

---

## 관심 기반 양안수렴 입체 렌더링

### Saliency-Driven Convergent Stereoscopic Rendering

김영욱, Younguk Kim\*, 이성길, Sungkil Lee\*\*

---

**요약** 올바른 3D 입체 영상을 만들기 위해서는 양안시차와 양안수렴을 동시에 고려해야 한다. 그러나 보편적인 입체 렌더링에서는 양안수렴을 고려하기 위한 초점을 찾기 어렵기에 양안시차만을 이용하여 영상을 생성하므로 눈이 피로할 수 있다. 이에 본 논문은 양안수렴을 고려하기 위해 초점을 찾는 방법으로 계산적인 관심 추적을 이용하는 것을 제안한다. 관심추적 맵을 이용하여 초점이 될 수 있는 영역이나 물체를 찾는다. 이렇게 생성된 초점 거리를 이용하여 양안수렴이 되도록 투영행렬을 수정하여, 보다 효과적으로 3D 입체 영상을 렌더링 할 수 있다.

**Abstract** Proper 3D Stereoscopic rendering requires to consider both binocular disparity and convergence. However, it is generally difficult to find a focus in a scene, and thus, a typical rendering generates images solely using binocular disparity, which might result in visual fatigue. This paper presents a computational attention tracking for finding a focus to consider convergence. The attention map selects an object or area as a potential focus. We reflect the focal distance to adapt the projection matrices, and thus, are able to perform effective stereoscopic imagery.

**핵심어:** *Stereoscopic Rendering, Visual Attention, Saliency Map*

---

본 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 기초연구, 중견연구자, <실감교류 인체감응솔루션> 글로벌프론티어사업의 지원으로 수행되었음(2011-00014015, 2012R1A2A2A01045719, 2012M3A6A3055695).

\*주저자 : 성균관대학교 컴퓨터공학과 학사과정 e-mail: ywkim91@skku.edu

\*\*교신저자 : 성균관대학교 컴퓨터공학과 조교수 e-mail: sungkil@skku.edu

## 1. 서론

3차원 입체 렌더링은 3D TV나 영화의 보급과 함께 관심이 증가하고 있다. 3차원 시뮬레이션이나 디자인, 광고에서도 실제 보는 것과 같은 효과를 얻기 위해 필요성이 대두되고 있다.

3차원 입체 영상을 만들기 위해 좌안과 우안에서 바라보는 두 장의 이미지가 필요하다. 두 장의 이미지가 양안에서 동시에 인식됐을 때, 양안시차(binocular disparity)를 인지하여 깊이를 느낄 수 있다. 그리고 두 이미지를 3D 디스플레이와 3D 안경을 사용하여 좌안과 우안에 따로 인식시켜 입체영상을 볼 수 있다. 3D 안경 방식으로는 서터글래스나 편광, 적청 방식을 주로 사용하며, 3D 디스플레이에서는 좌우 각각에만 투과될 수 있는 두 종류의 영상을 출력한다. 이를 통해 좌안과 우안에 다른 영상을 보여주어 입체감을 느낄 수 있다.

이러한 3차원 입체 영상은 양안에 대해 보편적인 컴퓨터 그래픽스 렌더링을 사용해서 만들 수 있다. 이때, 위에서 언급한 것처럼 양안시차를 고려하여 렌더링을 해야 한다.

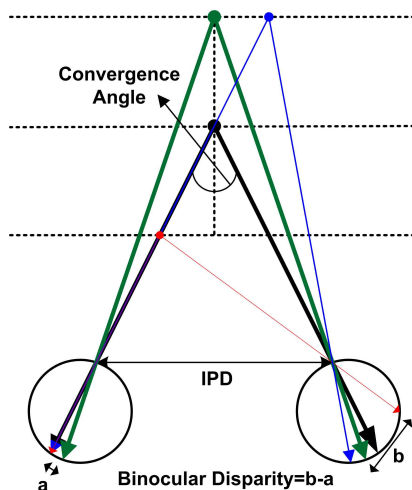


그림 1. 양안시차 및 양안수렴, IPD.

양안시차는 양안이 인식하는 물체의 위치 차이를 말한다. 그림 1에서 검은색 투사와 같이 동일한 물체가 양안의 초점상에 존재하면 망막의 같은 위치에 맺힌다. 그러나 빨간색 투사처럼 초점 외에 존재하게 되면 다른 위치에 맺힌다. 이 차이가 양안시차이고 이로 인해 깊이를 느낄 수 있다[1]. 그러나 입체 영상을 만들 때, 양안시차 뿐만 아니라 양안수렴(binocular convergence)도 함께 고려해야 한다.

양안수렴은 초점 깊이와 거리가 다른 물체를 초점에서 하나로 인식하기 위한 눈의 움직임을 말한다[1]. 그림 1에서 초점의 위치가 흑색에서 녹색 위치가 되면 양안수렴 정도가 달라진다. 그러나 실제 초점은 녹색이지만 흑색의 초점을 사용하여 렌더링을 하면 사람이 실제로 봤을 때의 양안수렴 정도와 렌더링을 통해 만들어진 양안수렴 정도가 일치하지

않게 된다. 이러한 불일치는 초점에서의 물체를 하나로 인식하지 못하게 하여, 입체영상을 볼 때 눈을 피로하게 만들고 어지러움을 유발한다. 이것은 3D TV나 입체 디스플레이의 대중화를 제한하는 하나의 문제점으로 잘 알려져 있다.

따라서 양안수렴을 고려하기 위해서는 사람이 보고 있는 위치인 초점(focus)을 찾아야 한다. 직접 3D 카메라를 사용하여 입체영상을 찍는 경우에는 초점을 찾기가 비교적 용이하다. 그러나 컴퓨터그래픽스로 입체영상을 만들 때는 미리 초점을 알려주어야 하는데, 복잡한 장면에서는 초점을 지정하기 쉽지 않다. 따라서 자동으로 초점을 찾는 방법을 사용하면 양안수렴을 적용하기가 용이해진다.

초점을 자동으로 찾아 양안수렴을 보정하기 위한 방법론으로, 우리는 계산적인 visual attention 추적을 사용하는 방법을 제안한다. visual attention을 추적하기 위해, 하향식(Top-Down) 맥락이 있어야 하지만 사용자의 의도를 알 수 없기 때문에 이는 사용하기 힘들다. 따라서 feature 기반의 상향식(Bottom-Up) 방법으로 saliency를 이용한다[4][5]. saliency map을 통해 이미지에서 salient 부분을 찾을 수 있고, 이로부터 초점이 될 수 있는 물체나 영역을 계산할 수 있다. 이렇게 계산된 초점거리를 이용하여 투영행렬의 양안수렴을 바르게 맞춰줄 수 있다.

## 2. 관련연구

입체 영상 이미지를 생성하는 방법에는 Toe-in 방법과 Off-axis view frustum을 사용하는 방법이 존재한다. Toe-in 방법은 양안이 초점을 바라보는 형태로, 동일한 초점거리(focal distance)에 있는 초점의 평면(plane of focus)이 일치하지 않고 양안이 서로 다른 투영평면을 갖는다. 그런데 이는 렌더링에서는 keystoneing이 발생할 수 있는 잘못된 접근 방법으로 알려져 있다 [2]. 보다 나은 방법인 Off-axis view frustum 방식은 양안이 평행한 축을 가지며, 단일 눈이 보고 있는 초점의 평면과 양안에 대한 초점의 평면이 일치한다. 따라서 양안에 대한 view frustum이 비대칭 형태를 가진다[2][3].

초점을 탐색하기 위한 방법으로 가장 직관적인 방법은 안구 추적 장치(eye-tracker)를 이용하는 것이다. 그러나 이러한 장치는 비싸고 착용하기에 불편하다. 다른 방법으로, 사람의 시각 인지 원리를 이용하여 관심이 될 영역이나 물체를 추정하는 것이다. 이는 앞서 말한 것처럼 상향식 방법으로, 계산적으로 초점을 찾는 방법이다[4][5][6].

## 3. 알고리즘

양안시차와 양안수렴을 모두 고려한 입체영상 이미지를 생성하기 위해, saliency map을 생성하는 과정과 saliency map에서 초점을 찾는 과정, 찾은 초점을 적용하여 이미지를 생성하는 세 과정이 필요하고 이를 기술한다.

### 3.1 Saliency Map 생성



그림 2. 생성된 Saliency Map.

saliency map을 생성하기 위해 가장 보편적인 [4]에서 제안한 방법론을 이용한다. 입력 RGB 이미지를 CIE Lab 공간으로 변환한 후,  $1 \times 1$  크기의 이미지가 될 때까지 4분의 1씩 이미지를 downsampling하여 이미지 피라미드를 만든다. 이미지 피라미드의 수준(level)이 blur의 정도를 조절하므로, 저수준의 집합을 center로 정의하고 고수준의 집합을 surround로 정의한다. 각 이미지 피라미드에서 center와 surround 수준이  $c \in \{2, 3, 4\}$ ,  $\delta \in \{3, 4\}$ ,  $s=c+\delta$  인 6개의 쌍을 정의한다. 모든 (c,s) 쌍에 대해서, coarser image를 finer image로 upscaling 하고 서로의 차를 계산하여 cross-scale image differences를 구하고, 이 이미지 차이를 feature map이라 한다. 본 논문에서는 [4]와 달리 이미지 feature로 CIE Lab 컬러공간에서의 색 거리를 이용하므로 feature map 6장의 선형 조합을 saliency map으로 정의할 수 있다. 그림 2는 위의 과정을 거쳐서 만든 saliency map으로, 밝은 색이 높은 saliency를 나타낸다.

### 3.2 초점 탐색

3.1에서 생성한 saliency map으로부터 초점을 찾으면, pixel 수준에서 탐색하므로 초점을 찾기가 어렵다. 이를 해결하기 위해 object 수준에서 초점을 찾을 수 있는데, item buffer를 쓰게 되면 object 분할 기준의 모호함으로 인해 일반적인 정의가 어려울 수 있다. 또한, object 간의 초점 이동에서 이산적인 이동이 일어나므로 탐색 과정에 부정적인 영향을 준다. 그래서 보다 일반적이고 연속적으로 초점을 이동하기 위해 saliency 값에 가중치를 주어서 saliency map을 만든다 (Algorithm 1 참조).

saliency map으로부터 가장 중요한 부분을 찾기 위해서, 가중치를 사용하여 이미지 피라미드를 만들어서(5-11줄) 마지막  $1 \times 1$  부분의 값을 취하면 초점의 위치를 알 수 있고, depth 값에서 초점거리를 계산할 수 있다(12-13줄). 이런 방법은 초점거리의 이산적인 이동 없는 부드러운 전이를 보장한다.

#### Algorithm 1 Focus Search

---

**Input:**  $S_0$ : Saliency map in level 0  
**Input:**  $I$ : Rendered image with depth

- 1: **for**  $(x, y) \in S_0$  **do** ▷ for all pixels
- 2:  $S_0(x, y) = S_0(x, y) \times (1 - I(x, y).Depth)$  ▷ weighting
- 3:  $S_0(x, y) = \text{Hermite\_Interpolation}(0, 1, S_0(x, y))$
- 4:  $M_0(x, y) = (x, y, I(x, y).Depth)$  ▷ create map
- 5: **for**  $L = 1$  to  $\text{MIP\_LEVEL}-1$  **do** ▷ for all levels
- 6: **for**  $(x, y) \in M_L$  **do** ▷ for all pixels
- 7:  $M_L(x, y) = \sum_{i=0}^1 \sum_{j=0}^1 M_{L-1}(2x+i, 2y+j) \times$
- 8:  $S_{L-1}(2x+i, 2y+j)$
- 9:  $S_L(x, y) = \sum_{i=0}^1 \sum_{j=0}^1 S_{L-1}(2x+i, 2y+j)$
- 10:  $M_L(x, y) = M_L(x, y) / S_L(x, y)$
- 11:  $S_L(x, y) = S_L(x, y) / 4$
- 12:  $\text{Focus} = M_{\text{MIP\_LEVEL}-1}$  ▷  $1 \times 1$  pixel
- 13:  $\text{Focal\_Distance} = \text{Focus.Depth} \times \text{Camera.Far}$

---

### 3.3 Stereoscopic Image 렌더링

입체영상 이미지는 좌안과 우안에 대해서 asymmetric view frustum을 적용하여 만들 수 있다. 양안이 초점의 평면으로 모이게 되는데, 이는 단일 눈에서의 초점거리에 존재한다. 이 초점의 평면을 기준으로 양안에 대해서 asymmetric view frustum을 적용한다.

asymmetric view frustum에서는 left와 right, top, bottom, front, back 값이 필요한데, 양안이 x축에 존재하므로 left와 right 값만 새로 고려해주면 된다[7]. 좌안과 우안의 동공 간의 거리를 나타내는 동공간 거리(interpupillary distance; IPD)  $I$  값과 초점거리  $D$  값을 고려하여 asymmetric view frustum의 left와 right 좌표가 이동한 오프셋  $O$ 를 식(1)과 같이 구할 수 있다. 이를 단일 눈에서의 symmetric view frustum  $L_S$ 와  $R_S(=-L_S)$  값에 적용하면 asymmetric view frustum을 위한 좌안의  $L_{L,A}$ 와  $R_{L,A}$  값(우안의  $L_{R,A}$ 와  $R_{R,A}$  값)을 구할 수 있다.

$$O = (((D - N) / D) \times (I / 2) - I / 2) \quad (1)$$

$$L_{L,A} = L_S - O, R_{L,A} = R_S - O \quad (2)$$

$$L_{R,A} = L_S + O, R_{R,A} = R_S + O \quad (3)$$

식(1)에서 초점거리  $D$  값은 3.2에서 계산한 초점거리와 같고,  $N$  값은 카메라의 near 값에 해당한다. 좌안의  $L_{L,A}$ 와  $R_{L,A}$  값(우안의  $L_{R,A}$ 와  $R_{R,A}$  값)을 이용하여 asymmetric view frustum을 위한 행렬을 계산하여 좌안과 우안에서의 렌더링의 투영행렬에 사용한다. 그리고 보편적인 컴퓨터그래픽스 렌더링 과정을 통해 입체영상 이미지 2장을 얻을 수 있다.

## 4. 결과

본 알고리즘은 Intel i7-3770, NVIDIA Quadro K5000,

DX10 API 환경에서 심도 렌더링과 함께 구현되었다. 그림 3은 품질 측정을 위해 사용된 모델로, 65,648 삼각형에 대해서 1920×1080 해상도로 53 FPS의 성능을 얻을 수 있었다. 그림 1의 위 이미지는 양안수렴이 고려되지 않은 입체영상 이미지를 얻은 것이다. 카메라의 초점거리 비율(투영평면부터 Far까지의 비율)은 0.133 (초록색 인형 위치)이지만 입체영상 이미지를 위한 임의로 고정한 초점거리는 0.228 (파란색 인형 위치)으로 실제 convergence가 더 크다. 따라서 초점거리에 있는 상들에서 양안시차를 느끼고 눈이 피로해진다.

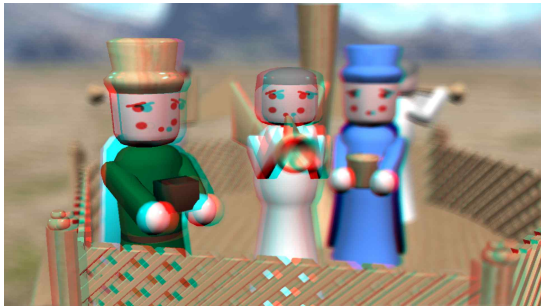


그림 3. 적형 애너글리프(red-cyan anaglyph) 입체영상 이미지 결과. 상: 카메라 초점거리(1671)와 입체영상 초점거리(3000) 비일치, 하: 카메라 초점거리(1671)와 입체영상 초점거리(1671) 일치.

그림 1의 아래 이미지는 양안수렴을 고려한 입체영상 이미지이다. 카메라의 초점거리와 입체영상 이미지를 위한 초점

거리는 동일하게 0.133이다. 따라서 초점거리에 위치한 상들에서 양안시차가 발생하지 않는다. 양안시차와 양안수렴이 모두 고려되어 실제 보는 것과 같은 입체영상 효과를 느끼게 되어 눈이 피로하지 않고 어지러움을 느끼지 않는다.

## 참고문헌

- [1] Stidwill, D. and Fletcher, R. Normal Binocular Vision: Theory, Investigation and Practical Aspects, Wiley, 2010. 11-13, 22, 140, 175.
- [2] Stegmaier, Simon, Dirc Rose, and Thomas Ertl. "A case study on the applications of a generic library for low-cost polychromatic passive stereo." Proceedings of the conference on Visualization'02, IEEE Computer Society, 2002.
- [3] Gateau, Samuel, and S. Nash. "Implementing stereoscopic 3D in your applications." GPU Technology Conference, 2010.
- [4] Itti, Laurent, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis." Pattern Analysis and Machine Intelligence, IEEE Transactions on 20,11 (1998): 1254-1259.
- [5] Lee, Sungkil, Gerard Joungyun Kim, and Seungmoon Choi. "Real-time tracking of visually attended objects in virtual environments and its application to lod." Visualization and Computer Graphics, IEEE Transactions on 15,1 (2009): 6-19.
- [6] Hillaire, Sébastien, et al. "Depth-of-field blur effects for first-person navigation in virtual environments." Computer Graphics and Applications, IEEE 28,6 (2008): 47-55.
- [7] Shreiner, D. and Sellers, G. and Kessenich, J.M. and Licea-Kane, B.M., OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3. Pearson Education, 2013. 227-229.