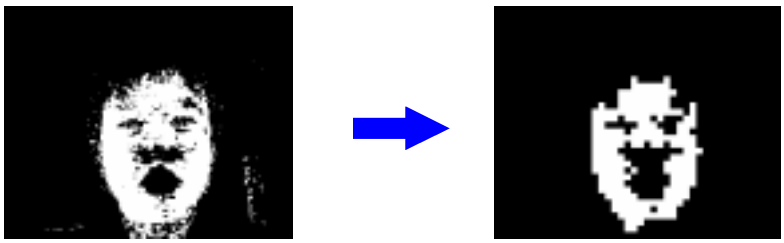


그림 3-12 블록화

3) 얼굴 영역 검출

얼굴 블록화 단계를 거치고 나면 얼굴의 경계가 명확히 구분되므로 얼굴 영역을 추출할 수 있다. 얼굴 영역 내부에 눈, 콧구멍, 입이 피부색깔로 인정되지 않아도 얼굴 영역이 틀림없으므로 피부 영역으로 판정된 모든 블록들을 포함하는 사각 영역을 얼굴 영역으로 추출하도록 한다.

3.2.3. Eye Detection

Eye Detection은 추출된 얼굴 영역에서 눈동자 후보 영역을 추출해 내고 후보 영역들 중에서 눈동자 영역을 검출해내는 과정이다. Eye Detection의 과정은 5단계로 이루어진다.

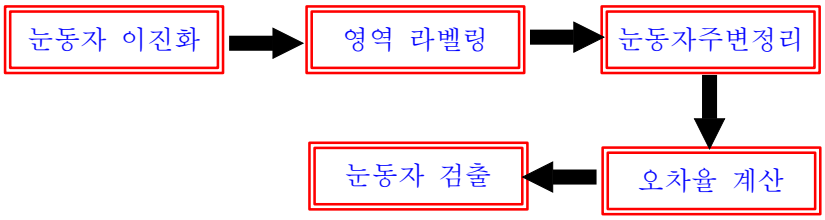


그림 3-13 Eye Detection의 세부 단계

1) 눈동자 이진화

눈동자는 검은 색을 많이 포함하고 있어서 얼굴의 다른 부분과 색상에서 뚜렷한 차이가 있다. 눈동자 이진화도 얼굴 이진화와 마찬가지로 눈동자의 색깔 범위를 미리 조사하여 각각 RGB의 Threshold 값을 구하여 수행한다. 눈동자는 검은 색이 주요한 색이므로 RGB 컬러 모델을 사용한다. 본 논문에서는 실험 영상으로부터 육안으로 판별하여 눈동자 영역을 추출해 내고 눈동자 영역의 색상 분포의 히스토그램을 구하여 밝기 0에서 누적값 80%인 지점을 눈동자 색깔 범위로 하였다. 이렇게

누적값 80%을 정한 이유는 실험 영상에서 가장 눈동자를 뚜렷하게 구분할 수 있었기 때문에 실험적으로 정한 값이다.

이렇게 실험적으로 구한 값을 이용하여 영상을 이진화 해 보면 그림 3-14와 같이 여러 개의 영역들이 나타나는데 그 중에 눈동자의 영역이 두 군데 뚜렷이 나타나는 것을 관찰할 수 있다.

그림 3-14 눈동자의 이진화

2) 영역 라벨링

눈동자의 이진화 과정을 거치고 나면 여러 가지 영역들이 나타내게 된다. 각각의 영역들에 대하여 눈동자의 영역인지 조사하는 단계를 거쳐야 한다. 하지만 이들 영역 상의 픽셀들은 육안으로 봐서는 같은 영역이지만 같은 영역인지 다른 영역인지의 정보가 없다. 따라서 인접한 픽셀들을 동일 영역으로 묶어 구분해 주는 작업이 필요하다. 이것을 영역 라벨링이라고 하며 본 논문에서는 Grass fire 라벨링 알고리즘을 사용하였다. 그림 3-15에서 영역들은 각기 다른 색깔들로 표시 되었다. 동일한 색깔로 표시된 픽셀들은 같은 영역임을 의미한다. 주로 머리카락 영역이 가장 넓은 영역을 차지한다.

그림 3-15 영역 라벨링

3) 눈동자 주변 정리하기

본 논문은 눈동자만을 추출하여 실제 영상에서 추출된 눈동자를 눈동자 수평 투영 함수 PF와 비교하여 오차를 계산한다. 하지만 실제 영상을 이진화 시켜보면 그림 3-15에서의 두 눈동자 영역에서와 같이 눈동자 이외에 눈동자 주변부가 같이 검출되게 된다. 눈동자 수평 투영함수 PF와의 오차를 줄이고 눈동자 검출의 확률을 높이기 위해서는 눈동자의 주변부를 정리하는 과정이 꼭 필요하다. 눈동자의 주변부를 정리하는 알고리즘은 3.1.1에서 제시한 알고리즘을 사용한다.

4) 오차율 계산

앞 단계에서 영역들이 잘 준비되었다면 영역의 넓이가 Amin < Area < Amax인 범위에 대하여 실제로 영역이 눈동자 영역인지 아닌지 검출하기 위해서 오차를 계산한다. 잡영과 머리카락 영역을 제외시키기 위하여 적당한 Amin과 Amax를 선정하고 이 범위를 벗어나는 영역들은 오차율을 계산하지 않는다. 이런 영역들을 제외하는 것은 눈동자 검출의 확률을 높일 뿐만 아니라 계산의 속도도 빠르게 할 수 있다. 오차를 계산하기 위해서는 먼저 눈동자 영역의 수직 중심을 구하여야 한다. 눈

동자의 수직 중심을 구하면 눈동자의 수직 중심을 기준으로 실제 이미지를 수평 투영시킨 값과 눈동자 수평 투영함수에서 구해진 두 값을 구할 수가 있고 두 값의 오차를 구한다. 아래는 실제 추출된 눈동자에 대해서 오차율을 구하는 과정은 나타내었다.

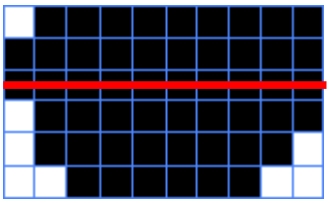


그림 3-16 추출된 눈동자 영역

r	이미지 투영값 IP(r)	수평 투영 함수값 PF(r)	오차
1	9	9.16515	0.16515
2	10	9.79795	0.20204
3	10	10	0
4	9	9.79795	0.79795
5	8	9.16515	1.1651
6	6	8	2
합계		55.9262	4.332029
오차율			0.077

표 3-2 그림 3-16의 오차율 계산 예제

눈동자와 눈동자 수평 투영 모델이 얼마나 유사한지를 알기 위해서 오차율을 구하는데 눈동자는 그 크기가 다를 뿐만 아니라 모양이 다양

하기 때문에 아래와 같이 오차율 공식을 제안한다.

$$\text{오차율} = \frac{\sum_{r} |IP(r) - PF(r)|}{\sum_{r} PF(r)} \qquad (3-12)$$

$IP(r)$: 이미지 투영값
$PF(r)$: 눈동자 수평 투영함수값

5) 눈동자 검출

오차율 계산 단계에서 각 영역에 대하여 오차율이 계산되었다면 영역들 중 가장 낮은 오차율을 가진 영역을 눈동자의 영역으로 선정한다. 그림 3-17에서 검은 색으로 표시된 부분은 Amin과 Amax내에 존재하는 눈동자 후보 영역으로 고려된 부분이며 이 중에서 오차율이 가자 작은 Blob 4번이 눈동자로 검출 됨을 볼 수 있다.

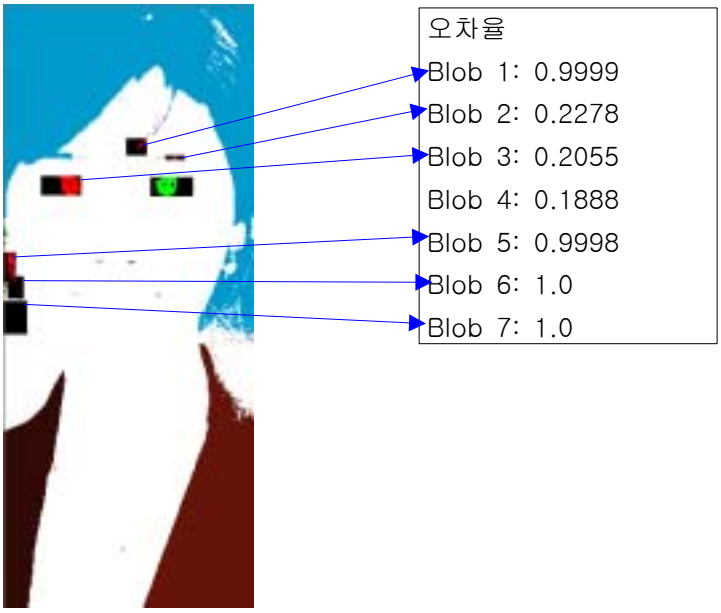


그림 3-17 눈동자 검출

눈동자 검출 도중 고려해야 할 것은 얼굴을 상단으로 향할 때 콧구멍이 검출되는데 콧구멍은 눈동자보다 원에 더 근접하기 때문에 콧구멍이 검출되지 않도록 콧구멍과 눈동자를 구분하여야한다. 본 논문에서는 눈의 영역의 종회비가 특정 AspectRatio 이상이라는 점을 이용하여 영역의 종회비가 특정 AspectRatio 이하인 영역은 눈동자 선정 후보군에서 제외하도록 하여 콧구멍이 눈동자로 오인되는 일이 없도록 하였다.

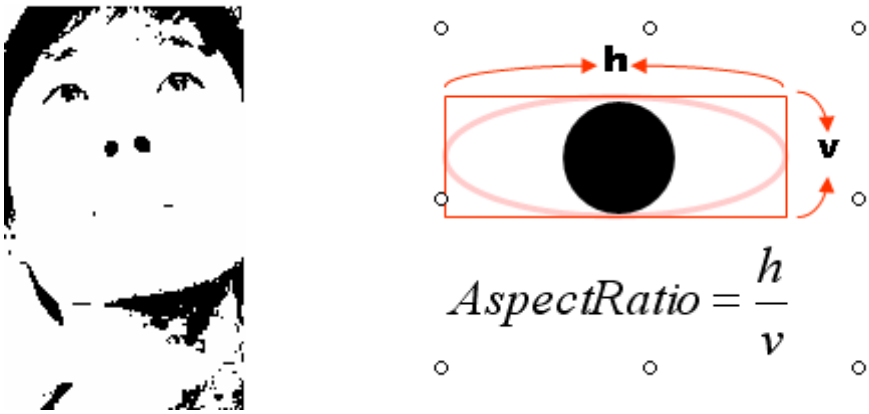


그림 3-18 콧구멍 검출 방지를 위한 종횡비 정의

3.2.4. Eyes Tracking

이 단계는 검출된 한 쪽 눈동자를 기준으로 다른 나머지 눈동자를 찾는 단계이다. 다른 눈동자를 검출할 때는 휴리스틱 정보를 이용한다.

●눈동자 검출을 위한 휴리스틱 정보

1)두 눈동자는 인접한 영역에 존재

2)눈동자는 동일 직선에 위치

3)눈동자는 서로 대칭

검출된 눈동자를 기준으로 상하 임의의 SearchRange 픽셀 범위를 검색 범위로 하는데 초당 20 프레임을 처리한다고 했을 때 0.05초 동안 얼굴이 움직일 수 있는 범위를 추정하여 임의의 SearchRange 범위를 선정하면 된다. 이 법위 안에 있는 모든 영역들 중 검출된 눈동자와 면적의 차가 가장 작은 영역을 다른 나머지 눈동자라고 판정한다.

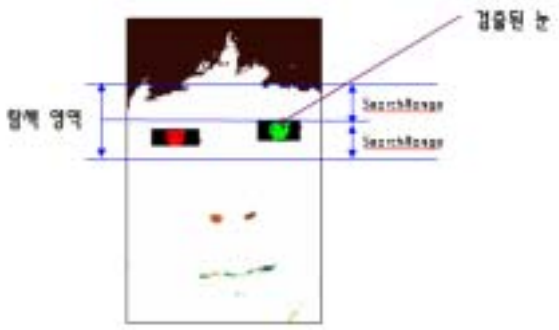


그림 3-19 나머지 한쪽 눈동자의 검색 범위

3.2.5. 마우스 컨트롤

마우스 컨트롤은 두 눈동자 사이의 중심점의 좌표로 마우스를 제어하는 단계이다. 화면 상에 가상의 격자 구간을 설정하고 중심에서 멀어질수록 큰 가중치의 속도 값을 부여해 둔다. 두 눈동자 사이의 중심 좌표가 머무는 격자 구간의 속도로 마우스를 이동하게 한다.

-2,-2	-1,-2	0, -2	1,-2	2, -2
-2,-1	-1,-1	0, -1	1, -1	2, -1
-2, 0	-1, 0	0, 0	1, 0	2, 0
-2, 1	-1, 1	0, 1	1, 1	2, 1
-2, 2	-1, 2	0, 2	1, 2	2, 2

그림 3-20 화면의 가상 격자 설정과 격자 구간의 속도값

4. 실험 및 결과 분석

본 연구에서 제안한 방법을 적용하기 위하여 두 가지의 실험을 하였다. 첫 번째는 PC 카메라로 촬영한 정지 영상에 대해서 눈동자를 검출하는 실험이고 두 번째는 PC 카메라로 Video 스트림을 입력받아 실시간으로 마우스를 이동시키는 실험을 하였다. 첫 번째 실험에서 제안된 눈동자 검출 알고리즘의 타당성을 검증하였고 두 번째 실험에서는 제안된 눈동자 알고리즘을 이용한 마우스 포인터의 이동이 가능함을 테스트 하였다.

본 논문에서 제안한 알고리즘을 Visual C++언어를 사용하여 프로그래밍하고 펜티엄 컴퓨터에서 실행하였다. 샘플 정지 영상은 연구실에 상주하는 4명을 집적 PC 카메라로 촬영한 영상을 사용하였으며 무작위로 다양한 포즈를 취하도록 하였다.

4.1. 실험 방법

4.1.1. 실험 환경

하드웨어	웹 카메라	MCAM 100
	컴퓨터	Intel Pentium 4 3.0GHz 메모리 1GB
소프트웨어	실행 환경	Windows XPservice pack 2
	사용 언어	Visual C++ 2005
실험 환경	조명	일반 실내 조명
	카메라와 사람과의 거리	20cm ~ 50cm
	얼굴 각도(상하)	-20도 ~ +20도
	얼굴 각도(좌우)	-30도 ~ -30도
실험 대상	실험 대상	연구실 4명
	실험 영상 크기	352x288

표 3-5 실험 환경

4.1.2. 정지영상 눈동자 검출 실험 방법

연구실 6명을 대상으로 카메라 앞에 앉아 카메라와 거리가 20cm ~ 50cm 내에서 얼굴 상하 각도 -20도 ~ +20도 사이 그리고 얼굴 좌우 각도 -30도 ~ -30도 사이에서 자유롭게 포즈를 취하게 하고 실험에 사용하는 PC 카메라로 10장씩의 사진을 획득하였다. 획득된 영상을 본 논문에서 제안된 알고리즘이 적용된 프로그램으로 처리하여 결과를 관찰하였다.

4.1.3. 실시간 마우스 이동 실험 방법

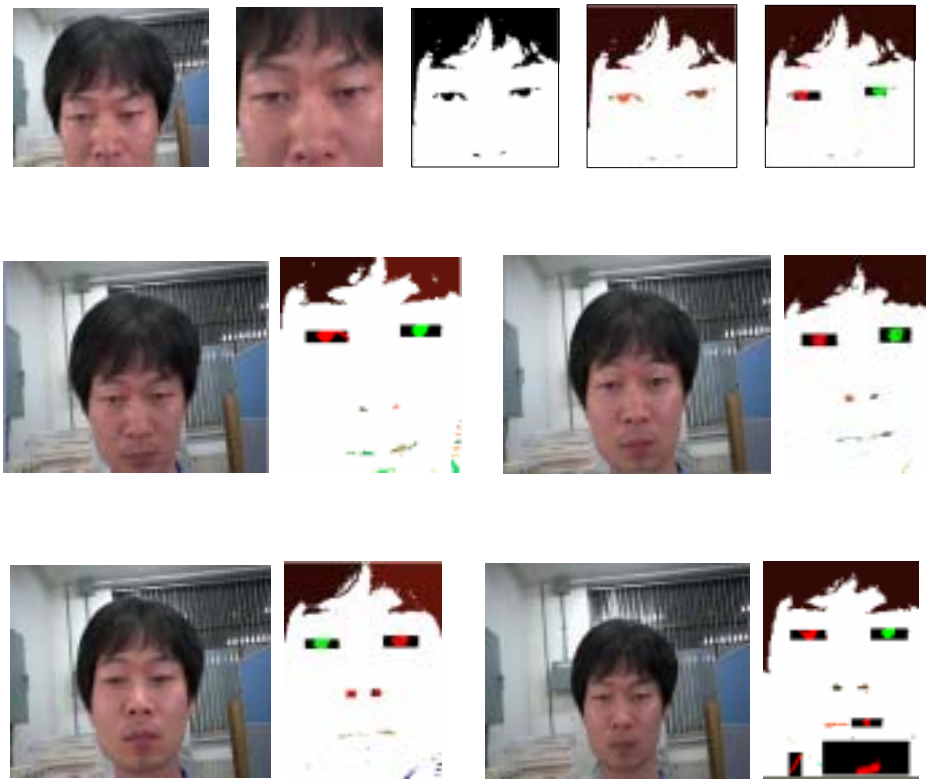
PC 카메라를 모니터 상단에 부착하고 본 논문에 제안된 알고리즘이 적용된 프로그램을 화면에 중앙에 띄워 놓고 화면상에 보이는 얼굴의 위치를 관찰하면서 마우스 포인터를 이동하는 실험을 하였다. 그림 4-1은 실시간 마우스 이동을 위해서 제작된 프로그램의 외형을 캡처한 것이다.

그림 4-1 마우스 실시간 이동을 위한 Eye Mouse System

4.2. 결과 영상

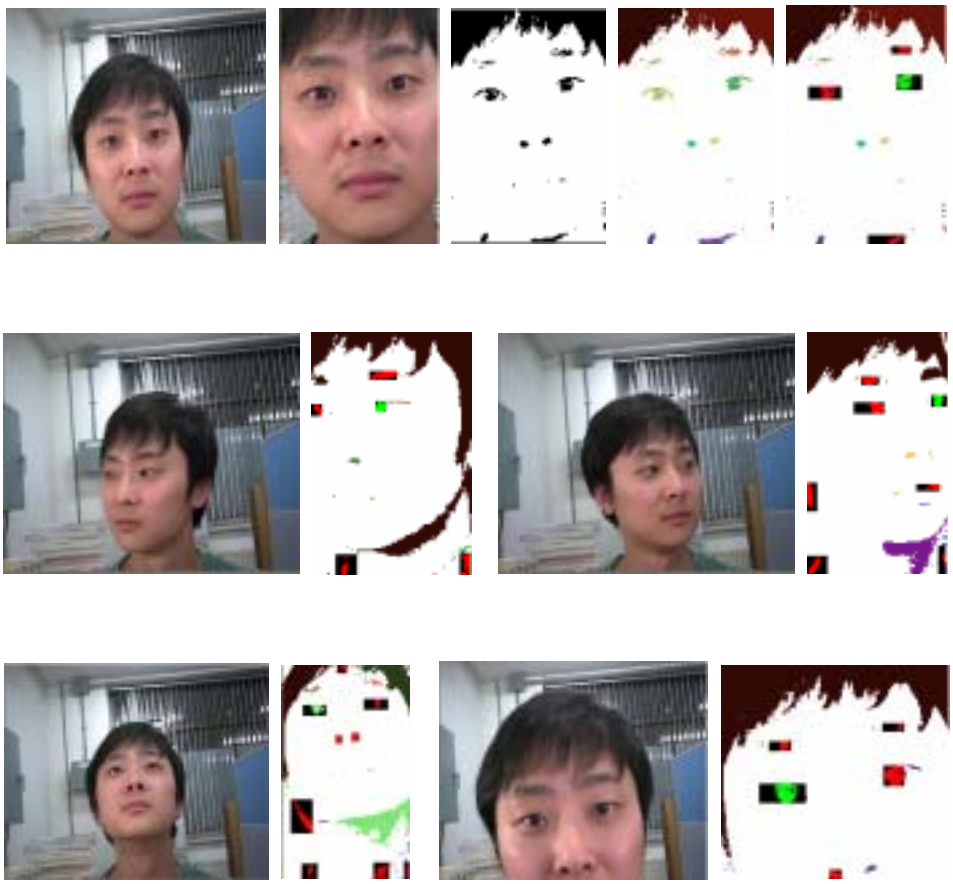
4.2.1. 정지영상 눈동자 검출

1) 피실험자 - A



2) 피실험자 - B

3) 피실험자 - C



4) 피실험자 - D

5. 결 론

눈동자는 기학학적 원에 가장 근접한 신체 기관이므로 눈동자의 이러한 기학적 성질을 이용하는 것이 가장 빠르게 눈동자를 검출할 수 있는 방법이다. 또 투영을 이용하면 2차원 공간상을 1차원으로 변화시킬 수가 있기 때문에 2차원 상의 특정한 물체도 상당히 빠르게 비교할 수가 있다. 본 논문에서는 이 두 가지를 이용하여 빠르게 눈동자를 검출할 수 있는 방법을 제안하였다. 뿐만 아니라 눈꺼풀이나 눈두덩에 의해서 눈동자가 잘려지더라도 여전히 원의 특징을 가지고 있다는 점을 이용하여 눈동자의 실제 수직 중심을 찾는 알고리즘과 비교의 범위를 실제 눈동자가 존재하는 부분에 국한시키는 방법을 제안하여 완전한 모양을 가지지 않는 눈동자에 대해서도 검출 성능을 높이는 방법을 고안하였다. 또 본 논문에서는 이렇게 제안된 눈동자 검출 알고리즘을 가지고 실제 마우스를 제어하는데 적용하였고 마우스를 이동할 때 가상의 격자 구간에 속도 개념을 도입함으로서 마우스의 이동을 효율적으로 제어하는 방법을 제안하였다. 본 논문에서 제시된 알고리즘은 기존에 제시된 방법들 보다 빠른 속도를 보이며 다양한 크기의 눈동자에도 강건하게 눈동자를 검출할 수 있고 눈이 작아서 눈꺼풀과 눈두덩에 의해 눈동자가 잘려지더라도 눈동자를 검출할 수 있다는 특성이 있다.

웹 카메라를 이용한 마우스 포인터의 이동 방법은 웹 카메라의 보급률이 상당히 높은 오늘날 가장 손쉬운 방법이고 별도의 부가장치 없이 소프트웨어만으로 구현할 수 있는 점에서 상당히 매력적인 방법이라 하겠다. 하지만 모든 영상 처리 어플리케이션들이 그렇듯이 여러 가지 엄격한 입력 조건을 유지 시켜 주어야 동작하는 까다로운 면이 있으므로 여러 가지 입력 조건을 적응적으로 선택하게 하는 기법들이 좀 더 많이

연구되어져야할 것이다. 본 논문에서 제안된 방식은 조명의 영향을 많이 받으므로 주변 밝기에 따라 적응적으로 눈동자 이진화 Threshold 값 7ribee을 결정하는 부분에서 더 연구되어져야 할 과제를 남기고 있다.

참 고 문 헌

[1]Shuyan Zhao, Grigat, R.-R, "Robust Eye Detection under Active Infrared Illumination",Pattern Recognition, 2006. ICPR 2006. 18th International Conference on Volume 4, 20-24 Aug. 2006 Page(s):481 - 484.

[2]Kothari, R, Mitchell, J.L. "Detection of eye locations in unconstrained visual images", Image Processing, 1996. Proceedings., International Conference on Volume 3, 16-19 Sept. 1996 Page(s):519 - 522 vol.3

[3]Z.-H. Zhou and X. Geng. "Projection functions for eye detection." Pattern Recognition (PRJ), 2004, 37(5): 1049-1056

[4]Kumar, R.T, Raja, S.K, Ramakrishnan, A.G, "Eye detection using color cues and projection functions", Image Processing. 2002. Proceedings. 2002 International Conference on Volume 3, 24-28 June 2002 Page(s):III-337 - III-340 vol.3

[5]D'Orazio, T, Leo, M, Cicirelli, G, Distante, A, "An algorithm for real time eye detection in face images", Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on Volume 3, 23-26 Aug. 2004 Page(s):278 - 281 Vol.3

[6]Jilin Tu, Huang, T, Hai Tao, "Face as mouse through visual face tracking", Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on 9-11 May 2005 Page(s):339 - 346

[7]Eun Yi Kim, Sin Kuk Kang, Keechul Jung, Hang Joon Kim, "Eye mouse: mouse implementation using eye tracking",

Consumer Electronics, 2005. ICCE. 2005 Digest of Technical Papers. International Conference on 8-12 Jan. 2005 Page(s):207 - 208
[8]Kawato, S, Ohya, J, "Real-time detection of nodding and head-shaking by directly detecting and tracking the between-eyes", Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on 28-30 March 2000 Page(s):40 - 45
[9]Norris, G, Wilson, E, "The Eye Mouse, an eye communication device", Bioengineering Conference, 1997., Proceedings of the IEEE 1997 23rd Northeast 21-22 May 1997 Page(s):66 - 67

눈동자 추적에 의한 마우스 커서 제어

韓 濠 明

嶺南大學校 大學院
컴퓨터工學科 컴퓨터공학專攻

(指導敎授 尹 英 雨)

요 약

본 논문은 일반 웹 카메라를 이용하여 PC 사용자의 얼굴 영상을 실시간으로 획득하여 두 눈동자를 검출하고 그 위치 정보를 이용하여 마우스 포인터를 이동시키는 것을 다루고 있다. 이것은 키보드와 마우스를 빈번히 사용하는 경우에서 오는 불편함을 해소 할 수 있을 뿐만 아니라 거동이 불편한 장애인들이 쉽게 컴퓨터를 이용할 수 있게 한다.

눈동자는 기학학적 원에 가장 근접한 신체 기관이므로 눈동자의 이러한 기학적 성질을 이용하는 것이 가장 빠르게 눈동자를 검출할 수 있는 방법이다. 또 투영을 이용하면 2차원 공간을 1차원으로 변화시킬 수가 있기 때문에 2차원 상의 특정한 물체도 상당히 빠르게 비교할 수가 있다. 본 논문에서는 이 두 가지를 이용하여 빠르게 눈을 검출할 수 있는 방법을 제안하였다. 뿐만 아니라 눈꺼풀이나 눈두덩에 의해서 눈동자가 잘려지더라도 여전히 원의 특징을 가지고 있다는 점을 이용하여 눈동자의 실제 수직 중심을 찾는 알고리즘과 비교의 범위를 실제 눈동자가 존재하는 부분에 국한시키는 방법을 제안하여 완전한 모양을 가지지 않는 눈동자에 대해서도 검출 성능을 높이는 방법을 고안하였다. 또 본 논문

에서는 이렇게 제안된 눈 검출 알고리즘을 가지고 실제 마우스를 제어하는데 적용하였고 마우스를 이동할 때 가상의 격자 구간에 속도 개념을 도입함으로서 마우스의 이동을 효율적으로 제어하는 방법을 제안하였다.

M.S. Thesis

Mouse cursor control by tracking pupils

Chan-Myung Han

Department of Computer Engineering
Graduate school
Yeungnam University

(Directed by Prof. Young-Woo Yoon)

Abstract

This paper deals with moving mouse cursor by detecting and locating pupils from real-time human's face images obtained with a common web camera. It makes work much easier and faster when people use a keyboard and a mouse at the same time and it enables the disable to control a mouse only with their head movement.

Pupils are body parts the most similar to a geological circle. It is the fastest way to detect pupils by taking advantage of this trait. Futhermore, projection functions make it possible to match two-dimensional objects fast because they reduce dimensions of objects from two to one dimension. This paper proposes a fast pupil detecting method with regard to two facts mentioned above. It points out that pupils still have the characteristics of circle when they are cut by the

upper or lower eyelid and enhances the performance of detecting pupils which don't have perfect circles with algorithm of finding real centers of pupils and limiting the matching areas to where pixels of the pupil really exist. It applies the proposed pupil detecting algorithm to moving mouse cursor with virtual lattice blocks of mouse speed.