

입체 영상을 이용한 몰입형 3차원 게임 제작

정재훈⁰, 김재희, 박인규
 인하대학교 정보통신공학부
 {jjhman@empal.com}

Development of Immersive 3-D Game Using Stereoscopic Imaging

JaeHoon Jung⁰, Jaehee Kim, and In Kyu Park

School of Information and Communication Engineering, INHA University

요 약

본 논문에서는 입체 영상을 이용한 몰입형 3차원 게임 제작 기법을 제안한다. 제안하는 기법은 서터글래스를 이용하여 좌안과 우안의 영상을 분리 투영하여 사용자로 하여금 입체감을 느낄 수 있도록 한다. 이 때, 사용자가 느끼는 눈의 피로를 최소화하기 위하여 입체 영상 디스플레이에 필요한 인자들을 최적으로 선택하는 방법을 제안한다. 또한, 예시되는 블록격파게임에서의 충돌처리를 효율적으로 구현하기 위하여 본 논문에서는 체적소와 광선 추적법을 응용한 효율적인 알고리즘을 제안한다.

1. 서 론

다양한 3차원 콘텐츠가 등장하고 개발되는 가운데 사용자의 몰입감을 높이기 위한 입체 영상에 관한 관심이 높아져가고 있다. 특히 2차원 영상으로는 입체감을 느낄 수 없기 때문에 3차원 그래픽 콘텐츠의 입체감을 몰입감있게 표현하기 위한 다양한 시도가 이루어지고 있다.

그러나 현재 입체 영상 콘텐츠 제작에는 몇가지 문제점이 존재한다. 우선, 영상제공을 위한 콘텐츠 제작이 힘들다는 점과 입체 표현 장비가 고가라는 점이다. 이와 같은 문제점을 보완하기 위해 본 논문에서는 저가의 서터글래스를 이용하되 사용자의 입체감과 눈의 피로도를 최소화하는 기법과 게임에서의 실시간 3차원 영상 처리를 위하여 체적소와 광선 추적법을 응용한 효율적인 충돌 알고리즘을 제시한다. 또한 블록격파게임을 통하여 실제 응용 사례를 제시한다.

2. 3차원 입체 영상의 구현

2.1 입체 영상의 원리 및 문제점

같은 방향으로 향하는 두 개의 시선은 하나의 장면에 대하여 유사하지만 서로 상이한 다른 두 개의 상을 인지하게 된다. 이 때 인간의 두뇌에서 두 상이 정합되어 인간은 입체를 느낀다. 이러한 과정을 컴퓨터에서 구현하기 위하여 가상 카메라를 이용한 좌우 영상을 생성하여야 하는데, 하드웨어를 이용한 실제 구현에는 크게 나누어 순차주사(interlaced)와 페이지 뒤바꿈(page flipping)의 두 가지 방식이 있다. 그러나 순차주사 기법은 눈의 피로도가 높고 실시간 구현이 힘들기 때문에 본 논문에서는 페이지 뒤바꿈 기법을 활용한다 [1].

페이지 뒤바꿈을 구현하기 위하여 본 논문에서는 서터 글래스를 활용한다. 가상 카메라로부터 취득된 좌측 영상과 우측 영상을 모니터의 수직주파수에 동기시켜 서터글래스에 보내주면, 사용자의 왼쪽 눈은 항상 좌측 영상을 관측하게 되고 오른쪽 눈은 항상 우측 영상을 관측하게 된다. 이러한 변환 과정이 1초에 60번 이상 고속으로 이루어지면 사용자는 마치 연속된 입체 영상을 관측하는 것과 같은 몰입감을 느끼게 되는 것이다.

3차원 입체 영상의 원리는 이와 같이 간단하지만, 대부분의

경우 사용자가 느끼는 눈의 피로도가 매우 심하다. 따라서 사용 시간이 증가할수록 사용자는 어지러움을 느끼게 되어 장기간 사용이 매우 힘든 실정이다 [2].

2.2 제안하는 입체 영상 구현 기법

2.2.1 시차(Parallax)

많이 이용되는 시차에는 평행시차(Parallel parallax)와 영시차(Zero parallax)가 있다. 이중 영시차가 가장 일반적으로 사용되는데, 이것은 인간이 가까운 사물을 관측할 때 그 사물에 초점을 맞추도록 적용하는 시차이다. 그러므로 게임 제작에 유용하게 사용할 수 있다.

2.2.2 입체 영상에서 3차원 물체의 깊이와 피로도 관계

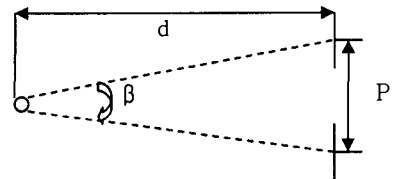


그림 1. 영시차에서의 양안과 사물의 각.

그림 1과 같이 눈 사이의 거리, 즉 내안각간 거리(eye separation)가 P이고 눈이 위치하는 영상 평면에서 관측하는 물체까지의 거리를 d라고 하자. 이 때, 기존의 의학에서의 연구에 따르면 사물을 관측할 때 시선의 수렴 각도 β 값이 1.5도가 넘으면 눈이 피로를 느낀다는 사실이 알려져 있다 [3]. 그림 1의 관계에서, 수렴 각도를 P와 d로 표현하면 식 (1)과 같다.

$$\beta = 2 \arctan \frac{P}{2d} \quad (1)$$

눈에서 가까운 위치에 있는 물체를 초점을 맞추어 응시하

게 되면 β 값이 커지므로 눈이 피로해지고, 눈에서 매우 멀리 있는 물체를 관측하는 경우 β 값이 0에 가까워지므로 눈이 피로를 느끼지 못하게 된다.

따라서 입체 영상 구현에서 사용자의 피로도를 줄이기 위해, 그래픽 가상 환경에서 물체의 위치에 따라 내안각간 거리 P와 시선 수렴 각도 β 를 적응적으로 정확히 설정해야 한다.

2.2.3 제안하는 입체 영상 생성 기법

입체 영상의 피로도를 줄이고 입체 영상의 깊이를 개인에 맞게 적절히 조절하기 위해 물체의 위치에 따라 내안각간 거리 P와 시선 수렴 각도 β 를 적응적으로 설정 가능하도록 한다. 이를 위해 제안하는 기법은 다음과 같다. 시점(Look at point)을 미리 지정하고 그래픽 장면을 구성하는 물체 중 주요한 관심 객체들의 깊이값(Zmin)을 차감하여 식 (1)로부터 식(2)와 같이 Pmax값을 계산한다. 이는 주어진 3차원 객체에 대해 최악조건분석(worst case analysis)을 수행하는 것과 같은데, 그림 2와 같이 Pmax값을 실시간으로 나타내고 최악조건분석 수행을 통해 설정을 조절한다.

블록격파게임을 예로들면 게임 시작시 Zmin은 블록의 첫번째 블록층이며 첫번째 블록층이 모두 격파된 후에는 두번째 층이 Zmin으로 변환 계산되어야 한다.

$$P_{max} = 2 * (\text{Lookpoint} - Zmin) \tan \frac{1.5^\circ}{2} \quad (2)$$

또한 최대 내안각간거리의 80%인 거리를 기본설정으로 두고 좀 더 세밀한 설정을 원하는 사용자를 위하여 개인에 맞는 최적의 값을 찾을 수 있도록 그림2과 같이 게임상에서 범위 조절이 가능하게 한다.

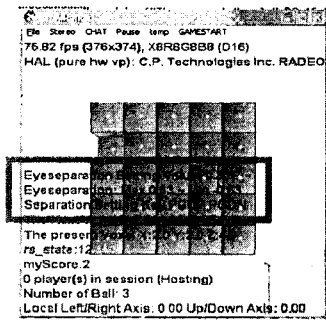


그림 2. 내안각간거리의 한계범위 조절.

그림 2와 같이 최대, 최소값을 기준으로 사용자가 자신의 눈에 맞는 내안각간 거리 값을 조정할 수 있는 인터페이스를 제공하여, 사용자가 눈의 피로를 최소화하고 몰입감을 최대로 높일 수 있는 상태를 찾도록 한다.

2.3 제안하는 입체 영상의 실험결과

제안하는 알고리즘을 검증하기 위해 주관적테스트(subjective test)를 수행하였다. 즉, 피실험자 20명을 대상으로 내안각간 거리와 피로도의 상하 관계에 대한 실험을 수행한 결과, 피실험자의 80%가 β 값이 1.5도가 넘는 내안각간 거리 P값에서 어지러움을 호소하였다. 또한, 나머지 20%의 피실험자도 최대 내안각간 거리에서 크게 벗어나지 않는 한도내에서 어지러움증을 호소하였다. 제안하는 기법에 의해 최대 내안각간 거리를 제한할 경우, 입체 영상의 품질에 대해 피실험자의 60%가 매우 만족하였고 40%가 대체로 만족하였다.

실험수행 결과, β 값이 1.5도가 넘지 않는 범위 내에서 입

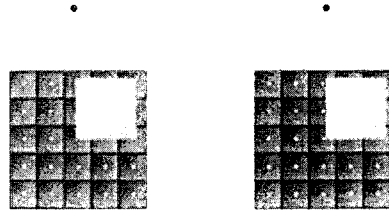


그림 3. 입체 영상의 좌우 쌍(stereo pair).

체영상 콘텐츠를 제작한다면 눈의 피로나 어지러움을 방지할 수 있으며 대체적으로 만족할만한 입체 영상의 품질을 구현할 수 있음을 알 수 있다.

그림 3에 구현된 입체 영상의 좌우 영상의 예를 도시하였다.

3. 몰입형 3차원 게임 제작

제안하는 입체 영상 생성 기법을 실제 응용 사례에서 실증하기 위해, 본 논문에서는 3차원 블록격파게임을 제작하여 입체 영상으로 구현하고 이를 서터글래스를 착용하고 관측하는 몰입형 3차원 게임을 제작하였다.

3.1 3차원 블록격파게임

제작된 게임은 2차원 블록격파게임을 3차원으로 확장한 게임이다. 이 게임은 벽면, 블록, 플레이트, 공과 같은 간단한 개체들로 이루어져 있어 입체 영상의 깊이가 분명하게 나타난다. 공간 분할의 기준이 분명하므로 본 논문에서 제시하는 알고리즘을 적용하여 블록과 블록의 위치를 모델링하였다.

3.2 제안하는 충돌알고리즘기법

3.2.1 충돌처리 알고리즘의 필요성

입체 영상은 좌안과 우안을 번갈아 초당 60프레임 이상을 실시간 렌더링해야 하므로 많은 양의 CPU와 그래픽 자원을 사용한다. 따라서 복잡한 연산이 필요한 게임의 경우, 실시간 렌더링을 가능하게 하기 위하여, 블록격파게임의 대부분의 계산량을 차지하는 충돌처리 알고리즘의 고속화가 필수적이다. 본 논문에서는 3차원 광선추적법 [4]을 응용한 충돌처리 알고리즘을 제안한다.

3.2.2 제안하는 충돌처리 알고리즘

3.2.2.1 공간분할 및 공의 이동

게임에 이용되는 오브젝트의 크기에 따라 그림 4과 같이 복셀(voxel)의 크기를 적절히 정하여 공간을 분할한다.

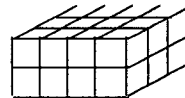


그림 4. VOXEL 공간분할.

전처리 및 후처리과정에서는 블록격파게임의 공이 복셀들로 이루어진 공간을 진행해나갈 때, 공이 각 복셀의 어느 면을 통과해 나가는가를 결정하게 된다.

3.2.2.2 공의 이동 복셀 결정

본 과정에서는 복셀의 각 면의 법선 벡터들과 공의 이동 방향 벡터와의 내적 연산을 통해 내적값이 가장 1에 가까운

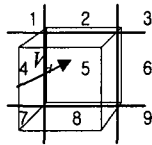


그림 5. 후처리 과정.

법선 벡터를 가지는 면을 구한다. 즉, 공이 뚫고 지나갈 확률이 가장 높은 면을 선택한다. 이 과정에서 구한 면을 확장하여 영역경계를 정하여 V_d (오브젝트의 진행방향)와 면의 교점을 구하여 교점의 위치를 통해 다음 복셀의 위치를 판정한다.

예를 들어 그림 5와 같이 현재복셀의 위치는 (x,y,z) 이고 전처리과정에서 구한 면은 N_{for} 를 법선벡터로 하는면이라 가정하면, V_d 와 N_{for} 를 법선벡터로 하는 면과의 교점의 위치 영역에 따라 다음복셀을 결정한다.

- 가) 교점의 위치영역이 2일경우 -> 다음복셀은 $(x,y+1,z)$
 - 나) 교점의 위치영역이 4일경우 -> 다음복셀은 $(x-1,y,z)$
 - 다) 교점의 위치영역이 5일경우 -> 다음복셀은 $(x,y,z+1)$
 - 라) 교점의 위치영역이 6일경우 -> 다음복셀은 $(x+1,y,z)$
 - 마) 교점의 위치영역이 8일경우 -> 다음복셀은 $(x,y-1,z)$
- 만약, 교점의 위치영역이 1,3일 경우 전처리 연산 결과를 N_{top} 으로 설정하고 다시 후처리를 수행한다. 또한, 교점의 위치 영역이 7,9일 경우 전처리 연산 결과를 N_{bottom} 으로 설정하고 다시 후처리 수행한다.

3.2.2.4 충돌 처리

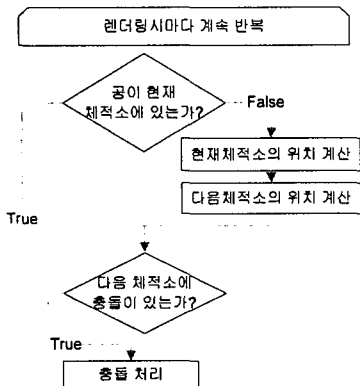


그림 6. 충돌수행과정.

3.2.2.5 제안하는 알고리즘의 특징

일반적으로 충돌 알고리즘은 계산량을 줄이기 위하여 서로 충돌하는 물체의 방향 벡터의 내적이 0보다 작은 경우만 충돌을 수행한다.

그러나, 그림 7과 같은 경우는 내적이 모두 0보다 작으므로 계산량을 줄일 수 없다. 따라서, 3차원 블록격자게임과 같은 게임에서의 충돌검출은 일반적인 충돌 알고리즘을 이용하면 비효율적이다. 하지만 제안하는 알고리즘의 경우 다음



그림 7. 블록과 공의 충돌.

복셀을 찾는과정, 즉 후처리과정에서 단 한번의 충돌 탐색을 하므로 매우 효율적이다. 또한 움직임은 물체 사이의 충돌의 경우도 일반적인 알고리즘의 경우 nC_2 의 충돌탐색이 필요하지만 제안하는 알고리즘의 경우 각각의 움직이는 물체가 후처리과정을 하는 경우에만 탐색하므로 총 n 번 수행하게 된다. 이와 같이 제안하는 충돌처리를 이용하여 게임의 계산량을 줄이고 3차원 입체 영상의 실시간 렌더링에 필요한 프레임수를 확보할 수 있다.

한편, 제안하는 알고리즘의 단점은 다음과 같다.

- 가) 체적소영역에 대한 정보를 담은 메모리를 확보해야 한다.
- 나) 동체 물체에 따라 체적소 크기를 결정하기 힘들다.
- 다) 셋째 물체가 체적소를 이동할 때 위치 판정의 정확도가 게임에 영향을 미칠 수 있다

다음으로 제안하는 알고리즘과 기존의 알고리즘의 계산량을 비교해본다. 우선, 물체의 이동은 프레임당 0.2만큼 움직이며 복셀의 크기는 $2x2x2$ 이고 1초에 30프레임을 렌더링한다고 가정하자 이 경우 두 알고리즘의 계산량을 비교해보면 다음과 같다.

블록의 수가 n 이라고 할 때, 이동물체와 정지물체 사이의 충돌은 기존의 알고리즘의 경우 초당 $30*n$ 번 충돌 처리를 수행하지만 제안하는 알고리즘의 경우 초당 3번 $((0.2*30)/2)$ 의 충돌 처리를 수행한다. 또한, 움직이는 물체의 총 수가 n 이라 할때, 움직이는 물체사이의 충돌의 경우 기존의 알고리즘은 $30*nC_2$ 번 충돌 처리를 수행하지만 제안하는 알고리즘의 경우 3번 $((0.2*30)/2)*n$ 의 충돌 처리를 수행한다.

4. 결론

입체 영상게임을 제작하기 위한 핵심 요소기술은 입체 영상의 최적화와 게임의 핵심이 되는 알고리즘의 최적화이다. 본 논문에서는 입체 영상 게임을 제작하는 데 있어 눈의 피로도는 낮추면서 고속의 렌더링이 가능한 알고리즘과, 사용자 인터페이스를 제시하였다.

향후 연구방향으로는 물체를 관찰하는 depth-of-field를 고려하여 입체 영상의 매개 인자를 더욱 최적화하여 눈의 피로를 더욱 줄이는 기법에 관하여 연구할 예정이다.

5. 참고문헌

- [1] Stereographics Corp. Developer Handbook 1997
- [2] L. Lipton, "Factors affecting ghosting in a time-multiplexed plano-stereoscopic CRT display system," SPIE Proceedings, vol.761,1987
- [3] L. Lipton, et al, *Stereoscopic Television System*. U.S. Patent No.4,523,226, June 11, 1985.
- [4] D. Hearn and M. Baker, *Computer graphics with OpenGL*, 3rd Edition, 2004.