

근접한 물체 사이의 공간 관찰을 위한 보조 카메라 위치 최적화

이지혜^O 한윤하 최명걸*

가톨릭대학교 미디어기술콘텐츠학과

jihye032@cg.catholic.ac.kr yghan38@cg.catholic.ac.kr mgchoi@catholic.ac.kr

Secondary Camera Position Optimization for Observing the Close Space between Objects

Ji Hye Lee^O Yun Ha Han Myung Geol Choi*

Dept. of media Technology and Media Contents, The Catholic University of Korea

요 약

본 논문에서는 3차원 가상 공간을 탐험하는 사용자가 가상 물체들 사이의 충돌 위험 영역을 정밀하게 관찰할 수 있도록 돕는 보조 카메라 최적화 기술을 소개한다. 보조 카메라의 역할은 두 물체 사이의 충돌 위험 영역을 자동으로 감지하고 해당 영역을 현재 메인 카메라가 보여줄 수 없는 새로운 시점에서 관찰하여 사용자에게 보여주는 것이다. 하지만 가상 물체의 기하학적 모양이 복잡한 경우 충돌 위험 주변 공간도 좁고 복잡해진다. 그 결과, 보조 카메라의 위치 설정에 많은 제약이 따르고 최적화 계산의 복잡도 역시 높아진다. 본 논문에서는 보조 카메라의 위치를 두 물체 사이에 계산된 바이섹터-서피스 상의 점으로 제한함으로써 공간적 제약 조건을 지킴과 동시에 최적화의 효율성을 높이는 방법을 제시한다. 검증을 위해 사용자가 핸드 모션 인식 장치를 이용해 3차원 가상 환경을 탐험하게 하는 프로그램을 제작하였으며 사용자 조사를 통해 효용성을 확인하였다.

Abstract

We present a secondary camera optimization method that helps the user exploring 3D virtual environment to precisely observe possible collisions between objects. The first role of our secondary camera is to automatically detect the area with the greatest possible collision. The second role is to show the detected area from a new angle of view that the current main camera cannot show. However, as the shapes of target objects are complex, the shape of the empty space between objects is also complex and narrow. It means that the space for the secondary camera position is highly constrained and its optimization can be very difficult. To avoid this difficulty and increase the efficiency of the optimization, we first compute a bisector surface between two target objects. Then, we limit the domain of the secondary camera's position on the bisector surface in the optimization process. To verify the utility of our method, we built a demonstration program in which the user can explore in a 3D virtual world and interact with objects by using a hand motion recognition device and conducted a user study

키워드: 카메라 위치 최적화, 바이섹터-서피스, 가상 환경, 사용자 인터페이스

Keywords: Camera position optimization, Bisector-Surface, virtual environment, User interface

*corresponding author: Myung Geol Choi/The Catholic University of Korea(mgchoi@catholic.ac.kr)

Received : 2018.06.23./ Review completed : 1st 2018.06.29. / Accepted : 2018.07.04.

DOI : 10.15701/kgcs.2018.24.3.33

ISSN : 1975-7883(Print)/2383-529X(Online)

1. 서론

컴퓨터 그래픽스로 구현된 가상환경을 이용한 작업의 장점 중 하나는 환경을 조명하는 카메라의 위치와 시점을 물리적 제약 없이 자유롭게 조정할 수 있다는 것이다. 예를 들어 사용자가 서로 다른 두 개의 가상 물체의 위치를 조정하여 밀접하게 배치할 때, 복수의 카메라를 통해 서로 다른 시점에서 관찰하면 물체 사이의 충돌을 쉽게 감지할 수 있고 정확도를 높일 수 있다.

가장 널리 사용되는 다중 카메라 인터페이스는 평면도, 정면도, 측면도를 원거리에서 조망하는 직교-다중화면(orthographic multi-view) 인터페이스이다. 직교-다중화면에서는 물체의 내부나 사각지대를 표시하기 위해 물체를 와이어프레임으로 렌더링 하는 것이 일반적이다. 이 방법은 사용자가 정면과 측면을 명확하게 정의할 수 있는 물체들(예: 집, 자동차)을 서로 근접하게 배치하고자 할 때 매우 유용하다. 하지만 물체의 기하학적 모양이 복잡하거나 두 물체가 서로 복잡한 형태로 얹혀 있는 경우에는 직교-다중화면이 유용하지 않다. 왜냐하면 관찰 대상의 정면이나 측면을 정의하기 어렵기 때문이다. 그리고 두 폴리곤 사이의 멀고 가까운 정도의 차이도 알 수 없으며, 특정 영역이 물체의 내부인지 외부인지도 판단하기 어렵기 때문이다.

본 논문에서는 3 차원 가상 공간을 탐험하는 사용자가 서로 근접한 물체들 사이의 충돌 위험을 정밀하게 관찰할 수 있도록 돕는 보조 카메라 위치 최적화 기술을 소개한다. 본 논문에서 정의하는 보조 카메라의 역할은 두 물체 사이의 충돌 위험 영역을 자동으로 감지하고 해당 영역을 현재 메인 카메라가 보여줄 수 없는 새로운 시점에서 관찰하여 사용자에게 보여주는 것이다. 와이어프레임과 같이 물체의 표면을 투시하는 방법은 지양하는 대신, 물체와 물체 사이의 빈 공간에 보조 카메라를 위치시켜 충돌 위험 영역을 관찰할 수 있도록 하였다. 문제는 물체 사이 공간의 형태가 복잡해지면 보조 카메라의 위치 설정에 많은 제약이 따르게 된다. 그 결과, 최적화 계산의 복잡도가 급격히 높아진다는 것이다. 본 연구에서는 이 문제를 피하기 위해 먼저 관찰 대상인 두 물체 사이를 가르는 바이섹터-서피스(bisector surface)를 계산한다. 다음 보조 카메라의 위치를 바이섹터-서피스 표면상의 점으로 제한함으로써 공간적 제약 조건을 쉽게 만족하게 하고 동시에 최적화의 효율성을 높인다.

제시하는 방법을 검증하기 위해 사용자가 핸드 모션 인식 장치를 이용해 3 차원 가상 환경을 탐험하게 하는 프로그램을 제작하였으며, 사용자 조사를 통해 효용성을 확인하였다.

2. 관련 연구

컴퓨터 그래픽스 분야에서 물체 사이의 관계나 애니메이션이 진행되는 모습을 효과적으로 보여주기 위한 연구가 활발하게 진행되고 있다. Yeh IC[1]은 다중 상호 작용 애니메이션에서 사용자가 지정하거나 시스템이 제안하는 흥미로운 사건들을 보기 위한 카메라 동작을 자동으로 계획하는 방법을 제안했다. Bill Tomlinson[2]은 3 차원 사람 형태의 캐릭터 애니메이션에서 캐릭터의 동작과 감정을 효율적으로 보여줄 수 있는 카메라의 위치와 조명을 설정하는 방법을 제안한다. 카메라의 위치를 이용하여 캐릭터의 동작을 효율적으로 보여주고, 조명을 이용하여 캐릭터의 감정을 강조하는 방법을 사용하였다. Li-wei He[3]은 버추얼 시네마토그래퍼(virtual cinematographer)라고 불리는 가상의 3 차원 환경에서 카메라를 자동으로 조정하는 패러다임을 제시했다. 가상환경에서 이벤트가 일어나고 있는 상황을 보여주면서 다음 장면이 효과적으로 전개되도록 카메라를 배치·이동하는 역할까지 담당한다. Christophe Lino[4]는 타릭(Toric) 공간을 제시하여 자동 시점의 효율적인 계산을 위해 간격 기반 검색 기법을 제안한다. 이 공간 안에서 캐릭터의 시점이 어떻게 변하는지 살펴보고, 시각적 속성이 변하는 것에 따라 직관적이고 실시간으로 제어하여 연속된 장면을 만들어 낸다. 이 외에도 3 차원 가상환경에서 일어나는 애니메이션을 효율적으로 보여주기 위해 카메라나 뷰-포트로 조작 혹은 예측하는 알고리즘이 많이 제시되었다[5][6][7][8]. 앞선 연구들은 3 차원 애니메이션에서 장면의 시점 변화를 위주로 연구되어 왔다. 이는 전반적인 애니메이션에서 주요한 이벤트 변화를 확인할 수 있다. 하지만 캐릭터가 책을 드는 동작 같은 정밀한 작업을 요구하는 동작의 경우, 두 물체가 겹쳐지거나 떨어져 있는지 등에 대한 세부적인 정보는 일반적인 메인 카메라에서는 확인하기 어렵다.

바이섹터-서피스(bisector surface)는 두 공간 혹은 물체를 이등분하는 표면이다. 이 표면을 계산하는 방법은 다양하게 제안되고 있다[9][10][11]. 그리고 컴퓨터 그래픽스 분야에서도 바이섹터-서피스를 응용한 방법들이 활발히 제시되고 있다. Xi Xhao[12]는 바이섹터-서피스를 이용하여 동적인 캐릭터와 물체 사이의 상호작용을 분류하고 검색하는 방법을 제안한다. Ruizhen[13]은 객체 카테고리에 대한 기능 모델을 학습하는 분석 방법을 제안한다. 기능에 따라 분류된 객체 카테고리에 대해 학습된 기능 모델을 지식 기반으로 제공하여 어떠한 장면이 주어졌을 때, 3 차원 객체를 이해할 수 있다. Paul[14]는 영역 사이의 관계에 기반한 릴레이션 확장 이미지 디스크립터 RAID(Relation-augmented Image Descriptor)를 제안하였다. 이 방법은 스케치 검색 기반을 사용하여 최소한의 학습으로

주어진 이미지 데이터베이스에서 중요한 이미지를 성공적으로 추출해 낸다. 이 외에도 바이섹터-서피스를 이용하여 객체 혹은 공간을 분리하여 분류 검색 등에 응용하는 기술은 다양하게 제안되고 있다[15][16]. 본 논문에서도 두 물체 사이의 충돌 위험을 파악하기 위해 바이섹터-서피스를 이용하였다. 다양한 방법으로 바이섹터-서피스를 계산하는 방법이 제안되었지만, 본 논문에서는 실시간으로 효율적인 계산을 하기 위해 Xi Zhao[17]이 제안한 바이섹터-서피스를 구하는 방법을 이용하였다. 이에 대한 설명은 4.1 절에서 확인할 수 있다.

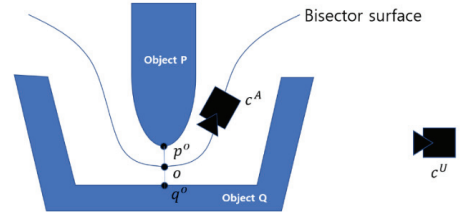


Figure 1 The position of our secondary camera (c^A) and bisector-surface between two objects P and Q.

3. 관찰 위치 최적화

두 물체 사이에서 충돌 위험이 가장 높은 영역을 관찰한다. 충돌 위험이 높은 곳은 두 물체의 거리가 가장 가까운 영역이다. **Figure 1** 에서 예시처럼 두 물체 P 와 Q 가 주어졌을 때, 표면 사이의 거리가 가장 가까운 공간의 가운데 지점 O 가 보조 카메라가 관찰할 위치가 된다. 계산의 효율성을 높이기 위해서 물체의 표면 전체를 살펴보는 대신, 각 물체가 메시 모델로 표현되었다고 가정하고 꼭지점의 위치만 고려하였다. 보조 카메라의 관찰 위치 O 는 아래와 같이 계산된다.

$$(p^O, q^O) = \underset{p_i \in P, q_i \in Q}{\operatorname{argmin}} \|p_i - q_i\|$$

$$O = \frac{p^O + q^O}{2}$$

p_i 와 q_j 는 각각 두 물체 P 와 Q 의 i 번째와 j 번째 꼭지점을 나타낸다. p^O 와 q^O 는 두 물체 표면의 거리가 가장 가까운 직선의 양 끝점으로 각각 첫 번째 물체의 표면과 두 번째 물체의 표면에 위치한다(**Figure 1** 참조). 점 O 는 p^O 와 q^O 사이의 중점이다.

이와 같이 물체 표면 전체를 보는 대신 꼭지점만을 대상으로 계산하면 복잡한 최적화 계산을 피할 수 있다는 장점이 있다. 반면 꼭지점 사이의 거리가 먼 경우에 정확도가 떨어진다. 이러한 부작용을 최소화 위해 필요한 경우 메시 모델의 표면상에 꼭지점을 추가로 샘플링하여 꼭지점 사이의 거리가 최대 약 1cm 가 넘지 않도록 하였다.

4. 카메라 위치 최적화

4절에서는 O 를 관찰할 수 있는 보조 카메라의 위치를 찾는다. 보조 카메라 위치를 위한 제약조건은 다음과 같다. 첫째, 보조

카메라의 위치가 물체 내부에 있어서는 안된다. 둘째, 보조 카메라의 위치에서 세 점 p^O, q^O, O 을 모두 관찰할 수 있어야 한다. 셋째, 보조 카메라는 관찰해야 할 충돌 위험이 있는 위치 O 와 가능한데 가까워야 한다. 다시 말해서, 보조 카메라는 항상 물체 밖에 위치하면서 두 물체 사이의 충돌 위험을 파악할 수 있어야 한다. 관찰 영역과 카메라 사이의 거리가 멀다면 근접하게 배치되어 있는 두 물체 사이의 충돌 위험을 관찰하기 어렵다. 따라서 보조 카메라가 점 O 와 근접하면서 두 물체 사이의 점 p^O, q^O, O 을 모두 관찰할 수 있어야 한다. 본 논문에서는 이러한 조건들을 만족시키기 위해 두 물체 사이에 존재하는 Xi Zhao[17]가 제안한 바이섹터-서피스의 메시 모델을 이용한다. 바이섹터-서피스 메시 모델을 이용하여 두 물체가 겹치는지 확인하고, 두 물체가 겹치지 않은 상태에서 메시 모델 위에 보조 카메라가 위치할 수 있는지 확인한다.

4.1 바이섹터-서피스

본 논문에서 필요로 하는 바이섹터-서피스는 3 차원 공간에서 서로 다른 두 물체 P 와 Q 가 있을 때, 이 둘을 각각 다른 영역에 포함되도록 나누는 2 차원 매니폴드(2D manifold)이다. 바이섹터-서피스의 모양은 가능하면 두 물체 사이 공간의 한 가운데를 지나도록 만드는 것이 좋다. 향후 보조 카메라의 위치를 바이섹터-서피스 위에서만 위치할 수 있도록 제약조건을 준다. 이 제약조건을 이용하여 카메라의 몸체가 물체의 몸체 속으로 들어갈 수 있는 가능성을 차단할 것이다(4.3 절 참고).

일반적으로 연속적인 표면을 가지는 정확한 바이섹터-서피스를 계산하는 것은 어려운 문제로 여겨지고 있다. 본 연구에서는 계산을 줄이기 위해 Xi Zhao[17]가 제안한 방법을 사용하였다. **Figure 2** 는 이 방법을 2차원 모델로 추상화하여 표현한 것이다. 먼저 두 물체 P와 Q가 메시 모델이라고 가정하고, 두 메시 모델의 모든 꼭지점을 대상으로 드로네 삼각분할(Delaunay triangulation)을 적용한다(그림 **Figure2-(a)** 참조).

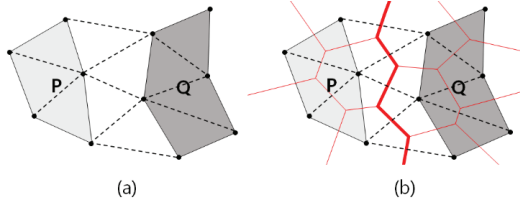


Figure 2 (a) represents the Delaunay triangulation result with respect to the vertices of P and Q. (b) represents the Voronoi diagram of (a). The red thicker line represents the bisector-surface between P and Q.

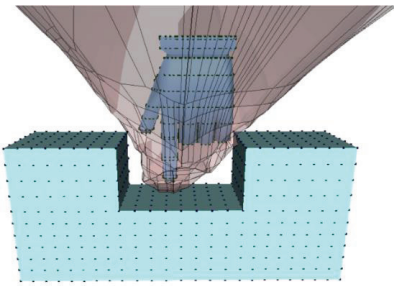


Figure 3 Bisector-surface between two objects. Vertices were sampled at a regular distance.

다음, 드로네 삼각분할 결과를 이용해 보로노이 다이어그램 (Voronoi diagram)을 만든다(그림 **Figure2-(b)** 참조). 마지막으로 보로노이 다이어그램의 선분 중에서 P와 Q 사이에 연결된 드로네 삼각분할 선분을 교차하는 선분들을 모두 선택하여 하나의 선으로 연결한다. 이렇게 만들어진 선 (또는 3차원 공간에서는 2차원 매니폴드)이 바이섹터-서피스가 된다.

Figure2-(b)에 빨간색 굵은 선으로 표시된 부분이 P와 Q 사이의 바이섹터-서피스이다. 이때 바이섹터-서피스는 무한하게 뻗어 나가며 생성된다. 따라서 마지막으로 바이섹터-서피스의 경계 범위를 정해주어 해당 범위 내에서만 생성이 되도록 한다.

이와 같은 방법으로 계산한 바이섹터-서피스의 모양은 비연속적이며 메시 모델의 꼭지점 분포에 따라 모양이 변한다는 단점이 있다. 특히 하나의 메시 모델 상에 있는 꼭지점들 사이의 거리가 지나치게 먼 경우 바이섹터-서피스 모양의 정확도가 떨어진다. 경우에 따라서는 바이섹터-서피스의 일부가 물체의 표면을 통과하는 부작용이 발생할 수 있다. 이 문제를 피하기 위해 3절에서 설명된 것과 같이 메시 모델의 표면에 꼭지점을 추가로 샘플링 하여 꼭지점 사이의 거리가 최대 약 1cm가 넘지 않도록 하였다.

Figure 3은 실제 3차원 메시 모델을 사용하여 계산한 바이섹터-서피스의 예를 나타낸다. 두 물체의 표면 위에 표시된

검은 점은 일정한 간격으로 샘플링 된 꼭지점을 나타낸다. 특히 하늘색 물체는 굴곡이 없는 평면이 많은 폴리곤으로 이루어져 있음에도 불구하고 많은 점이 샘플링 되어 있다. 반투명한 분홍색으로 렌더링 된 부분은 바이섹터-서피스의 메시 모델을 나타낸다.

4.2 카메라 위치 후보 탐색

보조 카메라의 위치를 찾기 전에 계산의 효율을 높이기 위해 보조 카메라의 위치가 될 수 있는 후보들을 먼저 탐색한다. 보조 카메라 위치는 제약 조건을 만족시키기 위해 바이섹터-서피스 메시 모델의 위에 있는 점으로 한정하였다. 계산에 이용된 바이섹터-서피스 메시 모델은 **Figure 3**과 같이 물체 표면의 점이 충분히 샘플링 되어, 두 물체에 겹치지 않으며 완곡한 형태를 보이고 있다.

보조 카메라의 위치 후보 집합을 찾기 위해 본 논문에서는 보조 카메라의 위치 c^4 와 관찰 위치 O 를 직선으로 연결한 선을 이용하였다. 이렇게 생성한 선이 다른 물체와 부딪히는 지점이 발생하면 어떤 물체에 의해 시야가 가려지는 곳이므로 보조 카메라 위치 후보에서 제외한다. 여기서 보조 카메라의 위치부터 관찰 위치까지 연결한 직선으로는 보조 카메라의 시야를 충분히 확보하기 어렵다. 그러므로 최소한의 시야를 확보하기 위해 직선이 아닌 두 물체 사이의 최소 거리인 p^o 와 q^o 사이의 거리를 지름으로 하는 원기둥을 이용한다. 생성한 원기둥 내에 다른 물체가 겹쳐지는지 여부를 판단하여 바이섹터-서피스 메시 모델의 점들 중에서 물체에 의해 시야가 가려지지 않는 점을 추려낸다. 이렇게 추려진 점들은 보조 카메라 위치 후보의 집합 \mathbb{C} 로 정의한다.

4.3. 카메라 위치 최적화

4.2절에서 구한 보조 카메라 위치 후보 집합 중에서 임의의 위치에 보조 카메라를 위치시켰을 때, 보조 카메라의 시야에서 세 점 p^o , q^o , O 를 모두 확인할 수 있다. 그러나 바이섹터-서피스 메시 모델의 형태에 따라 보조 카메라가 비추는 화면이 메인 카메라가 비추는 화면과 비교했을 때 효율적인 화면을 보여주지 않는 경우가 발생한다. 예를 들어 보조 카메라 위치로 바이섹터-서피스 위에 있는 임의의 점을 선정하였다. 이 점이 다른 점들의 간격에 비해 먼 거리에 위치하면 보조 카메라가 점 O 에서 발생하는 충돌 위험을 관찰하기 어렵다. 또한, 선택된 점에 보조 카메라가 비추는 화면이 메인 카메라가 비추는 화면과 큰 차이가 없는 경우도 발생한다. 이러한 문제들을 해결하기 위해서 첫 번째로 바이섹터-서피스의 꼭지점 사이 거리를 조절할 수 있어야 한다. 두 번째로 메인 카메라가 비추는

화면과 보조 카메라가 비추는 화면이 비슷하지 않도록 조절하여야 한다. 위의 두 조건을 만족하여 보조 카메라가 효율적으로 물체 사이의 충돌 위험을 관찰할 수 있어야 한다. 본 논문에서는 위의 문제를 해결하기 위해 사용자가 직접 조작하는 메인 카메라의 위치와 방향을 이용하여 \mathbb{C} 의 점들 중에서 보조 카메라의 위치를 최적화한다.

본 논문은 일반적인 3차원 투시도(perspective view)를 이용하여 공간을 표현한다. 이때 메인 카메라의 시점을 여러 방향으로 이동하였을 때, 두 물체의 형태를 파악하기 어려운 위치는 메인 카메라가 보고 있는 시점의 수직 방향이다. 간단한 예를 들어, 도미노 여러 개를 일렬로 세워놓고 첫번째 도미노의 정면에서 바라보면 뒤쪽으로 몇개의 도미노가 있는지 알 수 없다. 이는 도미노를 세워 놓은 측면에서 관찰해야 알 수 있는 정보이다. 본 논문에서도 계산을 최소화하고 실시간으로 두 물체 사이의 충돌 위험을 관찰하기 위해, 메인 카메라가 물체를 보는 방향과 수직 방향에서 점 O 를 관찰하는 보조 카메라의 위치를 찾기로 하였다. 본 논문에서 제안한 제약조건을 만족하는 보조 카메라의 위치 c^A 를 찾기 위해 다음과 같은 식을 최적화해야 한다.

$$c^A = \underset{c \in \mathbb{C}}{\operatorname{argmin}} (w_1 \times ((v^U)^n \cdot (v^C)^n)^2 + w_2 \times (c^U - c)^2)$$

$$v^C = O - c$$

\mathbb{C} 는 4.1절에서 구한 바이섹터-서피스 위의 점들 중 해당 위치에서 p^O, q^O, O 를 모두 관찰할 수 있는 연속적인 점들의 집합이다. c 는 \mathbb{C} 에 속하는 임의의 점이다. v^U 는 \mathbb{C} 에 속하는 임의의 점 c 에서 메인 카메라의 방향 벡터이다. v^C 는 임의의 점 c 에서 관찰 위치 O 를 향하는 방향 벡터이다. $(v^U)^n, (v^C)^n$ 는 v^U 와 v^C 를 각각 정규화(normalization)하였다. c^U 는 메인 카메라의 위치이다. 위의 식을 정확하게 최적화하기 위해서 바이섹터-서피스를 미분 가능한 연속적인 서피스로 표현해야 하나 계산의 복잡도가 높아진다. 본 연구에서는 계산 복잡도를 줄이기 위해 4.1절에서 제시한 바와 같이 Xi Zhao[17]의 방법으로 계산된 바이섹터-서피스 메시 모델의 꼭지점만을 대상으로 하여 이산적인 계산법을 적용하였다. 수정된 식은 다음과 같다.

$$c^A = \underset{d \in \mathbb{D}}{\operatorname{argmin}} (w_1 \times ((v^U)^n \cdot (v^d)^n)^2 + w_2 \times (c^U - d)^2)$$

$$v^d = O - d$$

\mathbb{D} 는 바이섹터-서피스 메시 모델에서의 꼭지점 중 세 점 p^O, q^O, O 으로 충돌 없이 직선으로 연결 가능한 점들의 집합이다. d 는 \mathbb{D} 에 속하는 임의의 점이다. v^d 는 임의의 점 d 에서 관찰 위치

O 를 향하는 방향 벡터이다. $(v^d)^n$ 는 v^d 를 정규화 하였다. w_1 과 w_2 는 실험을 통해 적합한 값을 찾았다($w_1 = 1, w_2 = 0.01$).

위의 식을 통해 구한 보조 카메라의 위치 c^A 에서 관찰 위치 O 를 향하는 방향 벡터($O - c^A$)는 메인 카메라의 방향 벡터와 수직이다. 이 위치는 두 물체에 의해 시야가 가려지지 않는다. 그러나 이 서피스는 Figure 3에서 확인할 수 있는 것처럼 두 물체보다 큰 크기로 생성된다. 따라서 c^A 는 O 에서 충분히 먼 위치에 선택될 수 있다. 이러한 문제를 방지하기 위해 c^A 가 O 에서 일정 거리 이상으로 멀어지지 않도록 최대 거리를 주었다.

5. 실험

본 논문에서 제안하는 자동 보조 카메라의 효용성을 확인해보기 위해 사용자 인터페이스를 제작하였다. 메인 뷰-포트는 메인 카메라의 시점에서 보여주는 화면으로 사용자가 직접 카메라를 조작할 수 있다. 오른쪽 상단의 서브 뷰-포트는 보조 카메라의 시점에서 보여주는 화면으로 본 논문에서 소개하는 알고리즘에 의해 보조 카메라의 위치가 조작되어 두 물체 사이에서 발생할 수 있는 충돌 위험 영역을 관찰할 수 있다.

5.1 구현

보조 카메라 위치를 한 번 최적화하는데 필요한 계산 시간은 P와 Q가 포함하는 꼭지점의 총 개수에 영향을 받는다. 가장 많은 계산을 차지하는 부분은 드로네 삼각화와 보로노이 다이어그램을 계산하는 시간이다. 본 연구에서는 계산의 효율성이 증명된 퀵헐(quickhull) 알고리즘[18]을 사용하였다. 실험결과 Intel(R) Core(TM) i7-6700 CPU를 장착한 컴퓨터에서 꼭지점의 수가 1700 개 미만일 때 보조 카메라 위치의 실시간 갱신이 가능함을 확인하였다.

최적화 계산에 비연속적으로 변화하는 파라미터 (꼭지점 위치)가 사용되므로 상황 변화에 따른 보조 카메라의 위치 변화도 비연속적일 수밖에 없다. 하지만, 비연속적인 카메라 위치 변화는 보기에 불편하고 사용자가 카메라의 현재 위치를 쉽게 인지할 수 없게 한다. 이 문제를 해소하기 위해서 보조 카메라 위치의 변화가 필요할 때 최적값을 바로 대입하지 않고 아래의 식과 같은 방법으로 현재 위치 l_0^A 를 30ms마다 갱신하여 최적값에 점진적으로 수렴하도록 하였다.

$$l_{i+1}^A = l_i^A * 0.9 + c^A * 0.1$$

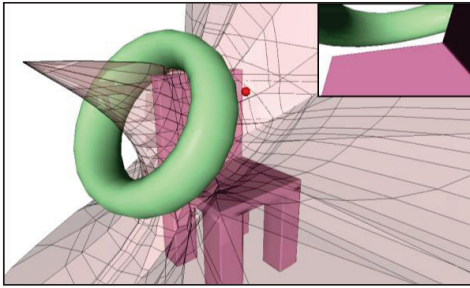


Figure 4 Chair and donut example.

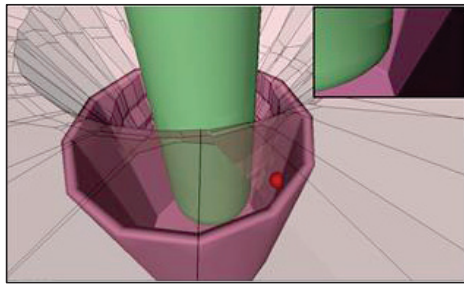


Figure 5 Cup and stick example.

5.2 사각지대 관찰 실험

본 실험은 3인칭 시점에서 두 물체의 충돌 위험 영역을 관찰한다. 본 논문에서 제시하는 보조 카메라의 효용성을 확인하기 위해 여러 가상 환경을 제작하여 실험하였다. 아래의 Figure 4와 Figure 5는 각각 두 가지의 물체가 근접하게 배치되어 있는 모습을 보여준다. 이때, 두 물체는 물체의 어느 부분도 충돌하지 않도록 조작하였다. Figure 4는 의자 모델에 도넛 모델이 걸쳐져 있는 모습이다. Figure 5는 컵 모양 모델 안에 막대 모델이 들어가 있는 모습이다. 두 그림 모두 고정된 물체가 근접하게 배치되어 있을 때, 충돌 위험 영역을 찾아낸다. 이 영역은 그림의 오른쪽 상단에 있는 서브뷰-포트에서 확인할 수 있다. Figure 4와 Figure 5에서 볼 수 있는 빨간 점은 보조 카메라의 위치를 나타낸다. 결과적으로 빨간 점은 바이섹터-서피스 위에 존재하며 물체와 서로 겹치지 않는다(Figure 4에서 빨간 점이 물체와 겹치는 이유는 사용자가 빨간 점을 쉽게 확인할 수 있도록 점의 크기를 크게 생성해서 생긴 것이다). 결과적으로 보조 카메라는 빨간 점의 위치에서 두 물체 사이에서 충돌이 발생할 수 있는 위치를 효율적으로 비추고 있는 것을 확인할 수 있다. 충돌 위험이 있는 영역이 다른 물체에 의해 가려지지 않으며, 두 물체의 영역을 모두 표시하여 이 영역의 거리 각격을 확인할 수 있다.

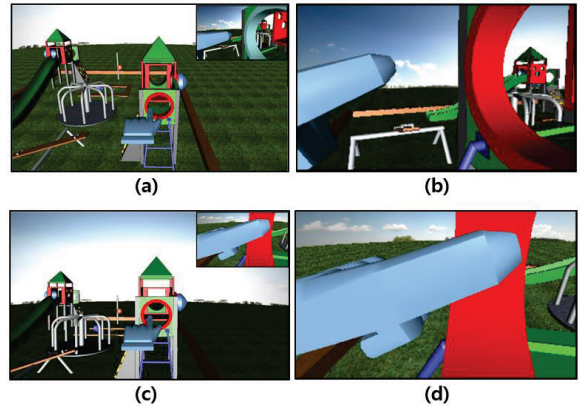


Figure 6 Exploring 3D virtual world with first-person perspective. (a) and (c) are the main camera views (first-person views) at different times, and (b) and (d) are their secondary camera view, respectively. In (a) and (c), it is very difficult to say at which time the finger is in the hole. However, in (b) and (d), it is obvious that the finger in (d) is in the hole but not in (b).

5.3 1인칭 시점 가상환경 탐험

본 실험은 1인칭 시점에서 핸드 모션 인식 장치를 이용하여 가상 환경을 탐험하며 충돌 위험을 관찰한다. 실험을 진행하기 위하여 본 논문에서는 핸드 모션 인식 장치 중 하나인 립 모션 컨트롤러(Leap Motion Controller)를 이용하였다. 립 모션 컨트롤러는 특별한 센서를 손에 장착하지 않고 컴퓨터에 연결하면 쉽게 사용할 수 있다. 또한 간편한 사용법에 비해 3차원 공간의 손의 미세한 움직임까지 추출해 낼 수 있다는 장점을 가지고 있다. 따라서 손 동작의 미세한 움직임이 필요한 본 연구에서는 립 모션 컨트롤러를 핸드 모션 인식 장치로 이용하였다.

본 실험은 1인칭 시점에서 움직이는 물체와 고정된 물체 사이의 충돌 위험을 살펴보기 위해 Figure 6과 같이 가상의 놀이터 모델과 손 모델을 제작하였다. 놀이터 모델은 주어진 가상 환경에 고정되어 있는 모델이고, 손 모델은 핸드 모션 인식 장치를 이용하여 사용자가 보고 있는 1인칭 시점에서 상하좌우 움직이는 등 미세한 동작이 가능하도록 설정하였다.

본 실험은 크기가 비슷한 두 물체를 이용한 5.2절에서 수행한 실험과 달리 놀이터 모델이라는 큰 가상 환경에서 진행된다. 이때 계산 상의 효율을 위하여 두 물체 사이에서 충돌 위험이 있는 영역만 고려하도록 하였다. 이는 사용자가 조작하는 손 모델의 위치에 따라 일정한 영역에 속하는 메시 모델의 점들만 계산에 이용하는 방법으로 해결하였다. Figure 6은 5.3절에서

제한한 1인칭 시점의 가상 환경을 구현해 놓은 그림이다. (a)와 (c)는 사용자가 확인할 수 있는 사용자 인터페이스의 전체 화면을 나타내는 그림이고, (b)와 (d)는 보조 카메라의 시점을 비추어주는 서브-뷰포트를 확대한 그림이다. (a)와 (c)의 그림의 메인 화면을 살펴봤을 때, 1인칭 시점에서는 손 모델이 미끄럼틀 모델에 닿았는지 혹은 겹쳤는지, 물체를 통과하였는지 알아보기 어렵다. 이를 보조 카메라의 시점에서 비춰주는 화면인 (b)와 (d)를 통해 확인할 수 있다. (b)의 경우 손 모델의 손가락이 물체에 닿지 않은 것을 알 수 있고, (d)의 경우 손 모델의 손가락이 미끄럼틀 모델의 입구를 통과한 것을 알 수 있다.

1인칭 시점에서 손 모델을 움직이다 보면 다른 물체와 겹쳐지는 문제가 발생한다. 이때 바이섹터-서피스도 물체 사이에 겹쳐져서 생성된다. 보조 카메라는 바이섹터-서피스 메시 모델의 꼭지점 위에 위치하기 때문에 이렇게 두 물체와 서피스가 겹친 경우 보조 카메라 시점에 문제가 발생한다. 두 물체가 겹쳐진 일부에 그 어느 물체 사이의 거리보다도 가까운 지점이 생기는데, 이 점을 보조 카메라가 관찰할 수 있어야 한다. 그러나 본 논문에서 제시한 방법으로는 이 점을 바라보는 위치가 물체 내부에 존재하는 바이섹터-서피스 꼭지점 위에 정해질 수 있다. 따라서 이러한 문제를 방지하기 위해 본 실험에서는 두 물체가 충돌하는 것을 막지 않지만, 두 물체 내부에 존재하는 바이섹터-서피스의 꼭지점은 제외하고 보조 카메라의 위치를 찾을 수 있도록 하였다.

6. 사용자 조사

6.1. 실험 방법

본 논문의 효용성을 확인하기 위해 5.3절의 실험 방법을 이용하여 사용자 조사를 진행하였다. 실험은 두 가지 방법으로 진행하였다. 첫 번째는 보조 카메라가 없는 환경에서 손 모델을 이용하여 놀이터 모델의 입구에 올바르게 손 모델의 손가락이 들어갔는지 확인한다. 사용자가 손 모델의 손가락이 미끄럼틀 모델 주변에 겹치지 않고 올바르게 중점에 위치하였다고 생각하면 첫 번째 실험을 종료한다. 그리고 메인 카메라를 조작하여 손 모델이 정확한 위치에 도달하였는지 확인하고 걸린 시간을 확인한다. 다음으로 두 번째 방법을 진행한다. 두 번째 실험 방법은 보조 카메라가 있는 환경에서 첫 번째 실험을 동일한 방법으로 수행한다. 사용자는 고정된 메인 화면에서 서브 뷰-포트를 확인하며 손 모델의 손가락을 위치시킨다. 마찬가지로 손 모델이 정확한 위치에 도달하였다고 생각하면

Test method	Type		Time
Without secondary camera	Success	1	32 s
	Collision	5	
	Not enter	4	
With secondary camera	Success	8	21 s
	Collision	2	
	Not enter	0	

Table 1 Result of our user study.

실험을 종료하고, 정확한 위치에 도달했는지 여부와 걸린 시간을 확인한다.

실험을 위해 제약조건을 하나 두었다. 이는 메인 카메라의 위치를 고정하는 것이다. 본 논문의 효용성을 확인하기 위해 제작한 사용자 인터페이스는 프로그램을 처음 실행하였을 때, 메인 카메라가 가상 환경의 중앙에 위치하도록 되어있다. 이때, 피실험자들의 사용자 인터페이스의 조작 환경을 접해본 유무에 따라 화면의 중앙에서 미끄럼틀 모델의 입구를 찾는데 걸리는 시간이 차이가 났다. 이는 본 연구의 효용성을 확인하는데 꼭 필요한 과정이 아니라 생각된다. 따라서 메인 카메라의 위치를 Figure 6의 (a)와 (c) 같이 미끄럼틀 모델의 입구에 위치시켜 놓은 상태로 진행하였다.

6.2. 실험 결과

본 실험은 대학생 10명(여자 4명, 남자 6명, 20~25세)을 모집하여 진행하였다. 실험 방법은 6.1절에서 소개한 방법으로 진행하였다.

첫 번째로 보조 카메라를 제공되지 않은 가상 환경에서 사용자는 손 모델을 미끄럼틀 모델의 입구에 통과시킨다. 실험을 진행하는데 걸린 시간은 평균 32초가 소요되었다. 10명의 피 실험자들 중 9명의 실험자가 두 물체끼리 충돌하거나 손 모양이 입구에 들어갔다고 판단하였다. 자세한 수치는 Table 1 과 같다. 두 물체가 충돌한 경우, 본 실험에서 제안하는 손 모델의 특성상 손가락이 미끄럼틀 입구에 들어갔으나 겹쳐져 있는 나머지 손가락들이 미끄럼틀의 입구가 아닌 부분과 충돌한 경우가 대부분이었다.

두 번째로 보조 카메라를 제공하는 가상 환경에서 같은 실험을 진행하였다. 피실험자들은 보조 카메라에 의해 두 물체 사이의 근접한 영역을 보여주는 서브 뷰-포트를 확인하며 손 모델을 미끄럼틀 모델의 입구로 위치시켰다. 두 번째 실험에서 피실험자들은 서브 뷰-포트를 확인하며 물체를 정해진 위치에 위치시키는데 첫 번째 실험에 비해서 평균 10초 가량 짧게 소요되었다. 또한, 10명의 피실험자 중에서 8명이 손 모델을

충돌 없이 정확한 위치에 위치시켰다. 피실험자 중 두 명은 손가락은 미끄럼틀 입구에 들어갔으나 접혀 있는 손가락 부분이 미끄럼틀 모델과 겹치면서 충돌이 발생하였다. 자동 보조 카메라를 이용한 두 번째 실험에서 두 명의 실험자만 실험에 성공하지 못했다. 이는 첫 번째 실험에서 한 명의 실험자만 성공한 것에 비해서 큰 성과라고 판단할 수 있다. 따라서 실험을 통해 본 논문에서 제안한 자동 보조 카메라의 효용성을 입증하였다.

7. 결론 및 향후 연구 과제

본 논문은 3차원 가상 환경에서 두 물체가 근접하게 배치되어 있을 때, 사용자가 메인 카메라를 크게 조작하지 않고도 두 물체 사이의 충돌 위험을 관찰하기 위한 자동 보조 카메라를 제안하였다. 수행 방법은 다음과 같다. 먼저, 두 물체 사이에 가장 근접한 위치를 찾아서 관찰 위치로 지정한다. 다음으로 바이섹터-서피스를 이용하여, 두 물체 사이에 존재하며 물체에 겹쳐지지 않는 보조 카메라의 위치를 찾는다. 이 위치에서는 두 물체 사이의 충돌 위험을 관찰할 수 있으며, 다른 물체에 의해 시야가 가려지지 않아야 한다. 본 연구의 효용성을 확인하기 위해 사용자 인터페이스를 제작하고, 1인칭 시점과 3인칭 시점으로 환경을 나누어 실험을 진행하였다. 또한, 사용자 조사에서는 1인칭 시점 환경에서 사용자가 두 물체 중 하나의 물체를 직접 조작하여 보조 카메라의 효용성을 입증하였다.

그러나 본 논문에서 제안한 방법은 두 물체의 전체 표면을 이용하여 보조 카메라의 위치를 계산한 것이 아니다. 계산 상의 효율을 위하여 물체에 샘플링 되어있는 점을 이용하였기 때문에 이 점들의 위치와 개수에 따라 실험을 진행할 때마다 오차가 발생하였다. 특히 1인칭 시점 환경의 경우 사용자가 물체를 조작하는 위치에 따라 바이섹터-서피스의 형태 달라졌다. 또한, 보조 카메라는 메인 카메라의 위치에 따라 자동으로 위치를 찾아내기 때문에 더 나은 위치를 표시하기 위해서 메인 카메라를 조금 더 조작해야 한다는 아쉬움이 있다. 따라서 물체 위에 있는 연속적인 점을 이용하여도 효율적인 시간 안에 보조 카메라가 자동으로 위치를 찾을 수 있도록 한다. 이는 두 물체 사이의 근접한 충돌 위험을 찾아내고 비추주는 보조 카메라 위치에 대한 정확성을 더 높일 수 있을 것이다. 또한, 본 논문에서는 핸드 모션 인식 장치(특히 립 모션)만을 이용하였지만, 사용자가 HMD 기기와 컨트롤러를 사용할 수 있는 가상 환경에서도 세밀한 조정이나 작업이 필요한 경우에 효율적으로 작업을 진행할 수 있도록 자동 보조 카메라를 제공하는 것이 본 연구의 최종 목표이다.

감사의 글

본 연구는 2017 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업 (NRF-2016R1D1A1B03930472)의 지원으로 이루어졌습니다. 그리고 2017 년도 가톨릭대학교 교비연구비의 지원으로 이루어졌습니다.

References

- [1] Yeh IC, Lin WC, Lee TY, Han HJ, Lee J, and Kim M, "Social-Event-Driven Camera Control for Multicharacter Animations." IEEE Trans Vis Comput Graph. 2012 Sep;18(9):1496-510. doi: 10.1109/TVCG.2011.273. Epub 2011 Nov 1.
- [2] Bill Tomlinson, Bruce Blumberg, Delphine Nain, "Expressive autonomous cinematography for interactive virtual environments", AGENTS 2000, Pages 317-324, doi>10.1145/336595.337513, 2000 June 03-07
- [3] Li-wei He, Michael F.Cohen, David Salesin, "The Virtual Cinematographer: A paradigm for Automatic Real-Time Camera Control and Directing", Siggraph. 1996 January, doi: 10.1145/237170.237259
- [4] Christophe Lino, Marc Christie, "Intuitive and Efficient Camera Control with the Toric Space", ACM Transactions on Graphics, Association for Computing Machinery, 2015, Proceedings of SIGGRAPH 2015, 34 (4), pp.82:1-82:12.
- [5] Steven M. Drucker, David Zeltzer, "Intelligent Camera Control in a Virtual Environment", Graphics Interface 1994
- [6] Christophe Lino, Marc Christie, "Efficient composition for virtual camera control", SCA 2012, pages 65-70, July, ISBN:978-3-905674-37-8
- [7] Adrian Secord, Jingwan Lu, Adam Finkelstein, Manish Singh and Andrew Nealen, "Perceptual Models of Viewpoint Preference", ACM Transactions on Graphics 30(5), October 2011.
- [8] Helin Dutagaci, Chung Pan Cheung and Afzal Godil, "A Benchmark for Best View Selection of 3D Objects", 3DOR 2010, proceedings of the ACM workshop on 3D object retrieval, pages 45-50, October 25, 2010, Doi> 10.1145/1877808.1877819.
- [9] ELBER, Gershon; KIM, Myung-Soo. "The bisector surface of freeform rational space curves." In: Proceedings of the thirteenth annual symposium on computational geometry. ACM, 1997. p. 473-474.
- [10] PETERNELL, Martin. Geometric properties of bisector surfaces. Graphical Models, 2000, 62.3: 202-236.
- [11] ELBER, Gershon; KIM, Myung-Soo. Bisector curves of planar rational curves. Computer-Aided Design, 1998, 30.14: 1089-

1096.

- [12] Xi ZHAO; Myung Geol CHOI; Taku KOMURA. "Character-Object Interaction Retrieval using the Interaction Bisector Surface." In: Computer Graphics Forum. 2017. p. 119-129.
- [13] HU, Ruizhen, et al. Learning how objects function via co-analysis of interactions. ACM Transactions on Graphics (TOG), 2016, 35.4: 47.
- [14] GUERRERO, Paul; MITRA, Niloy J.; WONKA, Peter. RAID: a relation-augmented image descriptor. ACM Transactions on Graphics (TOG), 2016, 35.4: 46.
- [15] ZHAO, Xi, et al. Relationship templates for creating scene variations. ACM Transactions on Graphics (TOG), 2016, 35.6: 207.
- [16] JIN, Taeil; KIM, Meekyoung; LEE, Sung-Hee. Aura Mesh: Motion Retargeting to Preserve the Spatial Relationships between Skinned Characters. In: Computer Graphics Forum. 2018. p. 311-320.
- [17] Xi Zhao, He Wang and Taku Komura, "Indexing 3D Scenes Using the Interaction Bisector Surface", ACM Transactions on Graphics (TOG), Volume 33 Issue 3, Article No. 22, May 2014, doi>10.1145/2574860
- [18] BARBER, C. Bradford; DOBKIN, David P.; HUHDANPAA, Hannu. "The quickhull algorithm for convex hulls." ACM Transactions on Mathematical Software (TOMS), 1996, 22.4: 469-483.

〈저자소개〉



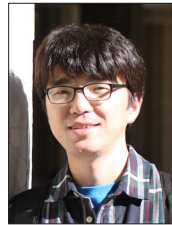
이 지 혜

- 2017 가톨릭대학교 미디어기술콘텐츠학과 학사
- 2017~현재 가톨릭대학교 디지털미디어학과 석사 재학중
- 관심분야 : 컴퓨터그래픽스 등



한 윤 하

- 2015~현재 가톨릭대학교 미디어기술콘텐츠학과 재학중
- 관심분야 : 영상처리, 딥러닝



최 명 길

- 2004년 경북대학교 컴퓨터공학과 학사
- 2010년 서울대학교 컴퓨터공학부 박사
- 2011년 JST ERATO Design Interface Project 박사후 연구원
- 2012년~2013년 IPAB, Edinburgh University, 박사후 연구원
- 2014년~현재 가톨릭대학교 미디어기술콘텐츠학과 미디어공학전공 조교수
- 관심분야: 컴퓨터 그래픽스, 캐릭터 애니메이션