단안 카메라 환경에서의 시선 위치 추적

정회원 박 강 령*, 김 재 희**

A Gaze Detection Technique Using a Monocular Camera System

Kang-Ryoung Park*, Jaihie Kim** *Regular Members*

요 약

시선 위치 추적이란 사용자가 모니터 상의 어느 지점을 쳐다보고 있는 지를 파악해 내는 기술이다. 시선 위치를 파악하기 위해 본 논문에서는 2차원 카메라 영상으로부터 얼굴 영역 및 얼굴 특징점을 추출한다. 초기에 모니터상의 3 지점을 쳐다볼 때 얼굴 특징점들은 움직임의 변화를 나타내며, 이로부터 카메라 보정 및 매개변수 추정 방법을 이용하여 얼굴특징점의 3차원 위치를 추정한다. 이후 사용자가 모니터 상의 또 다른 지점을 쳐다볼 때 얼굴 특징점의 변화된 3차원 위치는 3차원 움직임추정방법 및 아핀변환을 이용하여 구해낸다. 이로부터 변화된 얼굴 특징점 및 이러한 얼굴 특징점으로 구성된 얼굴평면이 구해지며, 이러한 평면의 법선으로부터 모니터 상의 시선위치를 구할 수 있다. 실험 결과 19인치 모니터를 사용하여 모니터와 사용자까지의 거리를 50~70cm정도 유지하였을 때 약 2.08인치의 시선위치에러 성능을 얻었다. 이 결과는 Rikert의 논문[7]에서 나타낸 시선위치추적 성능(5.08 cm 에러)과 비슷한 결과를 나타낸다. 그러나 Rikert의 방법은 모니터와 사용자 얼굴까지의 거리는 항상 고정시켜야한다는 단점이 있으며, 얼굴의 자연스러운 움직임(회전 및 이동)이 발생하는 경우 시선위치추적 에러가 증가되는 문제점이 있다. 동시에 그들의 방법은 사용자 얼굴의 뒤 배경에 복잡한 물체가 없는 것으로 제한조건을 두고 있으며 처리 시간이 상당히 오래 걸리는 문제점이 있다. 그러나 본 논문에서 제안하는 시선 위치 추적 방법은 배경이 복잡한 사무실 환경에서도 사용가능하며, 약 3초 이내의 처리 시간(200MHz Pentium PC)이 소요됨을 알 수 있었다.

ABSTRACT

Gaze detection is to locate the position on a monitor screen where a user is looking. To detect the gaze position, we locate facial region and facial features in 2D camera images. From the movement of feature points detected in starting images, we can compute the initial 3D positions of those features by camera calibration and parameter estimation algorithm. Then, when a user moves his face in order to gaze at one position on a monitor, the moved 3D positions of those features can be computed from 3D rotation and translation estimation and affine transform. Finally, the gaze position on a monitor is computed from the normal vector of the plane determined by those moved 3D positions of features. As experimental results, we can obtain the gaze position on a monitor(19 inches) and the gaze position accuracy between the computed positions and the real ones is about 2.08 inches of RMS error. This result is similar to that of Rikert's method. However, Rikert's method has the drawback that the distance between the user and the monitor screen must be always the same. Thus, when the user rotates and translates his head, the gaze detection error increases in Rikert's method. In addition, Rikert's method is restricted to the application with simple background and also takes much time to detect the gaze position, whereas our algorithm can be applied to the case with many complicated objects in the background and takes less than 3 seconds to compute the gaze position with a Pentium-Pro 200MHz PC..

Keywords : Gaze Detection, Camera Calibration, Parameter Estimation

* LG전자기술원 Digital Vision Group(parkgr@lgcit.com), ** 연세대학교 전기 · 컴퓨터 공학과(jhkim@bubble.yonsei.ac.kr)
논문번호 : 010065-0416, 접수일자 : 2001년 4월 16일

Ⅰ. 서 론

시선 위치 추적이란 사용자가 모니터 상의 어느 지점을 쳐다보고 있는 지를 파악해 내는 기술이다. 이러한 시선 위치 추적 기술은 많은 응용 분야를 가지고 있는데, 그 대표적인 예로는 손발을 사용하지 못하는 심신 장애자를 위한 컴퓨터 인터페이스, 다중 윈도우 환경에서 마우스 커서의 움직임을 사용자의 시선 위치 추적으로 대용하거나 혹은 공정 제어 환경과 같이 동시에 조정해야할 버튼들이 많은 상황에서 사용자의 양손 이외에 제 3의 입력 수단으로 시선 위치 추적 기술을 이용할 수 있다[1]. 이 논문에서는 카메라 및 영상 입력 장비 외에 특별한 장비 없이 컴퓨터 비젼 방법에 의해 시선 위치 추적 알고리듬을 구현하였다. 기존 대부분의 연구에서는 컴퓨터 비젼 방법에 의해 얼굴의 3차원 움직임량(회전(rotation) 및 이동(translation))만을 파악하는 방법들이 주로 연구되었으며[2-6], 이로부터 모니터 상에 사용자가 쳐다보고 있는 위치를 파악하는 연구는 최근 들어서 조금씩 수행되고 있다[7][8]. Azarbayejani와 Fukuhara등은[2][3] 각각 확장 칼만 필터(Extended Kalman Filter)와 신경망(Neural Network)등을 이용하여 얼굴의 3차원 회전 및 이동량을 추정하는 연구를 수행하였으며, Ballard등은[4] 는 연속적 근사 방법(successive approximation method)을 사용하여 시선 벡터의 회전량을 파악하였다. 또한 Gee등[5]과 Heinzmann등[6]은 아핀 투영 방법(affine projection algorithm)을 사용하여 시선 벡터의 3차원 회전량을 구하였다. 그러나 그들의 연구들은 전술한 바와 같이 사용자 얼굴의 3차원 상대적인 회전량 및 이동량만을 추정하였을 뿐, 모니터 상에 사용자가 쳐다보고 있는 위치는 계산하지 않았다. 이러한 얼굴의 3차원 움직임 추정 결과로부터 모니터 상의 시선 위치를 알기 위해서는 모니터와 사용자 얼굴 사이의 3차원 거리(depth) 파악, 모니터, 카메라 및 얼굴 좌표계 간의 결합 방법 등이 추가로 연구되어야 하기 때문이다. 이에 반하여 Rikert등[7]은 신경망에 의해 학습된 변환 2차원 얼굴 모델(morphable face model)을 이용하여 모니터 상의 시선 위치를 파악하였다. 그러나 그들의 방법은 얼굴의 3차원 위치 및 움직임(회전 및 이동)을 파악하지 않는 상태에서 모니터 상에 쳐다보는 위치를 직접 계산하는 방법을 사용하였으며, 이러한 이유로 모니터와 사용자 얼굴 사이의 3차원 거리(depth)를 고정(50cm)시켜야 하고, 제한된 범위의 얼굴 움직임만 허용하여 만일 얼굴의 자연스러운 3차원 움직임이 발생하는 경우에 시선 위치 추출 에러가 증가되며, 또한 시선 위치 추적 시스템에 대해 학습된 제한된 사용자들만이 이용 가능한 문제점이 있었다. 이러한 문제점을 해결하기 위하여 Tomono 등[8]은 양안 카메라(stereo camera)를 이용하여 사용자 얼굴의 3차원 위치를 파악하고 이를 바탕으로 모니터상의 시선 위치를 파악하는 연구를 수행하였으나, 이러한 경우 두 대의 카메라 이용으로 시스템 가격이 상승하고, 두 대의 카메라로부터 영상이 입력되므로 영상 입력 속도가 저하되어 전체적으로 시스템 처리 속도가 감소되는 문제점이 있다. 이러한 기존의 연구들이 가지고 있는 문제점들을 해결하기 위하여 이 논문에서 얼굴의 3차원 이동 및 움직임 추정에 의해 시선 위치를 파악하는 방법을 사용한다. 사용자의 시선 위치를 파악해 내는 과정은 다음의 4 단계로 구성되어 있다. 첫 번째 단계에서는 초기에 자세 보정을 위해 모니터의 3지점을 쳐다보는 사용자의 얼굴 영상으로부터 얼굴 영역 및 얼굴 특징점(양 눈, 양 콧구멍, 입의 양 끝점)등을 추출한 후 이들의 2차원 카메라 영상에서의 움직임 정보로부터 3차원 위치를 추정해 낸다. 두 번째 단계에서는 이 후 모니터의 한 지점을 쳐다보기 위해 얼굴을 움직이는 순간(회전 및 이동), 얼굴의 3차원 움직임량을 추정해 낸다. 세 번째 단계에서는 추정된 얼굴의 3차원 움직임 정보 및 초기 얼굴 특징점의 3차원 위치 정보로부터 아핀 변환을 이용하여 변화된 얼굴 특징점의 3차원 위치를 구할 수 있게 된다. 그리고 마지막 단계에서는 이로부터 얼굴 특징점이 형성하는 평면 방정식을 구하게 되며 이 평면의 법선 방향을 가지는 직선과 모니터 평면이 만나는 점을 사용자의 시선 위치로 정할 수 있게 된다.

Ⅱ. 얼굴 영역 및 얼굴내 특징점 추출

사용자의 응시 위치를 파악하기 위하여, 이 논문에서는 얼굴내의 특징점(양눈, 코, 입)의 위치 변화 및 특징점들이 형성하는 기하학적인 모양의 변화도를 이용한다. 이를 위해 이 논문에서는 먼저 얼굴 영역을 검출한 후 추출된 얼굴 영역내의 제한된 범위 내에서 양 눈과 코 및 입의 양 끝점을 추출한다.

2.1 차영상 정보와 칼라 정보를 이용한 얼굴 영역의 추출

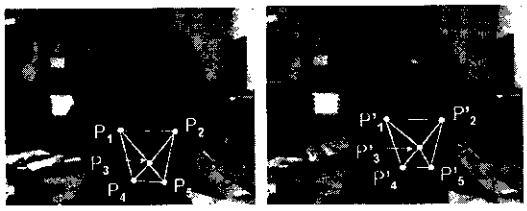
이 논문에서는 시간적으로 연속된 두 영상간의 차영상 정보와 칼라 정보를 이용하여 얼굴 영역을 검출한다. 얼굴의 살색 영상 처리부에서는 입력된 얼굴의 살색 칼라 정보에 대한 RGB신호를 YIQ model로 변환함으로써 얼굴의 살색 정보에 민감한 I성분 구간(110~150)을 바탕으로 얼굴 영역을 검출한다.

2.2 수평·수직 히스토그램 분석법을 이용한 눈동자, 코 및 입의 양 끝점 추출 및 움직임 추적

추출된 얼굴 영상은 히스토그램 평활화 및 이진화과정을 통해 이진 영상으로 변환한다. 이때, 얼굴 내의 눈/코/입의 위치에 대한 사전정보와 이진 영상에 대한 제한된 범위 내에서 수직, 수평히스토그램의 최고치를 계산함으로써 눈/코/입의 위치를 정확하게 추출할 수 있다. 또한 초기 영상에서의 특징점 추출 방법과는 달리 이후 연속 영상에서는 매번 얼굴 영역을 다시 추출하지 않고, 이전에 추출된 특징점의 위치로부터 현재 특징점의 위치를 예측하는 알고리듬을 사용함으로써 특징점의 움직임을 추적한다.

Ⅲ. 얼굴의 3차원 회전량 추정을 위해 사용하는 특징값

사용자의 응시 위치를 파악하기 위해 이 논문에서는 양눈과 코, 입의 위치를 특징점으로 사용하였다. 아래의 그림 1.(a)와 그림 1.(b)는 각각 모니터의 정중앙과 한 지점을 응시하는 순간에 추출된 특징점들의 위치를 나타낸 것이다. 이때, 추출된 특징점들로부터 응시 위치를 파악하기 위해 이 논문에서는 다음과 같은 20개의 특징값들을 신경망의 입력 노드로 사용하였다.



(a) 모니터 정 중앙응시 (b) 모니터 한 지점응시

그림 1. 모니터 정중앙과 일정 영역을 응시하는 순간의 특징점의 위치 변화

▷ 모니터의 정중앙을 응시 할 때
P1 (왼쪽눈 : X1, Y1), P2 (오른쪽눈 : X2, Y2),
P3 (코 : X3, Y3), P4 (입의 왼쪽 끝 : X4, Y4)
P5 (입의 오른쪽 끝 : X5, Y5)

▷ 모니터의 일정 영역을 응시 할 때
P'1 (왼쪽눈:X'1, Y'1), P'2 (오른쪽눈:X'2, Y'2),

P'3 (코:X'3, Y'3), P'4 (입의 왼쪽 끝:X'4, Y'4)
P'5 (입의 오른쪽 끝:X'5, Y'5)

특징값 1 ~ 5 : X'i-Xi (i = 1, 2, ... 5),
특징값 6 ~ 10 : Y'i-Yi (i = 1, 2, ... 5)
특징값 11 : S(△P'1P'2P'3) - S(△P1P2P3),
특징값 12 : S(△P'1P'3P'4) - S(△P1P3P4)
특징값 13 : S(△P'2P'3P'5) - S(△P2P3P5),
특징값 14 : S(△P'3P'4P'5) - S(△P3P4P5)
특징값 15 : S(△P'1P'3P'4)/S(△P'2P'3P'5) - S(△P1P3P4)/S(△P2P3P5)
특징값 16 : S(△P'3P'4P'5)/S(△P'1P'2P'3) - S(△P3P4P5)/S(△P1P2P3)
특징값 17 : {(X'1+X'4)/2- X'3} - {(X1 + X4)/2 -X3}
특징값 18 : {X'3-(X'2+X'5)/2 } - {X3 - (X2 +X5)/2}
특징값 19 : {(Y'4+Y'5)/2-Y'3} - {(Y4 + Y5)/2 -Y3}
특징값 20 : {Y'3-(Y'1+Y'2)/2} - {Y3 - (Y1 +Y2)/2}

그러나 사용자의 앉은 키와 모니터와의 거리에 따라 입력 특징값의 차이가 크다면 정확한 응시 위치를 나타낼 수 없을 것이다. 그러므로 이 논문에서는 정규화 과정을 통해 입력 특징값의 변화도를 수용하였다.

Ⅳ. 초기 얼굴 특징점의 3차원 거리 추정

초기에 모니터와 사용자까지의 3차원 거리를 구하기 위해 이 연구에서는 다음과 같은 얼굴/카메라/모니터 모델을 사용한다.

이때, 위의 그림 2와 같이 사용자가 모니터의 중앙과 다른 한 지점을 쳐다보는 순간 얼굴 좌표계 (X_f, Y_f, Z_f)에서 정의된 (X_0, Y_0, Z_0)는 β의 회전각만큼 회전하여 (X_t, Y_t, Z_t)위치로 변화된다. 다음 그림 3은 이러한 얼굴 이동 모델을 바탕으로, 얼굴 좌표계와 카메라 좌표계, 그리고 모니터 좌표계간의 변환 과정을 나타낸 것이다.

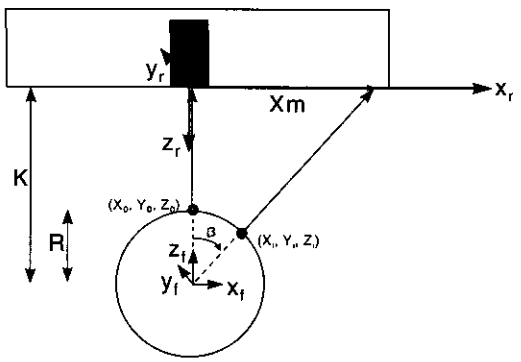
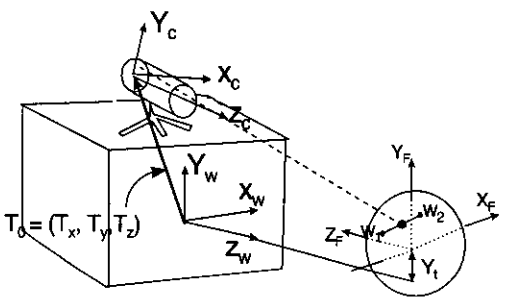


그림 2. 모니터의 중앙과 모니터의 오른쪽 한 지점을 응시하는 순간에 관측된 사용자의 얼굴 회전 모델



(X_F, Y_F, Z_F 와 X_W, Y_W, Z_W 간의 관계)

$X_W = X_F,\ Y_W = Y_F + Y_t,\ Z_W = -Z_F + K,$

그림 3. 얼굴좌표계, 모니터좌표계, 카메라 좌표계 간의 관계

이때, 위의 그림 3에서 카메라의 상하각을 α, 좌우각 $\theta = 0$으로 가정하고 모니터 좌표계와 카메라 좌표계 간의 translation 정도를 ($T_x = 0,\ T_y, T_z \neq 0$) 으로 가정하면, 얼굴 특징점의 3차원 좌표와 카메라에 투영된 2차원 좌표사이에는 다음 식 (1)과 같은 관계가 성립한다.

$$c_i = P \cdot R_\alpha \cdot T_0 \cdot w_i \qquad (1)$$

그런데 3차원 좌표 (X_0, Y_0, Z_0)와 (X_i, Y_i, Z_i) 는 얼굴 좌표계 (X_f, Y_f, Z_f) 에서 정의되어 있으므로, 이를 모니터 좌표계 (X_W, Y_W, Z_W)를 기준으로 변환한다. 이에 따라 3차원 좌표 $(X_{0_W}, Y_{0_W}, Z_{0_W})$와 $(X_{i_W}, Y_{i_W}, Z_{i_W})$가 2D image plane에 투영된 점인 (x_0, y_0, f)와 (x_i, y_i, f)를 식 (1)에서부터 구할 수 있으며, 앞의 $X_0 \simeq 0$, $Z_0 = R$ 를 이용하면, 최종적으로 사용자가 모니터의 중앙과 한 영역을 응시하는 순간에 2D camera에서 관측된 특징점의 x방향 움직임(d_x)과 모니터의 응시 영역에 대한 X축 변위 (X_m)사이에는 다음 식 (2)와 같은 관계가 성립한다.

$$d_x = x_i - x_0 = \frac{f \cdot \sin[atan(\frac{Xm}{K})] \cdot R}{((Y_0 + Y_t) \cdot \sin\alpha - (-\cos[atan(\frac{Xm}{K})] \cdot R + K) \cdot \cos\alpha - G)}$$

$$G = T_y \cdot \sin\alpha + T_z \cdot \cos\alpha + f) \qquad (2)$$

그런데 위의 식 (2)에서는 8개의 미리 알아야 할 변수 ($f, K, Ty, Tz, Yt, Yo, \alpha, R$)가 존재하며, 이는 크게 두 종류로 나누어, 사용자의 앉은 자세에 따라 달라지는 parameter와 사용자의 자세에 관계없이 카메라 자체의 setup 정보를 나타내는 parameter로 나눌 수 있다. 이중, 카메라 자체의 setup 정보를 나타내는 parameter로는 α(카메라의 상하각), Ty, Tz (모니터 좌표계로부터 카메라 좌표계로의 translation vector), f(카메라의 초점 거리)등이 있다. 또한 사용자의 앉은 자세에 따라 달라지는 parameter로는 Yt(모니터 좌표계에 대한 얼굴 좌표계의 Y축 translation vector), Yo(얼굴 좌표계내의 얼굴 특징점의 Y축 높이), K(모니터 좌표계에 대한 얼굴 좌표계의 Z축 거리), R(얼굴의 반지름)등이다. 이때 camera 자체의 setup 정보는 사용자의 앉은 상태나 거리등에 관계없이 변하지 않는 parameter이기 때문에, 본 연구에서는 보정점들을 이용한 camera calibration방법에 의하여 미리 측정한다. 이때 camera calibration에 의해 구한 camera setup parameter의 정확도를 측정하기 위해, 이 연구에서는 보정 panel의 위치에 대한 상하 거리 및 원근 거리등을 조정하여 새롭게 관측된 보정점들의 3D 위치와 위에서 구한 parameter값으로부터 추정된 보정점들의 3D 위치와 차이를 비교하였다. 이때 구해진 보정점들의 3D 위치와 실제 3D위치사이의 RMS error는 다음 표 1과 같다.

표 1. 구해진 보정점들의 3D 위치와 실제 3D위치와의 RMS error (단위 cm)

X	Y	Avg
0.079	0.1679	0.1237

위의 식 (2)에서 camera calibration 이후 남는 미지의 parameter로는 Yt(모니터 좌표계에 대한 얼굴 좌표계의 Y축 translation vector), Yo(얼굴 좌표계내의 얼굴 특징점의 Y축 높이), K(모니터 좌표계에

대한 얼굴 좌표계의 Z축 거리), R(얼굴의 반지름) 등의 4개가 있다. 이 연구에서는 이 4개의 parameter (K, R, Yt, Yo)를 구하기 위해 parameter estimation 과정을 수행하였다. 즉, 사용자가 모니터의 3 영역을 응시하게되면 3개의 얼굴 특징점(양눈의 중심, 코의 중심, 입의 중심)으로부터 6쌍의 (dx, Xm) 데이타를 얻을 수 있게 된다. 이로부터 식 (2)를 이용하여 parameter estimation 방법에 의해 미지의 4개의 parameter(K, R, Yt, Yo)들을 구하게된다. 이때, parameter estimation 방법으로 이 연구에서는 Gauss-Newton Method, Steepest Descent Method, Davidon-Fletcher-Powell method등을 비교하여 가장 우수한 성능을 나타내는 알고리듬을 사용하였다. 이때 매번 3가지 방법을 수행하여 성능이 가장 우수한 것을 선택하는 것이 아니라, 미리 얻어진 10명분의 데이터들로부터 위의 3가지 방법을 각각 수행하여 얻어진 성능으로부터 1가지(Davidon-Fletcher-Powell method)를 선택하여 사용하였다. 이로부터 구한 얼굴 특징점의 Z축 거리 ($Z = K - R$)와 앞에서 구한 camera setup parameter인 (f, Ty, Tz, α) 그리고 식 (I-13)으로부터 특징점의 3D 위치를 구할 수 있게 된다. 이때 추정한 얼굴 특징점의 정확도를 구하기 위해 이 연구에서는 polhemus sensor와 비교하였다. 실험환경은 19인치 모니터를 사용하였고 사용자는 모니터와 50-70cm정도의 거리만큼 떨어져 있을 때이다. 그러나 이렇게 추정한 얼굴 특징점의 위치 자체에도 에러 요소가 있기 때문에 이 연구에서는 다음과 같은 초기 특징점의 위치 보정 알고리듬을 사용한다. 앞에서 구한 얼굴 특징점의 위치는 사용자가 모니터 중앙을 볼 때 추정된 값이다. 그러므로 이때 구한 얼굴 특징점(f_1, f_2, f_3)로부터 얼굴 평면(M)과 이의 법선(L)을 구할 수 있으며 이때 L과 모니터 평면과 만나는 위치가 사용자의 응시 위치가 된다. 즉, 이때 계산된 응시 위치가 모니터의 중앙에 놓일 수 있도록 추정된 얼굴 특징점의 위치를 보정하게 되는 것이다. 이러한 보정 결과 다음 표 2와 같이 보다 정확한 얼굴 특징점의 3차원 위치를 얻을 수 있었다.

표 2. 보정된 얼굴 특징점의 초기 3D위치와 polhemus position sensor로부터 구한 특징점의 3D 위치와의 RMS error (단위 cm)

in X	in Y	in Z	Total
0.64	0.81	0.5	1.15

V. 신경망에 의한 얼굴의 3차원 회전량 추정

얼굴의 3차원 움직임량을 추정하기 위해 사용하는 특징값은 앞에서 소개한 20개의 특징값들을 사용한다. 다음 그림4는 얼굴의 회전량을 추정하기 위해 사용한 신경망 구조이다. 신경망으로는 다층 퍼셉트론을 사용하였으며, 신경망의 학습으로는 역전파 알고리듬(BackPropagation)방법을 사용하였다. 신경망의 출력은 X, Y축의 연속적인 회전각을 나타낼 수 있도록 연속적인 선형 함수(Continuous Linear Function)를 사용하였다. 입력 노드 수는 20개이며, 은닉 노드 수는 일반적으로 입력 노드 수의 60% ~ 70% 정도가 적합한 것으로 알려져 있으므로 12 ~14개를 각각 실험하였다. 실험 결과 12 ~ 14개 사이에 성능 변화가 없는 것으로 나타났으며, 이로부터 처리 속도를 고려하여 12개를 은닉 노드 수로 사용하였다.

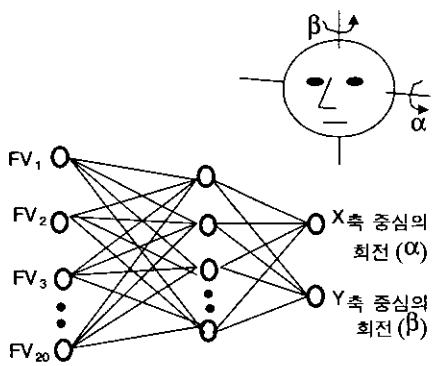


그림 4. 얼굴의 3차원 회전량을 추정하기 위한 신경망 구조

전술한 바와 같이 출력 노드로는 X, Y축 회전각을 나타내는 2개를 사용한다. 학습 데이터를 얻기 위하여 polhemus position tracker sensor얼굴에 부착하여 얼굴이 회전하는 순간 취득된 영상 및 polhemus 측정 데이터를 이용하였다. 신경망의 학습을 위해서는 모니터 중앙을 볼 때와 화면상의 한 지점을 볼 때의 차이를 학습하는 방법을 사용하였다. 이때 한 가지 문제점으로는 모니터에서 사용자까지의 거리가 신경망을 학습시킬 와 실제로 test할 때 차이가 커진다면 3차원 회전각 추정에 많은 에러가 발생할 것이라는 것이다. 그러므로 이 연구에서는 IV절의 방법으로 일단 초기에 모니터에서 사용자까지의 거리를 측정한 후에는 이 거리의 변화도가 크지 않는다고 가정했으며, 또한 초기에 사용자가 모니터의

중앙을 볼 때 구한 얼굴의 폭 정보에 대한 현재 입력 얼굴의 폭 정보를 비교함으로써 모니터에서 사용자까지의 거리에 대한 변화도를 어느정도 흡수할 수 있도록 하였다. 신경망의 학습을 위하여 42곳의 시선 위치를 응시할 때 추출된 420개(42시선 위치 ×10명분)의 회전각 데이터를 사용하였다. 이때 출력함수로는 학습된 42영역에 대한 얼굴의 회전각이외의 회전각을 출력으로 나타낼 수 있도록 미분 가능하며 연속적인 출력함수들을 사용하였다. 추정된 3차원 움직임량의 정확도는 다음 그림 5와 같이 polhemus position tracker sensor와 비교하였다.

(X축 중심의 회전)

(Y축 중심의 회전)

그림 5. 얼굴의 3차원 회전량 추정을 위한 실험 데이터의 예

실험 결과 신경망에 의해 추정한 회전각과 polhemus sensor에 의해 측정된 회전각사이에는 평균 3.1 도의 RMS error가 존재함을 알 수 있었다.

Ⅵ. 얼굴의 3차원 이동량 추정

이 논문에서는 얼굴의 3차원 이동량을 추정하기 위하여 얼굴 윤곽의 이동량을 이용하였다. 여기서 얼굴의 윤곽을 추출하기 위하여 다음 그림 6과 같은 방법을 사용하였다.

그림 6의 (a)에서 얼굴 영상이 입력되면 (b)에서 2방향 sobel edge operator를 이용하여 edge 영상을 얻고 (c)와 같이 이진영상으로 변환한다. 이때 이진화 임계치를 높이는 경우에 얼굴내부에는 거의 에지 성분이 남지 않게 된다. 그리고 이 순간 입력 영상 내에서는 얼굴 특징점, 특히 양눈의 위치를 추적하고 있는 상태이므로 (d)와 같이 추출된 양 눈을 중심으로 외곽으로 탐색하면서 만나는 에지 부분을 얼굴 윤곽으로 정할 수 있게 된다. 그림 (e)는 추출

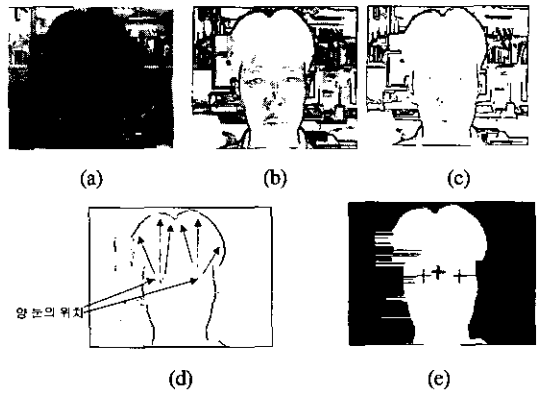


그림 6. 얼굴의 특징점 추적 정보를 이용한 얼굴 윤곽 추출

된 얼굴 영역내의 무게중심점과 양 눈의 위치를 나타낸 것이다. 이와 같은 방법으로 얼굴 윤곽 및 무게 중심 추출의 정확도를 높일 수 있게 되었다. 이 시점에서 이때 추출된 얼굴 무게중심의 이동량이 실제 얼굴의 3차원 이동량을 제대로 반영하는지를 조사해 볼 필요가 있을 것이다. 이를 위하여 이 연구에서는 다음과 같은 실험을 하였다. polhemus sensor를 이용하여 얼굴의 3차원 이동량을 측정하는 동안 취득된 영상에 대해 앞에서 소개한 윤곽 추출 방법을 이용하여 얼굴 무게 중심점의 2차원 이동량을 측정해 보았다. 실험 결과 모니터와 사용자까지의 거리가 50cm일 때 3차원 공간에서 약 3.04cm의 이동량에 대하여 2차원 영사에서는 31 pixel의 이동량으로 측정되었다. 그러면 이 시점에서 과연 이 값이 얼마나 정확하게 3차원 이동량을 나타내는지 알아보기 위하여 이 연구에서는 다음과 같은 실험을 하였다. 수평, 수직으로 6cm간격으로 표시된 점이 있는 보정판을 모니터 앞 50cm 거리 설치하고 이때 카메라에 의해 관측된 영상내의 점 사이의 간격을 조사한다. 실험 결과 모니터와 카메라가 50cm 떨어졌을 때 보정판의 6cm는 취득된 영상의 63 pixel로 대응됨을 알 수 있었다. 이로부터 앞의 실험의 정확도를 다음 식 (3)에 의해서 계산할 수 있다.

$$6 : 63 = X : 31 \rightarrow X = 2.95 \text{ cm} \qquad (3)$$

즉 취득된 영상에서의 31 pixel 간격은 실제에서는 2.95cm의 거리를 나타낸다. 이때 polhemus sensor에 의해 실제로 측정된 3차원 이동량은 3.04cm이므로 약 0.09 cm의 오차를 보임을 알 수 있었다. 이외에도 보다 많은 데이터에 대해 실험한 결과 X축

방향으로는 0.29cm, Y축 방향으로는 약 0.45cm의 에러를 나타냄을 알 수 있었다. 즉, 실험 결과 모니터에서 사용자까지의 거리를 알고 있는 상황에서는 추출한 얼굴 윤곽 및 무게중심의 이동량이 비교적 정확하게 얼굴의 실제 3차원 이동량을 반영함을 알 수 있었다. 여기서 모니터와 사용자까지의 3차원 거리는 앞의 IV절에서 구한 얼굴 특징점의 3차원 위치 정보를 이용하여 구하며, 이로부터 얼굴의 3차원 이동량을 구할 수 있게 된다. 이때 추정된 얼굴의 3차원 이동량의 정확도를 구하기 위하여 이 연구에서는 polhemus sensor를 이용하였으며, 실험 결과 약 1.8cm(X축으로는 1.1 cm, Y축으로는 1.42cm)의 이동량 추정 에러를 나타냈다.

Ⅶ. 모니터상의 응시 위치 파악

III절에서 구한 얼굴 특징점(양눈, 입의 중심점)의 초기 3차원 위치는 모니터 좌표계를 기준으로 추정된 값이므로 이 연구에서는 이를 그림 3의 식을 이용하여 얼굴 좌표계의 값으로 변환한다. 얼굴 좌표계를 기준으로 정의된 3개의 특징점의 3차원 좌표는 V절에서 구한 3차원 회전량 및 VI절의 방법으로 구한 3차원 이동량 결과와 함께 다음 식 (4)와 같이 아핀 변환 과정을 거쳐 변화된 얼굴 특징점의 3차원 위치가 되다.

$$\begin{bmatrix} X'_i \\ Y'_i \\ Z'_i \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha \\ 0 & -\sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} + \begin{bmatrix} T_X \\ T_Y \\ 0 \end{bmatrix} \quad (4)$$

(i = 1, 2, 3) 왼쪽 눈, 오른쪽 눈, 입의 중점

변화된 3개의 얼굴 특징점의 3차원 위치 (X'_1, Y'_1, Z'_1), (X'_2, Y'_2, Z'_2), (X'_3, Y'_3, Z'_3)로부터 다시 그림 3의 관계식을 이용하여 모니터 좌표계를 기준으로 한 특징값으로 변환하고 이로부터 이 3점으로 구성된 평면을 구할 수 있게 된다. 3차원 공간에서 일반적인 평면 방정식으로 다음과 같이 정의G할 때,

$$AX+BY+CZ=D \quad (5)$$

이로부터 사용자의 시선 방향은 다음 그림 7과 같이 양 눈, 입의 중심점으로 구성된 평면의 법선 벡터 방향 (A, B, C)을 가지고 동시에 양 눈의 중심점을 지나는 직선과 모니터와 만나는 점이 사용자가 모니터를 쳐다보는 위치가 된다.

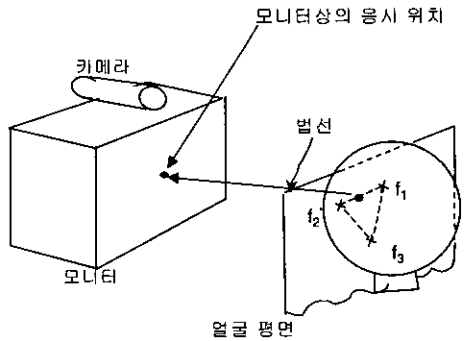


그림 7. 양눈 및 입의 중점을 포함한 평면의 법선에 의한 시선 위치 추출

모니터상의 응시 위치 :

$$X_m=-\frac{A}{C}*\overline{Z}+\overline{X}, \qquad Y_m=-\frac{B}{C}*\overline{Z}+\overline{Y}$$

$$(\text{단, } \overline{X}=\frac{X_1+X_2}{2},\ \overline{Y}=\frac{Y_1+Y_2}{2})$$

여기서 직선의 시점을 양 눈과 입의 무게중심으로 하지 않고 양 눈의 중심으로 정한 것은 사람의 시선 방향이 양 눈의 중심에 위치하기 때문이다. 그러나 이렇게 계산된 사용자의 시선 위치에는 3차원 회전량 및 이동량 추정 에러가 포함되어 있다. 그러므로 이러한 에러 요인을 줄이기 위해 이 연구에서는 다음과 같은 추가적인 알고리듬을 사용하였다. 즉, 앞의 III절에서 구한 특징점의 초기 3차원 위치로부터 3차원 상의 양 눈 사이의 거리(L_1), 왼쪽 눈과 입의 중심사이의 거리(L_2), 그리고 오른쪽 눈과 입의 중심 사이의 거리(L_3)를 구하게 된다. 이렇게 초기에 특징점들 사이의 3차원 거리를 구하게 되면 이 거리는 얼굴이 회전하거나 이동하더라도 변화되지 않아야 한다. 그러므로 이 연구에서는 식 VI절에서 구해진 특징점의 3차원 위치로부터 다시 한번 3차원 상의 양눈사이의 거리($\overline{L_1}$), 왼쪽눈과 입의 중심사이의 거리($\overline{L_2}$), 그리고 오른쪽 눈과 입의 중심 사이의 거리($\overline{L_3}$)를 구하고 다음 식 (6)에서처럼 특징점들 사이의 3차원 거리 변화도 (T)가 최소가 되는 방향으로 특징점의 3차원 위치를 변화시킴으로써 보다 정확한 얼굴 특징점의 위치 및 모니터상의 시선 위치를 구할 수 있게 된다.

$$T=\sqrt{(L_1-\overline{L_1})^2+(L_2-\overline{L_2})^2+(L_3-\overline{L_3})^2} \quad (6)$$

시선 위치 추적의 정확도는 19인치 모니터 앞에 약 50~70 cm떨어진 거리에서 10명의 사용자가 쳐다보는 실제 위치와 시선 위치 추적 알고리듬에 의해 파악된 위치사이의 RMS error로써 측정하였다. 이 때 시선 위치 추적의 정확도는 표 3과 같이 모니터 상의 임의의 지점을 쳐다볼 때 얼굴의 회전만 존재하는 경우와 회전 및 이동이 같이 발생하는 경우로 나누어 실험하였다.

표 3. 모니터 상의 시선 위치 파악 정확도 (단위 : cm)

	얼굴의 회전만 존재하는 경우	얼굴의 회전과 이동이 같이 발생하는 경우
선형 보간법[1][3]	4.67	11.5
단일 신경망[11][9]	4.16	11.1
제안하는 방법	4.24	5.28

실험 결과, 얼굴의 회전만 존재하는 경우보다 회전과 이동이 같이 발생하는 경우에 시선 위치 에러가 조금 증가됨을 알 수 있었다. 또한 사용자의 얼굴의 회전만이 있는 경우에는 단일 신경망 방법[11][9] 만으로도 좋은 성능을 나타내며 약 4.16cm의 평균 오차를 보이는 것을 알 수 있었다. 그러나 얼굴의 회전과 이동이 함께 허용되었을 경우에 단일 신경망 방법만을 사용하는 방법은 약 11.1 cm의 응시위치 추정 오차를 보이고 있으며, 이 경우 본 논문에서 제안하는 방법이 보다 우수한 성능(5.28cm)을 나타냄을 알 수 있다. 에러는 입력 영상에서 얼굴 특징점 및 얼굴 윤곽 추출의 에러, 얼굴 특징점의 3차원 위치 및 움직임 추정 에러들이 합쳐진 결과이다. 또한 이 결과는 Rikert의 논문[7]에서 나타낸 시선 위치 추적 성능(5.08 cm 에러)과 비슷한 결과를 나타낸다. 그러나 Rikert의 방법은 얼굴의 3차원 위치 및 움직임에 대한 고려 없이 카메라에서 관측된 2차원 얼굴 영상으로부터 모니터상의 시선 위치를 직접 파악하므로, 모니터와 사용자 얼굴까지의 거리는 항상 고정(50cm)시켜야한다는 단점이 있으며, 얼굴의 자연스러운 움직임(회전 및 이동)이 발생하는 경우 시선 위치 추적 에러가 증가되는 문제점이 있다. 동시에 그들의 방법은 사용자 얼굴의 뒤 배경에 복잡한 물체가 없는 것으로 제한 조건을 두고 있으며 처리 시간이 상당히 오래 걸리는 문제점이 있다 (333MHz alphastation에서 약 1분). 그러나 본 논문에서 제안하는 시선 위치 추적 방법은 배경이 복잡한 사무실 환경에서도 사용가능하며, 약 3초 이내의 처리 시간(200MHz Pentium PC)이 소요됨을 알 수 있었다. 두 번째 실험에서, 모니터와 사용자 얼굴사이의 거리를 변화 시켜가면서(55, 60, 65 cm) 시선 위치 추출의 정확도를 측정하였다. 실험 결과 RMS는 다음과 같았다. 55cm거리에서 4.85 cm, 60cm거리에서 5.09 cm 그리고 65cm 거리에서 5.13 cm. 이로부터 모니터와 사용자간의 거리에 관계없이 본 논문에서 제안한 시선 위치 추출 방법은 거의 유사한 성능을 나타냄을 알 수 있었다.

Ⅶ. 결 론

이 논문에서는 카메라 및 영상 입력 장비 외에 특별한 장비 없이 컴퓨터 비전 방법에 의해 시선 위치 추적 알고리듬을 구현하였다. 모니터 상의 시선 위치를 파악하기 위해 본 논문에서는 2차원 카메라 영상으로부터 얼굴 영역 및 얼굴 특징점을 자동으로 추출하였으며, 이로부터 카메라 보정, 매개변수 추정 방법 및 3차원 움직임 추정 방법등을 이용하여 모니터 상의 시선 위치를 구하였다. 이때 본 논문에서는 눈동자의 움직임은 거의 없다고 가정했으며 이에 대한 추가 연구는 현재 진행중이다.

실험 결과 19인치 모니터를 사용하여 모니터와 사용자까지의 거리를 50~70cm정도 유지하였을 때 약 2.08인치의 응시 위치 에러 성능을 얻었으며, 처리시간은 Pentium PC환경에서 320×240 pixel 크기의 영상을 사용할 때 총 3초 이내가 됨을 알 수 있었다. 향후 얼굴 특징점 및 윤곽 추출의 정확도를 향상시키고 그리고 눈동자 움직임을 추가로 고려한다면 보다 정확한 시선 위치 추출 성능을 얻을 수 있을 것으로 기대된다.

참 고 문 헌

[1] Jaihie Kim, Kang Ryoung Park, Jeoung Jun Lee, S.R.LeClair, "Intelligent Process Control via Gaze Detection Technology", Engineering Applications of Artificial Intelligence, Vol. 13, pp. 577-587, Aug. 2000

[2] A. Azarbayejani, "Visually Controlled Graphics", IEEE Trans. PAMI, Vol. 15, No. 6, pp. 602-605, June, 1993

[3] T. Fukuhara, T. Murakami, "3D-motion estimation of human head for model-based

image coding", IEE Proc., Vol. 140, No. 1, pp.26-35, 1993.

[4] P. Ballard, G. Stockman, "Controlling a Computer via Facial Aspect", IEEE Trans. on System Man and Cybernetics, Vol. 25, No.4, pp.669-677, 1995.

[5] A. Gee, R. Cipolla, "Fast visual tracking by temporal consensus", Image and Vision Computing, Vol. 14, pp. 105-114, 1996.

[6] J. Heinzmann, A. Zelinsky, "3-D Facial Pose and Gaze Point Estimation using a Robust Real-Time Tracking Paradigm", Proceedings of the International Conference on Automatic Face and Gesture Recognition, pp.142-147, 1998.

[7] T. Rikert, M. Jones, "Gaze Estimation using Morphable Models", Proceedings of the International Conference on Automatic Face and Gesture Recognition, pp.436-441, 1998.

[8] A. Tomono, F. Kishino, "Gaze Point Detection Algorithm Based on Measuring 3D Positions of Face and Pupil", IEICE Transactions on Information and Systems}, D-II, Vol. J75-D-II, No.5, pp.~861-872, 1992. 5.

[9] 박강령, "얼굴의 2차원 및 3차원 움직임을 이용한 시선 위치 추적에 관한 연구", 연세대학교 전기·컴퓨터 공학과 대학원 박사 졸업 논문, 2000, 2

박 강 령(Kang-Ryoung Park) 정회원

1994년 2월 : 연세대학교 전자공학과 졸업
1996년 2월 : 연세대학교 전자공학과 석사
2000년 3월 : 연세대학교 전기·컴퓨터공학과 박사

<주관심 분야> Biometric영상처리, 패턴인식, 컴퓨터vision

김 재 희(Jaihie Kim) 정회원

1982년 8월 : Case Wetsern Reserve Univ. Electrical Eng. 석사
1984년 5월 : Case Wetsern Reserve Univ. Electrical Eng. 박사
1984년 3월~현재 : 연세대학교 전기·컴퓨터공학과 교수

<주관심 분야> Biometric영상처리, 패턴인식, 컴퓨터vision