3D 디스플레이 디스페리티 제어를 위한 안간 거리 측정 방법

박지훈 · 이재성
한국교통대학교 전자공학과

Pupil Distance Measurement for 3D Display Disparity Control

Ji-hoon Park · Jaesung Lee
Department of Electronic Engineering, Korea National University of Transportation
E-mail : jaesung.lee@ut.ac.kr

요 약

3D-TV 시청시 눈에 피로를 주는 원인은 시청자의 안간 거리와 시청거리에 상관없이 고정된 disparity 의 좌/우 영상이 재생되기 때문이다. 본 논문은 TV 단말의 양쪽에 카메라 모듈을 이용하여 시청자의 위치와 눈동자 간격을 알아내어 3D TV 영상재생에 그 정보를 반영하는 방법을 제안한다.

ABSTRACT

The reason why visual fatigue occurs while watching 3D TV is because the left/right images of the fixed disparity are played. This paper proposes a measuring method of pupil distance and the 3D location of a viewer using stereo camera modules, and a reflection method of the data into 3D TV.

키워드
눈동자 검출, 거리측정, 눈동자 간격 측정

Ⅰ. 서 론

2009년 말 영화 '아바타'의 개봉, 2010년 'FIFA 월드컵' 스테레오 3D (Stereoscopic 3D 또는 S3D) 중계를 시작으로[1] 입체 콘텐츠에 대하여 많은 사람들이 관심을 갖게 되었고 최근에 이르러서는 박스 오피스 상위권 작품들 대부분 S3D 상영을 병행하고 있으며 삼성, LG 등 세계 유수 가전업체들은 3DTV 판촉에 열을 올리고 있다. 이처럼 3D 입체 콘텐츠는 빠르게 우리의 생활의 한 부분으로서 자리매김을 하고 있는 중이다. 입체 영상은 양 눈에 각각 서로 다른 영상 신호로 전달되어 뇌에서 공간에 대한 개념과 함께 재구성되는 미디어이다. 그러나 이러한 인간의 입체 인지 시스템(3D perception system)은 극도로 민감하기 때문에 하나의 카메라로 촬영된 기존 2D 영상을 시청하는 경우와 달리 시청자의 양안 시차(binocular disparity)와 시청 거리가 컨텐츠의 제작 당시 적용된 기준 값들과 조금만 달라지더라도 실세계에서 인간이 느끼는 입체감과 커다란 괴리가 발생하게 되고 이에 따라 극심한 시각 피로와 두통을 유발하게 된다. 부연 설명하자면 3D-TV 용 영상을 촬영할 때는 고정 간격의 두 대의 카메라를 이용하여 일정거리에서 촬영하고 출력 TV화면은 사람들의 평균적인 눈 간격에 대해서만 출력하므로 눈동자 간격이 사람마다 다르고 화면으로 부터의 시청 거리 차이가 다르면 초점이 맞지 않아 문제가 발생하는 것이다. 따라서 눈동자 간격을 실시간으로 측정해 이를 3D 화면의 재생 과정에 반영해주면 이러한 문제는 해결될 것이다.

본 논문에서는 카메라 두 대를 이용해 사람의 눈동자들을 촬영 후 눈동자간 거리를 측정하는 방법을 소개한다. 그러기 위해서 우선 얼굴영역 인지한 후 그 영역 안에서 눈동자를 검출한다.

II. 본 론

1.얼굴영역 검출

촬영된 시청자의 영상에서 눈을 찾기 위해 우선 얼굴 영역 인식부터 시행한다. 이러한 이유는 circular hough transform 에 의해 눈을 찾으면 눈이 아닌 부분도 눈인 것으로 인지되는 경우가 종종 발생하는데 이를 필터링하기 위해 얼굴 영역부터 찾은 후 영역의 상단에 있는 눈으로 추정되는 원들만 눈동자로 인식하기 위함이다.

얼굴을 인식하는 방법은 Viola Jones[2] 알고리즘을 이용한다. 이 방식은 특징 점을 이용해 얼굴을 검출한다. 아래 그림의 haar-like feature 라는 사각 모양의 그림들을 이용하여 얼굴의 특징 점들을 인지하여 통계를 낸 결과(haar training 및 ada-boost)를 활용하는 방법인데 각 그림의 흰 색 영역 아래에 위치한 픽셀들의 밝기 값의 합에서 회색 영역 아래 위치한 픽셀들의 밝기의 값의 합을 뺀 값(threshold)을 기준으로 통계를 낸다.

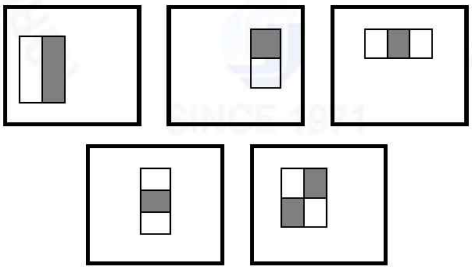


그림 1. 사각 특징 점들의 모양

얼굴표본의 사진 크기가 24x24라면 위의 특징 점들을 이용해, 얼굴 데이터 이미지들과 얼굴이 아닌 이미지를 이용하여 정면 얼굴 검출에 필요한 특징 점들을 학습을 통해서 얻는다. 이제 새로운 사진에 대한 특징 점들이 얼굴이라 판단할 수 있는 어떤 임계값 이상일 때 얼굴로 판단하게 한다.

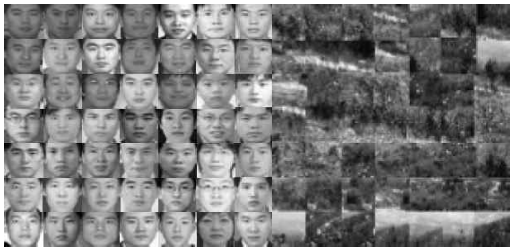


그림 2. 얼굴 데이터와 얼굴이 아닌 데이터

2. 눈동자 검출

얼굴영역을 찾았다면, 그 안에서 눈을 검출해야 한다. 눈을 검출하기 위해 얼굴영역 안에서 원을 찾으면 되는데 이때 원의 크기는 얼굴영역의 크기에 대한 일정한 비율로 정한다. 홍채는 이상적인 원과 아주 흡사하여 얼굴에 있는 다른 부위들 보다 가장 원에 가깝다. 따라서 보다 까다롭게 원을 찾는다면 눈을 찾을 수 있다.

눈을 찾는 원리로는 Circle Hough Transform [3]을 이용한다. 이것은 사진의 각 에지에 대하여 설정된 지름의 원에대하여 픽셀 값을 부여하여 원이라 판단할 수 있는 임계값보다 많이 축적된 지점을 찾는 방식이다. 그림에서 보듯이 원의 중심에는 매우 많은 원이 교차하여 매우 높은 값을 갖게 되는 것이다.

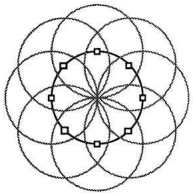


그림 3. Circle Hough Transform 원리

이제 얼굴 영역 안에서 원을 찾았으면 찾은 원 중에서 어떤 것이 눈동자인지 판별해야 한다. 우선 첫 번째로 눈동자는 눈동자 밖에 있는 흰자보다 훨씬 어둡다. 따라서 원의 안의 밝기보다 원 밖의 밝기가 훨씬 밝아야 한다.

$$(x-i)^2+(y-j)^2 \le r^2$$

$$c=\cos^{-1}\left(\frac{y-j}{r}\right)$$

$$T_1=\sum_{y=j-r}^{j+r}\sum_{x=i-c}^{i+c} S_{x,y}$$

$$T_2=\sum_{y=j-r}^{j+r}\sum_{x=i-c}^{i+c} \times 1$$

여기서 i, j는 원의 중심의 x, y 좌표이며, c는 원의 수직 반경에 따른 수평 반경이며, $S_{x,y}$은 해당 좌표에 대한 픽셀의 밝기 값이며, T_1는 원의 안쪽의 픽셀들 밝기 값들의 합이고, T_2는 원의 안쪽 픽셀의 개수이다. 원의 바깥쪽에 대해서도 위와 같이 한 후 픽셀에 대한 픽셀의 값 평균을 비교하여 바깥쪽이 안쪽보다 요구되는 비보다 높으면 그것을 눈동자 후보로 한다.

그런 다음, 각 원에 대해서 서로에 대한 반지름의 크기의 비가 크지 않은 것과, 두 원의 반지름에 대한 두 원의 거리의 비가 원하는 범위 안에 있다면 이것을 눈동자로 인정하는 것이다.

이렇게 하면 사진에서 홍채의 중심 좌표를 얻게 되며, 이것을 이용하여 두 홍채 중심점 사이의 거리를 측정할 수 있게 된다.

3. 거리측정

카메라에서 각 픽셀 좌표는 3차원 공간상에 해당되는 직선에 대응한다.

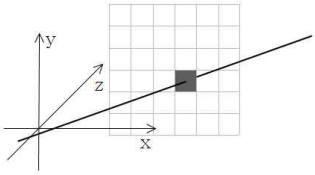


그림 4. 픽셀 값에 대한 직선

카메라가 두 대라면 하나의 눈동자에 대해 두 개의 직선이 그 눈동자 중심을 통과할 것이다. 이 두 직선이 교차하는 지점이 눈동자의 3차원 상의 공간 좌표가 된다. 왼쪽과 오른쪽 눈동자에 대한 공간 좌표를 구하면 이제 눈동자 사이의 거리와, 카메라에 대한 거리를 구할 수 있게 된다.

3차원 공간상에서는 두 직선이 만나지 않는 경우가 대부분이므로 이미 카메라 측정과 원 검출에 의한 오차가 들어가 있는 홍채 좌표로는 실질적으로 두 직선이 만나는 경우가 없다. 결국 두 직선이 서로 가장 가까운 지점을 찾아야 한다. 두 직선이 가장 가까운 곳을 이은 직선은 직선 A와 직선 B에 수직인 것은 자명하다.

따라서 직선 A와 직선 B에 동시에 수직인 직선을 찾아야 한다. 이 A와 B에 수직인 직선과 교차하는 부분이 바로 A, B 두 직선에 가장 가까운 지점이 된다.

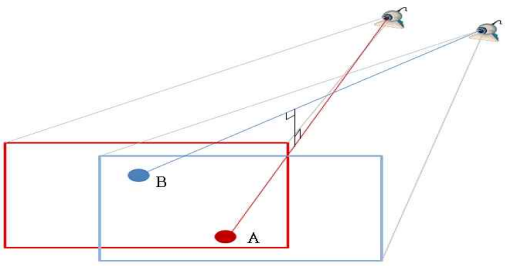


그림 5. 위 그림은 각 카메라에서 원의 좌표에 대한 직선을 표현한다.

두 직선에 수직인 직선의 3차원 벡터는 A와 B의 벡터 곱이다 두 직선과 수직인 직선의 벡터와 A의 직선을 또다시 벡터 곱을 하면 직선 A와, A와 B를 포개는 평면의 벡터가 나오며 이평면과 직선 B가 교차하는 부분이 직선 B가 직선 A에 대하여 가장 가까운 부분이 된다.

계산하는 순서는 A와 B를 벡터 곱을 하며 다시 이것을 둘 중 하나의 직선과 벡터 곱을 한 평면이 다른 직선과 만나는 지점을 찾는다.

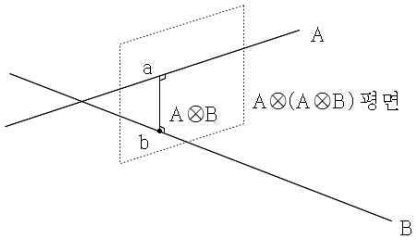


그림 6. A⊗B벡터 직선과 A⊗A⊗B벡터 평면

각 카메라가 좌표 (-R/2, 0, 0)와 (R/2, 0, 0)에 있다면 각 카메라에서의 찾은 눈동자 픽셀의 3차원 공간상의 직선 방정식은

$$l_1 = (a_1 t + R/2,\ b_1 t,\ c_1 t)$$
$$l_2 = (a_2 t + R/2,\ b_2 t,\ c_2 t)$$
$$l_3 = (a_3 t - R/2,\ b_3 t,\ c_3 t)$$
$$l_4 = (a_4 t - R/2,\ b_4 t,\ c_4 t)$$

l_1은 왼쪽 카메라의 왼쪽 눈을 나타내며, l_2는 왼쪽 카메라의 오른쪽 눈, l_3은 오른쪽 카메라의 왼쪽 눈, l_4는 오른쪽 카메라의 오른쪽 눈을 나타낸다. 따라서 계산할 때에는 l_1은 l_3과하며, l_2는 l_4와 하게 된다.

직선 1과 직선 3의 벡터 곱을 한다면

$$l_1 \otimes l_3 = \begin{pmatrix} x & y & z \\ a_1 & b_1 & c_1 \\ a_3 & b_3 & c_3 \end{pmatrix} =$$
$$((b_1 \times c_3 + c_1 \times b_3), (c_1 \times a_3 + a_1 \times c_3), (a_1 \times b_3 + b_1 \times a_3))$$

이 벡터와 직선 1에 의한 평면을 구한다면

$$l_1 \otimes (l_1 \otimes l_3) = \begin{pmatrix} x & y & z \\ a_1 & b_1 & c_1 \\ b_1 c_3 - c_1 b_3,\ c_1 a_3 - a_1 c_3,\ a_1 b_3 - b_1 a_3 \end{pmatrix}$$
$$= (((a_1 b_1 b_3 - a_3 b_1 b_1) - (a_3 c_1 c_1 - a_1 c_1 c_3)),$$
$$((b_1 c_1 c_3 - b_3 c_1 c_1) - (a_1 a_1 b_3 - a_1 a_3 b_1)),$$
$$((a_1 a_3 c_1 - a_1 a_1 c_3) - (b_1 b_1 c_3 - b_1 b_3 c_1)))$$

여기서 각각의 x, y, z에 대한 계수를A, B, C로 정의하면

$$l_1 \otimes (l_1 \otimes l_3) = (A_1, B_1, C_1)$$
$$l_2 \otimes (l_2 \otimes l_4) = (A_2, B_2, C_2)$$
$$l_3 \otimes (l_3 \otimes l_1) = (A_3, B_3, C_3)$$
$$l_4 \otimes (l_4 \otimes l_2) = (A_4, B_4, C_4)$$

이평면과 만나는 직선의 좌표를 구하는 식은, 평면의 방정식이 $A_1 x + B_1 y + C_1 z = A_1 R/2$ 라면

$$A_1 (a_3 t - R/2) + B_1 b_3 t + C_1 c_3 t = A_1 R/2$$
$$A_1 a_3 t + B_1 b_3 t + C_1 c_3 t = A_1 R/2 + A_1 R/2$$
$$t_3 = \frac{A_1 R}{A_1 a_3 + B_1 b_3 + C_1 c_3}$$

이렇게 구한 매개 변수로 각 직선에서 해당하는 좌표를 구하면 홍채의 3차원상의 공간의 좌표를 알 수 있게 되며 이것을 피타고라스의 정리를 이용하면 거리를 구할 수 있다.

각각의 좌표는

$$(x_1, y_1, z_1) = (a_1 t_1 + R/2,\ b_1 t_1,\ c_1 t_1)$$
$$(x_2, y_2, z_2) = (a_2 t_2 + R/2,\ b_2 t_2,\ c_2 t_2)$$
$$(x_3, y_3, z_3) = (a_3 t_3 - R/2,\ b_3 t_3,\ c_3 t_3)$$
$$(x_4, y_4, z_4) = (a_4 t_4 - R/2,\ b_4 t_4,\ c_4 t_4)$$

이것으로 거리를 구하면 된다.

$$L = (\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (y_1 - y_2)^2} + \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2 + (y_3 - y_4)^2})/2$$

III. 실제 측정과 성능평가

그림 7. 찍은 사진

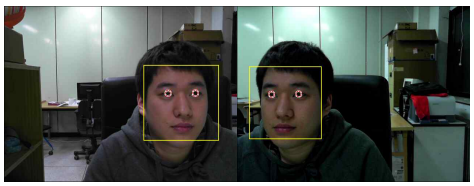


그림 8. 얼굴영역과 눈동자 검출이 된 사진

아래의 표는 카메라에 대한 눈동자의 거리와 눈동자 간격을 실제 값과 논문에서 사용한 방법으로 측정한 값에 대한 표이다. 측정한 데이터는 각 25회 측정한 데이터이며, 카메라간의 사이 간격은 30cm로 하였다.

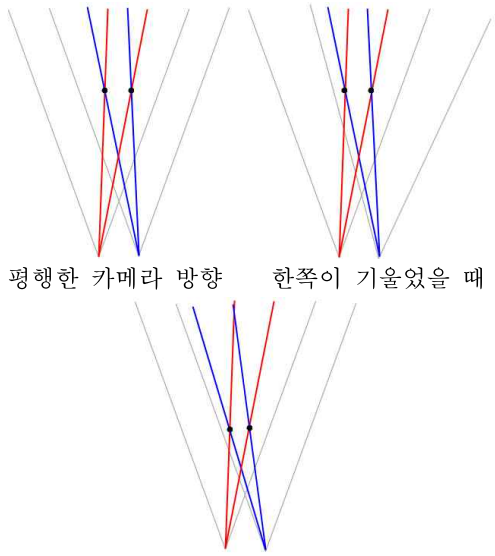
카메라거리	초점거리	실제 눈동자 간격	측정된 눈동자 간격	측정 표준편차	변이계수
60cm	60cm	64.5mm	62.5mm	0.14mm	0.22%
60cm	1m	64.5mm	63.2mm	0.12mm	0.19%
60cm	5m	64.5mm	64.2mm	0.13mm	0.20%
2m	2m	64.5mm	64.1mm	0.16mm	0.25%
2m	6m	64.5mm	64.3mm	0.16mm	0.25%

표에는 기재하지 않았지만, 실제 카메라와의 거리와 측정된 거리의 값은 표준편차로 약 2%의 오차율을 보였으며, 눈동자 사이 값의 변동 폭은 약 0.2%의 변화를 보이는 적은 오차를 가진다는 것을 알 수 있다.

오차율이 크진 않으나 이렇게 오차가 나는 이유는 여러 가지가 있다. 우선 카메라에서 원을 인식하는 부분에 대한 오차가 있다. 이 오차는 카메라 픽셀의 한계와, 원을 감지할 때 정확한 원을 감지하지 못해 생기는 오차로 사진을 찍을 때마다 측정값이 제각각 나오게 하는 오차이다.

오차 중 가장 큰 부분을 차지하는 오차는 두 대의 카메라의 방향이 같은 방향벡터 값을 갖지 않는 경우이다. 카메라의 방향이 서로 평행 되지 않아 생기는 오차는 생각보다 매우 큰 영향을 미친다. 이 오차는 측정값을 실제 값에 대하여 편향된 값을 나타내게 하는 오차이다.(측정값의 변동률(잡음 같은)은 이 원인과는 무방하다고 본다.)

또한 측정을 할 때 그 사람이 보는 초점거리도 중요한데 이는 사람이 사물을 볼 때 눈은 그 사물을 향해 몰리기 때문이다. 결과표를 보면 알 수 있듯이 보는 초점거리가 짧을수록 눈동자 사이 값은 작게 측정된다.



IV. 결론

3D-TV의 시청에 있어서 눈의 극심한 피로를 주게 되는 원인인 3D-TV에서 고정된 시청 거리와 고정된 사람의 눈동자 간격에 대한 화면 출력을 피하기 위해 본 논문의 방법을 이용하여 TV의 위쪽 양옆에 달아 측정하게 된다면, TV에 대한 보는 사람의 위치를 제공하며, 시청하는 사람의 눈동자 간격을 제공하여 3D-TV에서 그에 해당하는 화면 출력을 요청할 수 있도록 할 수 있기에 이 기술은 유용할 것이라 생각한다.

* 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2012R1A1A1038515)

참고문헌

[1] Tekalp, A. M., Smolic, A., Vetro, A., and Onural, L., Eds. Special issue on 3-D Media and Displays, vol. 99, 4. Proceedings of the IEEE. 2011.

[2] P. Viola and M. Jones, “"Rapid Object Detection using a Boosted Cascade of Simple Features”", in Proc. IEEE Int’1 Conf. on CVPR, 2001.

[3] Atherton, T.J., D.J. Kerbyson. "Size invariant circle detection." Image and Vision Computing. Volume 17, Number 11, 1999.