적외선 조명 카메라를 이용한 시선 위치 추적 시스템

정회원 박 강 령*

Gaze Detection System by IR-LED based Camera

Kang Ryoung Park* *Regular Member*

요 약

사용자의 시선 위치를 파악하는 연구는 많은 응용분야를 가지고 지난 몇 년간 눈부시게 발전되어 왔다. 기존의 대부분 연구에서는 영상 처리 방법만에 의존하여 시선 위치 추적 연구를 수행하였기 때문에 처리 속도도 늦고 많은 사용 제약을 가지는 문제점이 있었다. 이 논문에서는 적외선 조명이 부착된 단일 카메라를 이용한 컴퓨터 비전 시스템으로 시선 위치 추적 연구를 수행하였다. 사용자의 시선 위치를 파악하기 위해서는 얼굴 특징점의 위치를 추적해야하는데, 이를 위하여 이 논문에서는 적외선 기반 카메라와 SVM(Support Vector Machine) 알고리즘을 사용하였다. 사용자가 모니터상의 임의의 지점을 쳐다볼 때 얼굴 특징점의 3차원 위치는 3차원 움직임량 추정(3D motion estimation) 및 아핀 변환(affine transformation)에 의해 계산되어 질 수 있다. 얼굴 특징점의 변화된 3차원 위치가 계산되면, 이로부터 3개 이상의 얼굴 특징점으로부터 생성되는 얼굴 평면 및 얼굴 평면의 법선 벡터가 구해지게 되며, 이러한 법선 벡터가 모니터 스크린과 만나는 위치가 사용자의 시선위치가 된다. 또한, 이 논문에서는 보다 정확한 시선 위치를 파악하기 위하여 사용자의 눈동자 움직임을 추적하였으며 이를 위하여 신경망(다층 퍼셉트론)을 사용하였다. 실험 결과, 얼굴 및 눈동자 움직임에 의한 모니터상의 시선 위치 정확도는 약 4.2cm의 최소 자승 에러성능을 나타냈다.

Key Words : EDFA; WDM; channel add/drop; gain-clamping; disturbance observer technique

ABSTRACT

tThe researches about gaze detection have been much developed with many applications. Most previous researches only rely on image processing algorithm, so they take much processing time and have many constraints. In our work, we implement it with a computer vision system setting a IR-LED based single camera. To detect the gaze position, we locate facial features, which is effectively performed with IR-LED based camera and SVM(Support Vector Machine). When a user gazes at a position of monitor, we can compute the 3D positions of those features based on 3D rotation and translation estimation and affine transform. Finally, the gaze position by the facial movements is computed from the normal vector of the plane determined by those computed 3D positions of features. In addition, we use a trained neural network to detect the gaze position by eye's movement. As experimental results, we can obtain the facial and eye gaze position on a monitor and the gaze position accuracy between the computed positions and the real ones is about 4.2 cm of RMS error.

Ⅰ. 본 론

사용자의 시선 위치를 파악하는 연구는 많은 응용분야를 가지고 지난 몇 년간 눈부시게 발전되어 왔다. 기존의 시선 위치 추적 연구들을 분류해 보면 2차원 및 3차원 얼굴 움직임량을 추정하는 연구[1][15][20][21], 얼굴의 움직임만에 의한 시선 위치를 파악하는 연구[2-8,16,17,19], 눈동자만의 움직임에 의

* 상명대학교 소프트웨어대학 미디어학부(parkgr@smu.ac.kr)
논문번호 : 030088-0306, 접수일자 : 2003년 3월 6일

한 사용자의 시선 위치를 파악하는 연구가 주종을 이루었다[9-14,18]. 반면, 얼굴 및 눈동자 움직임을 함께 고려하여 시선 위치를 파악하는 연구는 거의 이루어지지 않았다. Ohmura와 Ballard[4,5]등의 연구에서는 초기에 얼굴 특징점의 3차원 거리(depth) 정보를 알고 있어야 하며, 시선 위치를 파악하기 위해서는 많은 처리 시간(약 1분 이상)이 소요되는 단점이 있다. Gee[6]와 Heinzmann[7]들의 연구에서는 얼굴 좌표계에서의 시선 벡터의 방향을 계산하였을 뿐, 이로부터 모니터 상에 사용자 시선 위치 등을 구하지 않았다. 또한, 이들의 연구에서는 얼굴의 3차원 회전 및 이동이 동시에 발생하지 않는다고 가정했다. 이는 얼굴의 회전과 이동이 동시에 발생했을 때, 그들의 논문에서 사용하는 최소 자승 정합 알고리즘(least-square fitting algorithm)에서의 연산 복잡성과 처리 시간의 상승 등으로 3차원 움직임량을 정확하게 추정하기 어려웠기 때문이다. Rikert[8]의 연구에서는 학습 및 테스트 환경에서 얼굴 및 모니터 스크린사이의 거리가 변하지 않아야 한다는 가정이 있으며, 이러한 것은 실제 사용에 있어서 많은 불편함을 제공하게 된다. 기타 다른 연구들[10, 13, 14, 16, 17]에서는 사용자로 하여금 구분점이 부착된 안경을 착용하게 함으로써 얼굴 특징점을 추적하는 연구를 수행하였는데, 이처럼 별도의 안경을 착용해야 하므로 사용자에게 불편함을 제공하는 결과를 낳게 되었다. 얼굴 움직임에 의해 모니터상의 시선 위치를 파악했던 연구[2, 3]에서는 눈동자의 움직임은 전혀 고려치 않고 단지 얼굴의 움직임만에 의한 시선 위치 파악 연구를 수행하였다. 이러한 기존의 대부분 연구에서는 영상 처리 방법만에 의존하여 시선 위치 추적 연구를 수행하였기 때문에, 처리 속도도 늦고 많은 사용 제약을 가지는 문제점이 있었다. 그러므로 이 논문에서는 기존 연구에서의 문제점들을 해결하기 위하여 적외선 기반 시선 위치 추적 카메라 시스템을 개발하였으며, 이를 이용하여 얼굴 및 눈동자 움직임에 의한 시선 위치를 파악할 수 있는 새로운 방법을 제안하고자 한다.

Ⅱ. 얼굴 특징점의 추적

일반적으로 모니터 상에 사용자가 응시하고 있는 위치를 파악하기 위해서는 입력 영상에서 얼굴 특징점의 위치를 정확하게 추출할 수 있는 기술이 필수적으로 요구된다. 이 논문에서는 얼굴 특징점으로 양 눈의 중심 및 가장자리, 콧구멍 및 입의 가장 자리 위치 등을 사용하였다. 이러한 특징들을 사용한 이유는 다른 특징들에 비해 추출하기 쉽고, 본 논문에서 개발한 시선 위치 추적 시스템에 의해 보다 잘 검출될 수 있기 때문이다. 이 논문에서는 입력 영상으로부터 눈 영역을 먼저 추출한 후, 이를 기준으로 기타 다른 얼굴 특징점들의 위치를 실시간으로 추적하는 방법을 사용하였다. 일반적인 사무실 환경에서는 사용자의 얼굴 뒤에 복잡한 배경이 존재하며, 또한 다양한 외부 광 등의 영향으로 얼굴 및 얼굴 특징점의 위치를 영상 처리 방법만으로는 실시간으로 추적할 수 있는 일반화된 알고리즘은 현재까지 존재하지 않는 것으로 알려져 있다. 게다가 현재 얼굴 특징 추출 및 얼굴 인식에 있어서 세계 1위의 성능을 나타내고 있는 Identix사의 FaceIt™ 프로그램 역시, 인종 및 조명별로 얼굴 특징 추출 성능이 영향을 많이 받는 것으로 조사되고 있다[23]. 이와 같이 영상 처리 방법만에 의해 얼굴 특징 추출의 한계를 극복하기 위하여 본 논문에서는 그림 1과 같이 적외선 조명 기반 카메라 시스템을 개발하고, 이를 이용하여 입력 영상으로부터 얼굴 특징점을 실시간으로 추출하였다. 먼저, 입력 영상으로부터 눈 위치를 추출하기 위하여 본 논문에서는 적외선 조명을 켰을 때 눈의 각막(Corneal)에서 발생하는 반사위치(Specular Reflection)를 찾는 방법을 사용하였다.

그림 1을 보다 자세히 설명하면 다음과 같다. IR-LED(InfraRed-Light Emitting Diode) 조명(1)은 그림 3에 나타나 있는 것과 같이 눈에서의 반사 위치(Specular Reflection)를 생성하는 데 사용된다. 이때 LED조명으로는 사람 눈으로 감지하기 어려운 880nm의 조명을 사용함으로써, 동작중 사용자의 눈부심 현상이 없도록 하였다. 일반적으로 880nm이상의 적외선 조명에 대해서는 사람들이 감지하지 못하는 것으로 알려져 있다. 카메라 렌즈 앞에 부착된 HPF(High Pass Filter)는 적외선 영역(800nm)이상의 적외선 조명만을 통과시키기 때문에, 카메라 센서를 통한 영상 취득 시 800nm이하의 외부 광(자외선 및 가시광선)이 영향을 주지 못하게 된다. 일반적으로 태양광 및 백열등은 거의 모든 파장대의 광이 골고루 다 포함되어 있지만, 형광등의 경우는 가시 광선대에 비해 적외선 파장대의 광이 상대적으로 적은 것으로 알려져 있다. 즉, 입력된 영상 내에는 카메라에 설치되어 있는 적외선 조명만이 주로 영향을 주기 때문에, 외부 광에 의한 발생하는 영상내의 그림자 등의 영향은 적어지게 된다.

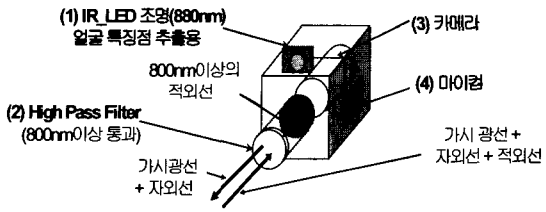


그림 1. 시선 위치 추적용 카메라
Fig. 1. The Gaze Detecting Camera

또한, 본 연구에서 개발한 시선 위치 추적 카메라에서는 일반적인 인터레이시브 CCD(Charge Coupled Device) 센서와 카메라의 동작을 제어하기 위한 마이크로 컨트롤러를 사용했다. 이를 이용하여 그림 2에서 나타나 있듯이 CCD 센서의 출력 신호 중 Even 과 Odd 필드의 VD(Vertical Drive)신호에 맞추어 조명의 On/Off를 조정하게 하였다. 일반적으로 카메라의 CCD 센서 방식으로는 인터레이시브(Interlasive) 방식과 프로그래시브(Progressive) 방식으로 구분할 수 있다. 인터레이시브 방식은 그림 2에서처럼 1/60초 단위로 취득된 Even 필드와 Odd 필드가 합쳐져서 하나의 frame을 형성하는 방식이고, 반면 프로그레시브 방식은 필드의 구분 없이 한 frame을 1/30 초 단위로 취득하는 방식이다.

그림 2를 보다 자세히 설명하면 다음과 같다. 초기에 사용자가 시선 위치 시스템을 시작하려는 순간, PC쪽에 구현되어 있는 프로그램으로부터 카메라 마이컴으로 RS-232C 인터페이스를 거쳐 시작 신호가 전달된다(①). 이와 같은 시작 신호를 받게 되면, 카메라 마이컴은 이 다음 frame부터 그림 2에 나타나 있듯이 CCD의 매 Even 및 Odd Field의 시작 위치 VD신호에 맞추어 적외선 조명을 계속 On/Off 시키게 된다. 이처럼 적외선 조명을 On/Off 시킨 필드 영상(각각 640×240 픽셀 크기)이 입력되면, 이로부터 눈에 발생하는 적외선 반사 위치 및 이를 이용하여 얼굴 특징점의 위치를 추출하게 된다(②). 그림 3은 그림 2의 frame 1에서 취득된 2개의 필드 영상(640×240 픽셀 크기)을 각각 수직 방향으로 2배 확대(interpolation)하여 640×480 픽셀 크기로 만든 것이다. 이때 눈에 발생하는 반사위치(Specular Reflection)의 영상 그레이 값(Gray Value)은 다른 반사 위치(뺨, 혹은 이마 등에 생기는 조명 반사)보다 상당히 높기 때문에, 그림 3의 (a)와 (b)로부터 그림 4와 같이 차영상을 구하게 되면 눈에 발생하는 반사 위치를 쉽게 찾을 수 있게 된다. 그러나 간혹, 눈에 의해 반사되는 영상의 그레이 값이 주변의 다른 반사 위치의 값과 비슷하게 나와서, 눈의 위치 추출에 있어서 어려움을 나타내는 경우가 있다. 이러한 문제점을 해결하기 위하여 본 논문에서는 적목 현상(Red Eye Effect)을 이용하였다. 적목 현상이란 일반적으로 카메라를 이용하여 인물 사진을 촬영할 때 많이 나타나는 현상인데, 카메라의 광축(Optical Axis)과 조명사이의 각도가 5도 이내 인 경우, 조명이 동공(pupil)을 통과하여 망막(retina)에서 밝게 반사됨으로써 사용자 눈이 하얗게 보이는 현상을 나타낸다. 입력 영상으로부터 이와 같은 적목 현상을 파악하기 위해서는 조명의 강도가 상당히 강해야 한다. 이러한 정보를 이용하여, 본 논문에서는 그림 3의 (a)및 (b)에 대한 차영상에서 그레이 값이 일정 임계치 이상 되는 영역(눈 후보)이 일정 비율 이상 발생하게 되면(눈 이외의 반사 영역들이 많이 존재함), 카메라에 부착된 적외선 조명의 강도를 높여서 적목 현상이 발생하도록 함으로써 눈 추출을 돕는 방식을 사용하였다.

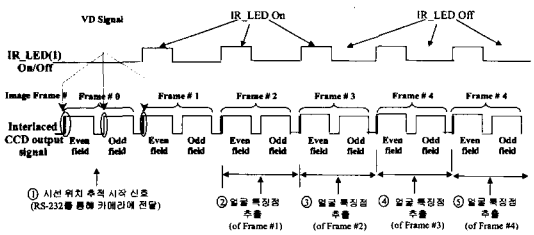
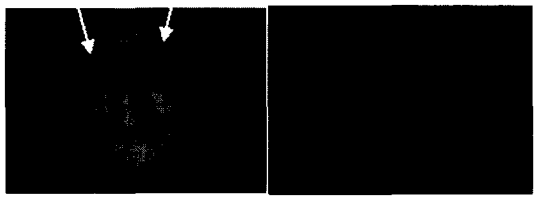


그림 2. 얼굴 특징점 추출을 위한 적외선 조명 조정
Fig. 2. The IR_LED controls for detecting facial features

그림 1의 적외선 조명(1)에 의해 발생하는 눈의 반사 위치(Specular Reflection)



(a) Even field 영상 (b) Odd field 영상

그림 3. Frame# 1에서의 Even 및 Odd Field
Fig. 3. The even and odd images of rame # 1

차영상에서 존재하는 눈의 반사 위치(Specular Reflection)

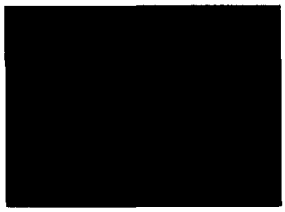
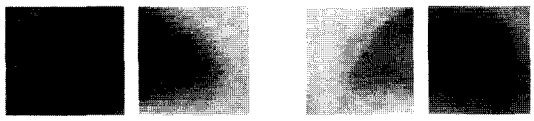


그림 4. Frame #1의 Even 및 Odd 필드사이의 차영상
Fig. 4. The difference image between the even & odd images of Frame # 1

입력 영상으로부터 눈의 반사위치를 추출하게 되면 이로부터 제한된 영역에서 정확한 눈 위치(눈의 중심 및 구석)를 추출하게 된다. 눈의 중심 위치를 추출하기 위해 이 논문에서는 원형 경계 검출 알고리즘(Circular Edge Detection)을 사용하였다. 원형 경계 검출 알고리즘은 가변 템플릿 정합의 일종으로 원형 템플릿의 중심 위치와 반경 등을 제한된 범위 내에서 변화시켜 가면서 가장 잘 정합되는 위치를 찾는 방식이다. 일반적으로 가장 잘 정합되는 위치를 찾기 위해서는 전체 탐색 방식(full search method), 그라디언트 기반 최적 탐색(gradient based search method), 그리고 피라미드 탐색 방식(pyramid search method)등이 존재한다[25]. 탐색 시 참조하는 비용 함수(cost function)의 형태를 고려했을 때, 지역적인 최소값 지점(local minimum)이 존재하지 않는 경우, 전체 탐색보다는 그라디언트 기반 탐색이나 피라미드 탐색 방식이 보다 유용한 것으로 알려져 있다. 본 논문에서는 이러한 비용 함수의 형태를 조사해 본 결과 지역적인 최소값이 존재하지 않는다는 사실을 확인하였으며, 이러한 이유로 그라디언트 기반 탐색과 피라미드 기반 탐색의 성능을 비교하여 보다 우수한 탐색 방식을 선택하여 사용하였다. 실험 결과 그라디언트 기반 탐색은 그라디언트 탐색 스텝의 크기에 따라 성능이 많이 좌우되는 결과를 나타냈기 때문에 본 논문에서는 보다 안정적이고 우수한 성능을 나타내는 피라미드 탐색 방식을 선택하여 눈동자 위치를 찾았다. 이처럼 제한된 영역에서 피라미드 탐색에 의해 눈 영역을 추출하기 때문에, 정확한 눈 중심 위치를 추출하는 데에는 처리 시간이 거의 소요되지 않았다 (Pentium-II 550MHz PC에서 약 3~5 ms). 그림 5는 피라미드 탐색 기반 원형 경계 검출 알고리즘에 의해 찾은 눈의 경계 및 중심 위치를 나타낸 것이다.

그림 5. 피라미드 탐색 기반 원형 경계 검출 알고리즘에 의해 찾은 눈의 경계 및 중심 위치
Fig. 5. The detected eye circles and eye centers by circular edge detection with pyramid search

눈의 중심 위치를 추출한 후, 본 논문에서는 그림 6과 같이 눈구석 모양 템플릿(Eye Corner Shape Template)을 이용하여 눈의 양 구석 위치를 추출하였다.



(a) 왼쪽 눈의 구석 모양 템플릿 (b) 오른쪽 눈의 구석 모양 템플릿
그림 6. 눈의 구석 모양 템플릿
Fig. 6. Eye corner template

본 연구에서는 그림 1에서와 같이 HPF(2)를 사용하므로 입력 영상 내에 외부 광에 의한 영향을 최소로 할 수 있었으며, 그 결과 입력 영상이나 템플릿 등을 조명에 대해 정규화 할 필요가 없었다. 이와 같은 눈구석 템플릿과 함께 본 논문에서는 SVM(Support Vector Machine)을 이용하여 정확한 눈의 구석 위치를 추출할 수 있었다. SVM은 일반적인 패턴 인식 분야에서 많이 사용하고 있는 방법으로 Support Vector(SV)라고 하는 학습 데이터의 집합에 의해 결정되어지는 결정 평면(Decision Surface)을 찾음으로써 두개의 클래스를 구분하는 방법이다. 일반적으로 패턴 분류를 위해 많이 사용하는 다층 퍼셉트론(Multi-Layered Perceptron)의 경우, 신경망의 학습을 위해 사용하는 입력 데이터에 노이즈들이 많이 포함되어 있고 학습을 위한 positive 및 negative 데이터의 양이 충분치 않을 경우 정확한 분류 성능을 나타내지 못하는 경향이 있다. 또한 다층 퍼셉트론의 학습 성능은 많은 초기 파라미터들의 정확한 설정에 의존하게 되는데 이들은 주로 사용자의 경험(heuristic experience)에 의존

하는 경향이 있다.

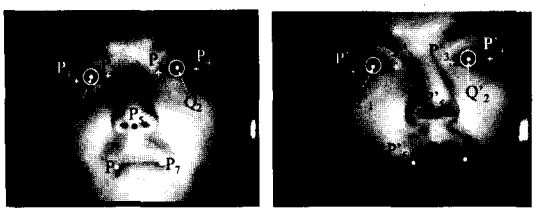
입력된 눈구석 영상은 30×30 픽셀 크기로 정규화하여 SVM의 입력으로 사용한다. 이 논문에서처럼 입력 데이터의 차원(dimension)이 큰 경우, SVM의 패턴 분류가 비선형(nonlinear) 분리 문제가 되는 경우가 많으므로, 본 논문에서는 5차원의 다항 커널(polynomial kernel)을 SVM에 사용하였다. SVM의 분류는 2 클래스로 정의하였으며, 첫 번째 클래스는 정확한 눈의 구석 영역을 두 번째 클래스는 눈의 구석이 아닌 영역으로 정의하였다. SVM의 내적 함수(inner product function)로는 RBF(Radial Base Function), MLP(Multi-Layered Perceptron), Splines, B-Splines등을 사용할 수 있으며, 어느 것을 사용해도 Support Vector를 생성하는데는 큰 영향을 미치지 않는 것으로 알려져 있다. 본 논문에서도 MLP와 다항 커널을 각각 사용했을 때의 성능을 비교한 결과, 거의 유사한 성능을 나타냄을 알 수 있었다. 그러나 C 변수는 SVM의 일반화 성능에 영향을 미치며, 본 논문에서는 반복적인 실험 결과 가장 우수한 성능을 나타내는 C 변수(10000)를 선택하여 사용하였다. 입력 데이터로는 2000장의 연속된 영상을 취득하여(다양한 자세로 앉아 있는 20명의 사람×100장의 영상) 이로부터 8000개의 눈 모서리 샘플(4개의 눈 모서리 샘플/명×2000장의 영상)을 취득했으며, 이외에 1000장의 영상은 추가로 획득함으로써 SVM의 테스트를 위해 사용하였다. 실험 결과, 약 798개의 positive support vector와 4313개의 negative support vector가 선택되어 졌다. 일반적으로 support vector는 학습 과정 중에 분류되기 어려운 데이터를 의미한다. 이로부터 본 논문에서 입력을 위해 사용한 데이터에는 많은 노이즈들이 포함되어 있으며, 이로 말미암아 데이터 분류에 많은 어려움이 있음을 알 수 있다. 실험 결과, 학습 데이터에 대한 분류 성능은 약 0.11%의 분류 에러(9/8000개)를 나타냈으며, 테스트 데이터 중에서는 약 0.2%(8/4000개)의 에러를 나타냈다.

또한 본 논문에서는 동일 입력 데이터들에 대해, MLP를 사용하였을 경우의 분류 성능도 비교를 하였다. 실험 결과, MLP를 사용했을 때에 학습 데이터에 대해서는 약 1.58%의 에러, 그리고 테스트 데이터에 대해서는 약 3.1%의 에러가 나타남을 알 수 있었으므로, 전체적으로 본 연구 분야에서는 SVM을 사용했을 때에 보다 우수한 분류 성능을 나타냄을 알 수 있다. 또한 SVM을 사용했을 때의 분류 시간 역시 Pentium-II 550MHz 환경에서 13ms정도 소요됨을 알 수 있었다. 그러므로 실시간으로 얼굴 특징점 및 시선 위치를 파악하고자 하는 본 연구 분야에서는 충분히 사용 가능한 알고리즘임을 알 수 있었다. 이처럼 입력 영상에서 사용자의 눈 위치를 찾게 되면, 찾은 눈 위치에 대해 코 및 입의 상대적인 존재 가능 영역을 설정할 수 있게 된다. 이 논문에서는 설정된 존재 가능 영역 내에서, SVM을 이용하여 눈의 구석 위치를 찾았던 것과 유사한 방법으로 양 콧구멍 및 입의 끝점 위치를 찾았다. 그림 7은 입력 영상에서 얼굴 특징점의 위치를 추출한 결과를 나타낸 것이다.

그림 7. 추출된 얼굴 특징점의 위치
Fig. 7. The detected facial feature points

얼굴 특징점 추출에 대한 실험 결과, 평균 1 픽셀(양 눈의 중심), 2 픽셀(양 눈의 구석 위치), 4 픽셀(양 콧구멍) 그리고 3 픽셀(입의 양 끝점)의 최소 자승 에러가 발생했음을 알 수 있었다. 본 논문에서의 입력 영상 크기는 640×480픽셀이며, 여기서 최소 자승 에러란 SVM으로 찾은 특징점의 위치와 사용자가 직접 눈으로 보고 찾은 위치 사이의 에러 값을 반영한다. 본 논문에서는 총 3000장의 연속 영상(2000장의 학습 영상 및 1000장의 테스트 영상)을 이용하여 특징점 추출 성능을 테스트했다.

입력 영상에서 얼굴 특징점의 위치가 추출되면, 그림 8에 나타난 것과 같이 총 9개의 특징점(P_1, P_2, P_3, P_4, P_5, P_6, P_7, Q_1, Q_2,)을 선정하게 되는데 이러한 특징점들은 얼굴의 3차원 움직임 량(3차원 회전량 및 이동량)을 추정하기 위해 사용된다. 사용자가 모니터상의 한 지점을 응시하게 되면 그림 8의 (b)에서 나타나 있는 것처럼 추출된 앞서 언급한 9개의 특징점 위치가 (P'_1, P'_2, ~ P'_7 Q'_1, Q'_2)로써 변하게 된다. 여기서 (Q_1, Q_2, P_1, P_2, P_3, P_4,) 과 (Q'_1, Q'_2, P'_1, P'_2, P'_3, P'_4)는 5장에서 언급할 눈동자 움직임에 의한 시선 위치 추출을 위하여 사용된다.



(a) 사용자가 모니터 중앙을 응시할 때 (b) 사용자가 모니터 상의 우측상단 지점을 응시할 때

그림 8. 얼굴 및 눈동자 움직임에 의한 시선 위치 추출을 위해 사용되는 얼굴 특징점

Fig. 8. The feature points for estimating 3D facial and eye ovements

Ⅲ. 초기 얼굴 특징점의 3차원 위치 추정

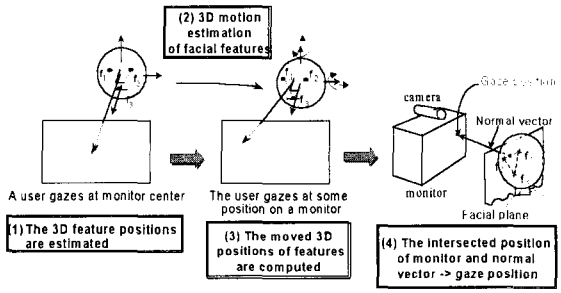


그림 9. 모니터 상의 시선 위치를 파악하기 위한 4단계

Fig. 9. 4 steps in order to compute a gaze position on a monitor

그림 9에 나타나 있는 것처럼, 얼굴 특징점을 추출한 후 모니터상의 시선 위치를 파악하기 위해서는 본 논문에서는 4단계의 과정을 거치게 된다. 첫 번째 단계에서는 사용자가 모니터 상에 미리 지정된 3 위치를 쳐다봄으로써 얻어진 정보들을 바탕으로 초기 얼굴 특징점의 3차원 위치를 자동으로 구하게 된다. 두 번째 및 세 번째 단계에서 사용자가 모니터상의 한 지점을 쳐다보기 위해 얼굴을 움직이게 되는 경우(회전 및 이동), 3차원 움직임 추정(3D Motion Estimation) 및 아핀 변환(Affine Transformation)에 의해 변화된 얼굴 특징점의 위치를 구할 수 있게 된다. 마지막 단계에서는, 변화된 얼굴 특징점들의 위치로부터 하나의 얼굴 평면과 법선을 구할 수 있으며, 이 법선과 모니터 평면이 만나는 위치가 얼굴의 움직임에 의한 시선 위치가 된다. 추가로 이 논문에서는 사용자 눈동자 움직임까지 고려하여 보다 정확한 모니터 상의 시선 위치를 파악했으며, 이에 대한 자세한 내용은 5장에서 거론된다. 그림 9의 첫 번째 단계에 대한 자세한 방법은 [2]에서 참조할 수 있다. 첫 번째 단계에서 구한 초기 얼굴 특징점의 3차원 위치와 이 얼굴 특징점의 실제 위치(3D position tracker sensor로 측정[22]) 사이의 최소 자승 에러(RMS error)는 약 1.15 cm(X축으로 0.64 cm, Y축으로 0.5cm, Z축으로 0.81cm)의 정확도를 나타냈으며, 실험을 위해서는 특징점 추출을 위해 사용했던 20명분의 데이터를 이용하였다.

Ⅳ. 얼굴의 3차원 움직임 및 이동량 추정

이 장에서 설명할 내용은 그림 9의 2번째 단계에 속한다. 기존에 많은 3차원 움직임 추정 방법들이 연구되어 왔는데, 예를 들어 확장 칼만 필터(Extended Kalman Filter)[1], 신경망[21] 그리고 아핀 투영 방법(affine projection method)[6][7] 등이 있다. Fukuhara[21]의 논문에서는 얼굴의 3차원 움직임량을 추정하기 위해 신경망의 다층 퍼셉트론(Multi-Layered Perceptron)을 사용했다. 그러나 이 논문에서는 신경망의 제한된 문제 해결 능력 때문에, 작은 범위에서의 3차원 회전 및 이동량 추정을 하였다. 아핀 투영 방법에 의해 3차원 움직임량을 추정하는 연구[6][7]에서는 카메라와 사용자 얼굴 사이의 Z 거리의 변화도가 초기 Z 거리에 비해 10% 이상 변하지 않는다는 가정 하에 사용하였다. 또한, 얼굴 특징점의 3차원 회전 및 이동이 동시에 발생하는 경우 이 논문에서 사용하는 최소 자승 정합 알고리듬에서의 복잡성 증가로 인하여 정확한 움직임량 추정이 어렵다는 문제점도 있다. 또한 확장 칼만 필터는 기존의 3차원 움직임 추정을 위해 가장 많이 사용하고 있는 방식이지만, 사용자의 얼굴 움직임에 있어서 급격한 방향 및 움직임 변화가 발생하여 확장 칼만 필터에서 사용하는 등가속도 모델을 벗어나는 경우 제대로 추정하지 못하는 단점이 있다. 이와 같이 문제점들을 고려하여, 본 논문에서는 확장 칼만 필터의 문제점을 보완한 반복적 확장 칼만 필터(Iterative Extended Kalman Filter)를 사용하였다. 반복적 확장 칼만 필터는 많은 부분이 확장 칼만 필터와 유사하므로, 본 논문에서는 먼저 확장 칼만 필터에 대하여 설명하고자 한다. 일반적인 확장 칼만 필터는 식 (1)과 같은 형태를 나타낸다.

얼굴의 3차원 움직임량 추정을 위해 사용되는 확장 칼만 필터는 사용자가 얼굴을 움직이는 동안 추출된 얼굴 특징점의 2차원 위치 정보를 등가속도 모델을 이용함으로써 얼굴의 3차원 회전 및 이동량 정보로 변환하는 데 주로 사용되는 방법이다[1]. 확장 칼만 필터에서 3차원 움직임량을 추정하기 위하여 본 논문에서는 각 특징점당 18×1의 상태 벡터(state vector)를 사용하였다. 여기서 상태 벡터는 x(t)라고 정의한다(x(t) = ($\boldsymbol{p}(t)$, $\boldsymbol{q}(t)$, $\boldsymbol{v}(t)$, $\boldsymbol{w}(t)$, $\boldsymbol{a}(t)$, $\boldsymbol{b}(t)$)^T). 이때 $\boldsymbol{p}(t)$ = (x(t),y(t),z(t)): 모니터 좌표계에 대한 얼굴 좌표계의 3차원 이동량, $\boldsymbol{q}(t)$ = $(\theta_x(t), \theta_y(t), \theta_z(t))$: 얼굴 좌표계에서 특징점의 3차원 회전량, $\boldsymbol{v}(t)$ = $(v_x(t), v_y(t), , v_z(t))$ $\boldsymbol{w}(t)$ = $(w_x(t), w_y(t), w_z(t))$: 얼굴 좌표계에서 특징점의 회전 및 이동량의 속도 성분, $\boldsymbol{a}(t)$ = $(a_x(t), a_y(t), a_z(t))$, $\boldsymbol{b}(t)$ = $(\alpha_x(t), \alpha_y(t), \alpha_z(t))$: 얼굴 좌표계에서 특징점의 회전 및 이동량의 가속도 성분 등등으로 정의한다. 이와 같은 상태 벡터가 주어졌을 때 본 논문에서는 식 (1)과 같은 확장 칼만 필터 식을 이용하여 얼굴의 3차원 움직임량을 추정하였다.

$$\hat{\mathbf{x}}(\mathbf{t}) = \hat{\mathbf{x}}(\mathbf{t})^- + \mathbf{K}(\mathbf{t})(\mathbf{y}(\mathbf{t}) - h(\hat{\mathbf{x}}(\mathbf{t})^-)) \quad (1)$$

여기서 K(t)는 칼만 이득 행렬(Kalman gain matrix)이다.

($\mathbf{K(t)} = \mathbf{P}(t-1)\mathbf{H}(t)^{\mathbf{T}}(\mathbf{H}(t)\mathbf{P}(t-1)\mathbf{H}(t)^{\mathbf{T}}+\mathbf{R}(t))^{-1}$, $\mathbf{H(t)} = \frac{\partial h}{\partial x(t)}\Big|_{\hat{\mathbf{x}}(\mathbf{t})^-}$) 그리고 P(t)는 상태 예측 에러 공분산(state prediction error covariance)이다. 확장 칼만 필터는 식 (1)에서와 같이 비선형 관측 함수(h)를 사용한다는 점에서 이산 칼만 필터(Discrete Kalman Filter)와 구별된다. 확장 칼만 필터는 3차원 움직임량이 등가속도 모델을 따른다는 가정 하에, 얼굴 특징점의 3차원 회전량 및 이동량을 포함하여 이전 시간에 변화된 상태 벡터 ($\hat{\mathbf{x}}(\mathbf{t}-\mathbf{1})$)로부터 현재의 상태 벡터를 예측하는 역할을 수행한다. 이에 대한 자세한 설명은 [1][20]에서 참조할 수 있다. 그러나, 이러한 확장 칼만 필터는 특징점의 움직임에 있어서 급격한 방향 변화가 발생하는 경우, 등가속도 가정을 벗어남으로써 이들의 움직임을 놓치는 결과를 종종 나타내곤 한다. 이러한 문제점을 해결하기 위하여 본 논문에서 사용하는 방식이 반복적 확장 칼만 필터이다. 반복적 확장 칼만 필터는 확장 칼만 필터에서 테일러 시리즈(Taylor Series)를 사용하여 비선형 방정식을 선형 방정식으로 근사화 함으로써 발생하는 에러를 보정하는 기능을 수행한다[24]. 즉, 반복적 확장 칼만 필터에서는 식 (1)에 있는 칼만 이득 행렬(K(t))에 반복적 적응 과정을 통해 에러 공분산(P(t))을 최소화하는 과정을 포함시킴으로써 이러한 에러를 보정하게 된다.

본 논문에서 반복적 확장 칼만 필터의 예측 정확도는 3차원 위치 추적 장비(Polhemus Sensor)와 비교되었다. 실험 결과, 확장 칼만 필터와 3차원 위치 추적 장비와의 정확도 차이는 이동량 및 회전량에서 각각 1.1cm 및 2.43o의 에러 성능을 나타냈다.

V. 모니터 상의 시선 위치 파악

1. 얼굴의 움직임에 의한 시선 위치 파악

이 장에서 설명하는 내용은 그림 9의 3, 4번째 단계에 해당된다. 앞서 3장에서 설명한데로 모니터 좌표계를 기준으로 파악된 초기 얼굴 특징점의 3차원 위치(그림 4의 P1 ~ P7)는 얼굴 좌표계를 기준으로 변환(X_i, Y_i, Z_i)되어 질 수 있다. 이 후, 이 특징점의 위치는 식 (2) 및 그림 10과 같이 사용자가 모니터 상의 한 지점을 쳐다 볼 때 발생하는 회전[R] 및 이동[T](확장 칼만 필터에 의해 추정되는) 행렬에 의해 변화되며, 이 행렬을 이용한 아핀 변환 계산에 의해 변화된 얼굴 특징점의 위치 (X'_i, Y'_i, Z'_i)를 구할 수 있게된다. 이처럼 사용자가 모니터 상의 한 지점을 쳐다볼 때의 얼굴 특징점의 위치를 계산하면, 이를 모니터 좌표계를 기준으로 다시 변환하고, 이 특징점들이 형성하는 얼굴 평면 및 평면의 법선이 모니터와 만나는 위치가 사용자의 시선위치가 된다[2].

$$\begin{bmatrix} X'_i \\ Y'_i \\ Z'_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ T_Z \end{bmatrix} \quad (2)$$

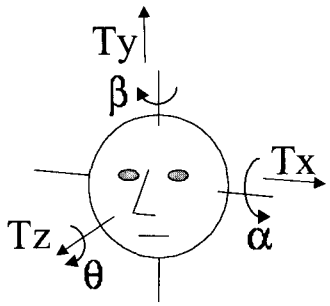


그림 10. 얼굴의 3차원 회전 및 이동
Fig. 10. 3D facial rotation and translation

2 눈동자의 움직임에 의한 시선 위치 파악

앞서 5.1절에서 설명한 사용자의 시선 위치는 눈동자의 움직임은 고려하지 않고 얼굴의 움직임에 의해서만 모니터상의 시선위치를 파악한 것이다. 그러나 대부분의 경우 사용자가 모니터 상의 한 지점을 쳐다볼 때에는 얼굴 및 눈동자의 움직임이 동시에 발생한다. 그러므로 본 논문에서는 보다 정확한 시선 위치를 파악하기 위하여 사용자의 눈동자 움직임을 파악하는 연구를 수행하였다. 사용자의 눈동자 움직임은 그림 12에 나타나 있듯이 입력 영상에서 추출된 특징 정보(Q_1, Q_2, P_1, P_2, P_3, P_4) 및 (Q'_1, Q'_2, P'_1, P'_2, P'_3, P'_4)를 바탕으로 파악된다. 그림 11에서와 같이 일반적으로 사용자의 시선 위치에 따라 눈동자의 움직임 량 및 형태 정보는 변하게 된다. 특히, 눈동자의 중심위치와 왼쪽 혹은 오른쪽 눈의 구석 위치 사이의 거리는 사용자의 시선 위치에 따라 변화하게 된다. 본 논문에서는 그림 12와 같이 눈동자의 움직임량 및 시선 위치 사이의 관계를 학습하기 위하여 신경망(다층 퍼셉트론)을 사용하였다.

(a) 사용자가 모니터의 왼쪽을 쳐다볼 때 (b) 사용자가 모니터의 중앙을 쳐다볼 때 (c) 사용자가 모니터의 우측을 쳐다볼 때

그림 11. 사용자의 시선 위치에 따른 눈동자의 움직임 량 및 형태 변화
Fig. 11. The eye movements and shape change according to user's gaze positions

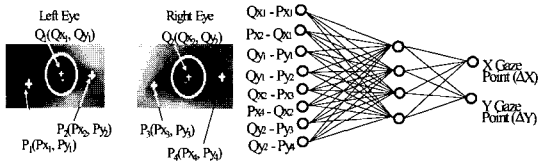


그림 12. 눈동자 움직임에 의한 시선 위치 파악을 위한 특징 및 신경망 구조
Fig. 12. The features and the neural network for detecting gaze position by eye movements

위에서 설명한 대로 눈동자 움직임에 의한 시선 위치를 파악하게 되면, 5.1절 및 5.2절의 결과를 종합하여 최종적으로 얼굴 및 눈동자 움직임에 의한 모니터 상의 시선위치를 그림 13과 같이 파악할 수 있게 된다. 즉, 5.1절에서 설명한 대로 얼굴 움직임에 의한 시선위치를 파악한 후, 이를 기준으로 추가로 눈동자 움직임에 의한 시선 위치 변이량(△X, △Y)을 파악함으로써 최종적인 모니터 상의 시선위치를 계산할 수 있게 되는 것이다.

VI. 실험 결과 및 고찰

본 논문에서 제안하는 시선 위치 파악의 정확도는 표 1, 2와 같이 기존의 연구 결과[2][3][19]와 비교하였다. 여기서[2], [3]의 논문은 3장에서 설명한 4단계를 거쳐 모니터 상의 시선 위치를 파악하는 방법을 취하고 있으나, 사용자의 눈동자 움직임은 고려하지 않고 오직 얼굴 움직임만을 고려하여 모니터 상의 시선 위치를 파악하는 방법을 사용하고 있다. 그리고 [19]의 논문은 [2][3] 논문 및 본 논문에서 사용한 방법과 같이 얼굴 특징점의 초기 3차원 위치 및 변화된 3차원 위치를 구하는 것이 아니라, 2차원 영상에서 취득된 얼굴 특징점의 위치로부터 선형 보간법(Linear Interpolation), 신경망(Neural Network) 등을 이용하여 직접 모니터 상의 시선 위치를 구하는 방법을 사용하고 있다. 또한 이 [19]논문 역시 사용자의 눈동자 움직임은 고려하지 않고, 오직 얼굴 움직임만을 고려하여 모니터 상의 시선 위치를 파악하고 있다.

실험 데이터로는 총 10명의 사용자가 19인치 모니터 상에 골고루 분포된 23개 지점을 쳐다볼 때 취득된 데이터를 사용하였다. 여기서 시선 위치 에러란 사용자가 모니터 상에 실제 쳐다보는 위치와 본 논문에서 제안하는 시선 위치 파악 알고리즘에 의해 계산되어 나온 위치사이의 최소 자승 에러를 나타낸다. 실험은 크게 두 가지로 나누어 사용자가

모니터상의 한 지점을 쳐다볼 때 눈동자의 움직임은 고정으로 하고, 얼굴의 움직임에 의해서만 응시하는 경우(표 1)와 얼굴 및 눈동자 움직임을 같이 허용하여 자연스럽게 쳐다보는 경우(표 2)로 구분하여 수행하였다. 먼저 얼굴 움직임에 의해서만 쳐다보는 경우의 정확도는 표 1에 나타나 있는 것처럼 본 논문에서 제안한 방법이 기존의 많은 방법들에 비해 가장 우수함을 알 수 있다. 그리고 표 2와 같이 얼굴 및 눈동자 움직임이 같이 발생하는 실험 데이터에 대한 시선 위치 추출 정확도 역시 본 논문에서 제안하는 방식이 가장 우수함을 알 수 있다. 전반적으로는 얼굴의 움직임만이 존재하는 경우보다 얼굴 및 눈동자의 자연스러운 움직임이 같이 존재하는 경우의 시선 위치 추출 정확도가 다소 떨어짐을 알 수 있다.

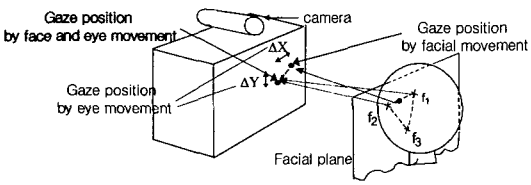


그림 13. 얼굴 및 눈동자 움직임에 의한 모니터상의 시선 위치 파악
Fig. 13. Detecting gaze position on a monitor by face and eye movements

표 1. 얼굴만의 움직임에 의한 시선 위치 추출 정확도
Table 1. Gaze error about test data including only facial movements (cm)

방법	선형 보간법 [19]	단일 신경망 [19]	다중 신경망 [19]	[2] 방법	[3] 방법	본 논문의 방법
error	5.1	4.23	4.48	5.35	5.21	3.40

표 2. 얼굴 및 눈동자 움직임을 같이 고려한 시선 위치 추출 정확도
Table 2. Gaze error about test data including face and eye movements (cm)

방법	선형 보간법 [19]	단일 신경망 [19]	다중 신경망 [19]	[2] 방법	[3] 방법	본 논문의 방법
error	11.8	11.32	8.87	7.45	6.29	4.2

두 번째 실험으로는 1280×1024 픽셀의 화면 해상도를 갖는 19인치 모니터에 수평 및 수직 150 픽셀 간격(2.8cm 간격)으로 반경 5 픽셀의 점들을 표시하고, 사용자가 이를 쳐다볼 때의 시선 위치 추출 정확도를 측정하였다. 이로부터 총 560개의 테스트 샘플(10명분 × 56개의 시선 위치)을 취득하였으며, 이러한 실험 조건은 Rikert의 연구[8]에서와 동일하다. 실험 결과 실제 사용자가 쳐다본 응시 위치와 본 논문에서 제안한 방법에 의해 계산된 응시 위치 사이의 최소 자승 에러는 약 4.33cm인 것으로 나타났으며, 이러한 실험 결과는 Rikert의 방법에서의 정확도(약 5.08 cm의 최소 자승 에러)보다 우수함을 알 수 있다. 또한 Rikert는 사용자 얼굴과 모니터사이의 Z거리는 학습 및 실제 사용 시 고정되었다고 가정하고 있으나, 본 논문에서 그러한 불편한 제약을 사용하고 있지 않다. 이의 검증으로 본 논문에서는 사용자의 Z거리를 변화시켜가면서(55, 60, 65cm) 시선 위치 정확도를 측정하였다. 실험 결과 최종 자승 에러는 다음과 같다. Z거리가 55cm일 때 4.26cm, Z거리가 60cm일 때 4.33cm 그리고 Z거리가 65cm일 때 4.42cm. 전술한 바와 같이 실험 결과, 본 논문에서의 방법은 사용자의 Z거리 변화에 따라 그 정확도에 영향을 받지 않는다는 것을 보여주고 있다. 또한 Rikert의 방법은 처리 시간이 많이 소요되는 문제점(alphastation 333MHz에서 약 1 분)이 있었으나 본 논문의 방법은 이에 비해 상당히 빠르고 실시간으로 구현 가능한 결과를 나타내고 있다(Pentium-II 550MHz에서 평균 730ms).

Ⅶ. 결 론

이 논문에서는 새로운 시선 위치 추적 방법을 제안하고 있다. 실험 결과 시선 위치 추적 에러는 약 4.2cm인 것으로 나타났으며, 실시간으로 사용자의 얼굴 및 눈동자 움직임을 추적하여 모니터 상의 시선 위치를 파악함을 알 수 있었다. 또한, 이와 같은 시선 위치 추적 에러는 얼굴 혹은 눈동자의 추가적인 움직임에 의해 보정될 수 있다. 즉, 본 논문에서 제안한 방법에 의해 표시되는 모니터상의 시선 위치를 추가적인 얼굴 혹은 눈동자 움직임에 의해 움직임으로써(마우스 드레깅과 같은 동작), 4.2cm의 시선 위치 에러를 보정할 수 있는 것이다.

향후, 보다 정확한 시선 위치 추출을 위해서는 눈동자 위치를 보다 정확히 추적할 수 있는 방법이 요구되며, 이를 위해서는 눈 영상만을 보다 고해상

도로 취득할 수 있는 별도의 카메라를 사용해야 할 것으로 예상된다.

참 고 문 헌

[1] A. Azarbayejani., "Visually Controlled Graphics". *IEEE Trans. PAMI*, vol. 15, no. 6, pp. 602-605, 1993.

[2] K. R. Park et al., "Gaze Point Detection by Computing the 3D Positions and 3D Motions of Face", *IEICE Trans. Information and Systems*, vol. E.83-D, no. 4, pp. 884-894, Apr. 2000.

[3] K. R. Park et al., "Gaze Detection by Estimating the Depth and 3D Motions of Facial Features in Monocular Images", *IEICE Trans. Fundamentals*, vol. E.82-A, no. 10, pp. 2274-2284, Oct. 1999.

[4] K. OHMURA et al., "Pointing Operation Using Detection of Face Direction from a Single View". *IEICE Trans. on Information and Systems*, vol. J72-D-II, no. 9, pp. 1441-1447, 1989

[5] P. Ballard et al., "Controlling a Computer via Facial Aspect". *IEEE Trans. on SMC*, vol. 25, no. 4, pp. 669-677, 1995.

[6] A. Gee et al., "Fast Visual Tracking by Temporal Consensus", *Image and Vision Computing*. vol. 14, pp. 105-114, 1996.

[7] J. Heinzmann et al., "3D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm". *in Proc. of ICAFGR*, pp. 142-147, 1998.

[8] T. Rikert et al., "Gaze Estimation using Morphable Models". *in Proc. of ICAFGR*, pp. 436-441, 1998.

[9] A. Ali-A-L et al., "Man-Machine Interface through Eyeball Direction of Gaze". *in Proc. of Southeastern Symposium on System Theory*, pp. 478-82, 1997.

[10] A. TOMONO et al., "Eye Tracking Method Using an Image Pickup Apparatus". *European Patent Specification-94101635*, 1994.

[11] Seika-Tenkai-Tokushuu-Go, *ATR Journal*, 1996.

[12] Eyemark Recorder Model EMR-NC, *Technical Report*, NAC Image Technology Cooperation.

[13] Porrill-J et al., "Robust and Optimal Use of Information in Stereo Vision". *Nature*. vol. 397, no. 6714, pp. 63-6, Jan. 1999.

[14] Varchmin-AC et al., "Image Based Recognition of Gaze Direction Using Adaptive Methods. Gesture and Sign Language in Human-Computer Interaction". *in Proc. of Int. Gesture Workshop*. pp. 245-257, 1998.

[15] J. Heinzmann et al., "Robust Real-Time Face Tracking and Gesture Recognition". *in Proc. of IJCAI*, vol. 2, pp. 1525-1530, 1997.

[16] Matsumoto-Y, et al., "An Algorithm for Real-Time Stereo Vision Implementation of Head Pose and Gaze Direction measurement". *in Proc. of ICAFGR*, pp. 499-504, 2000.

[17] Newman-R et al., Real-Time Stereo Tracking for Head Pose and Gaze Estimation. *in Proc. of ICAFGR*, pp. 122-128, 2000.

[18] Betke-M et al., "Gaze Detection via Self-Organizing Gray-Scale Units". *in Proc. of Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time System*, pp. 70-76, 1999.

[19] Jaihie Kim et al., "Intelligent Process Control via Gaze Detection Technology". *The journal of Engineering Applications of Artificial Intelligence*, vol. 13, no. 5, pp. 577-587, 2000.

[20] T. BROIDA et al., "Recursive 3-D Motion Estimation from a Monocular Image Sequence". *IEEE Trans. Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639-656, 1990.

[21] T. Fukuhara et al., "3D-Motion Estimation of Human Head for Model-Based Image Coding". *IEE Proc.*, vol. 140, no. 1, pp. 26-35, 1993.

[22] http://www.polhemus.com

[23] http://www.identix.com

[24] R. Brown and P. Hwang. *Introduction To Random Signals and Applied Kalman Filtering*, the 3rd Edition, Wiley, 1994.

[25] R. C. Gonzalez, *Digital Image Processing*, Addison-Wesley, 1995.

박 강 령(Kang Ryoung Park) 정회원

1994년 2월 : 연세대학교 전자공학과 학사

1996년 2월 : 연세대학교 전자공학과 석사

2000년 2월 : 연세대학교 전기 · 컴퓨터 공학과 박사

2000년 3월 ~ 2003년 2월 : LG전자 기술원 Digital Vision Group, 선임연구원

2003년 3월 ~ 현재 : 상명대학교 소프트웨어대학 미디어학부 전임강사

<관심분야> 영상처리, 컴퓨터비젼, 생체인식