

# Visibility를 이용한 3D게임에서의 grid-based filtering 개선방법

전승호<sup>o</sup> 김준태  
 동국대학교 대학원 컴퓨터공학과  
 (shjeon<sup>o</sup>, jkim)<sup>o</sup>@dgu.ac.kr

## Improvement of grid-based filtering using visibility in 3D game

Seungho Jeon<sup>o</sup> Juntae Kim  
 Dept. of Computer Engineering, Dongguk University  
 요약

MMORPG게임의 구현에 있어서 가장 큰 문제의 하나는 서버와 연결된 클라이언트들 사이의 동기화이다. 이를 위해서 MMORPG게임에서는 grid-based filtering을 적용하여 동기화 문제를 해결하고 있다. 그러나 MMORPG게임의 형태가 일인칭 관점 시점의 2D에서 일인칭 관점의 시점의 3D 게임으로 변함에 따라서 grid-based filtering으로 인한 동기화는 낭비적인 요소를 포함하게 되었다. 즉 일인칭 관점시점의 3D게임에서는 가까운 거리의 캐릭터 사이에도 주변 장애물의 영향으로 비 가시관계에 놓임에 따라서 동기화의 필요성이 없어지는 경우가 생기게 된다. 따라서 본 논문에서는 일인칭 관점 게임의 3D MMORPG게임에서 컴파일 시 각 셀을 중심으로 가시성 테스트를 수행하여 가시영역내의 셀들만을 동기화 대상으로 지정 한 후 게임 진행 시 현재 셀 속의 캐릭터들에게 동기화 대상으로 설정된 셀 속의 클라이언트들만을 동기화 시켜줌으로서 통신량을 줄여주는 방법을 제안한다.

### 1. 서 론

소프트웨어와 하드웨어의 발전 그리고 인터넷의 등장으로 게임의 형태는 과거 소수 유저가 함께 즐기는 게임에서 수백 수천 명이 서버에 연결되어 서로 상호작용을 하며 게임을 진행하는 클라이언트 서버 기반의 게임이 등장하였다. 이러한 클라이언트 서버 기반의 대표적인 게임에는 MMORPG(Massive Multi-player Online Role Playing Game)[7][8]이 있으며 MMORPG게임의 가장 큰 문제점의 하나는 클라이언트사이의 동기화의 해결이다. 대부분 MMORPG게임에서 동기화 문제의 해결을 위해 grid-based filtering[2][3]의 방법을 사용하여왔다.

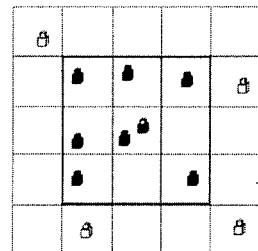
grid-based filtering은 게임속의 전체 맵을 각각의 요소가 직 사각형의 모양을 나타내는 2차원 배열의 셀들로 분할을 한다. 클라이언트의 캐릭터는 게임 진행시 맵의 한 부분에 속해 있으며 그 경우 맵의 부분을 가르치는 셀의 멤버로 들어가게 된다. 모든 클라이언트는 현재의 맵의 위치를 나타내는 셀을 점유하게 된다. 클라이언트의 캐릭터는 다른 캐릭터들과 상호작용을 위한 일정범위의 가시영역을 지고 있다. 따라서 서버에서는 하나의 셀 내의 클라이언트들의 모든 관심영역을 나타낼 수 있는 주변 셀 속의 클라이언트의 목록들을 추출하여 현재 클라이언트가 위치한 셀 내의 클라이언트들과의 동기화를 시켜 줌으로서 클라이언트는 실시간 가시영역내의 다른 캐릭터들과의 상호작용이 가능하다.[2][3][4]

본 논문에서는 일인칭 관점의 3D게임에서 컴파일 시 기존의 grid-based filtering을 구성하는 각각의 셀들 사이에 visibility test를 수행하여 게임 진행시 각각의 셀에서 가시관계의 셀들만을 동기화 시켜주는 개선방법을 제안한다. 이러한 방법을 통해서 일인칭 관점의 3D게임에서는 기존의 grid-based filtering에 비해서 장애물을 고려한 실제 가시영역내의 셀들을 한번더 추출하여 줌으로

서 서버측의 동기화를 위한 연산량을 감소시킬 수 있다.

### 2. grid-based filtering

grid-based filtering은 전체의 영역을 사각형 셀들로 이루어진 2차원 배열로 나눈 후, 셀 내의 모든 개체들의 상호작용 시 필요한 주변 셀을 현 셀에 동기화시켜주는 방식이다.[2][3][4] grid-based filtering을 MMORPG게임에 적용하는 방법은 다음과 같다. 게임속의 전체 맵을 각각의 요소가 사각형 셀들로 구성된 2차원 배열로 분할한다. 각각의 셀은 게임의 전체 맵의 한 부분을 나타낸다. 캐릭터는 게임 진행시 맵의 한 부분에 속해 있으며 그 경우 해당 셀의 멤버로 들어가게 된다. 각각의 캐릭터들은 게임 진행시 다른 클라이언트의 캐릭터들과의 상호작용을 위한 가시영역을 가지고 있다. 서버에서는 하나의 셀 속의 모든 캐릭터들의 가시영역 범위를 만족시킬 수 있는 주변 셀 내의 캐릭터들을 현 셀 내의 캐릭터들과 동기화를 시켜 줌으로서 모든 캐릭터 사이를 동기화 시켜주는 방법에 비해서 서버에는 많은 연산량의 감소가 발생한다.



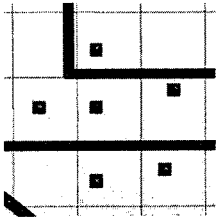
[그림 1]

위의 그림에서처럼 중심 셀에서 가시영역을 만족하는 주변 8개의 셀과 현 중심 셀 속의 클라이언트들을 현

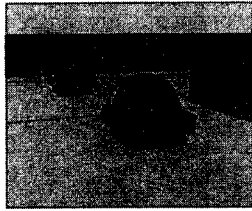
셀 속의 캐릭터들과의 동기화를 수행함에 즈음으로써 모든 캐릭터 사이에 동기화를 수행하는 기존의 방법에 비해서 많은 연산량의 낭비를 줄일 수 있다.

### 3. Visibility를 이용한 3D게임에서의 Grid-based filtering 개선방법

이인칭 관점 시각의 2D게임에서는 유저가 컨트롤 하는 캐릭터의 시각과는 상관없이 유저의 시각에는 캐릭터 주변의 모든 캐릭터들이 필요하였다. 그러나 일인칭 관점 시각의 3D게임에서는 서로 가까운 위치의 캐릭터사이에서도 지형이나 장애물의 특성으로 인해서 서로 비가시 영역에 존재할 경우 캐릭터 사이에는 동기화가 필요 없게 되며 이 경우 캐릭터를 컨트롤하는 유저의 화면에도 비가시영역의 캐릭터들은 필요가 없어진다.



[그림2]



[그림3]

[그림2]와 [그림3]은 같은 위치에서 하나의 캐릭터를 중심으로 이인칭 관점시점의 2D게임과 일인칭 관점시점의 3D게임의 시각효과를 보여준다. [그림2]에서는 캐릭터들의 시각과는 상관없이 유저에게는 캐릭터 주변의 모든 캐릭터들이 필요하다. 그러나 [그림3]에서는 게임속의 캐릭터가 볼 수 있는 것만 유저 또한 볼 수 있다. 따라서 [그림2]의 경우엔 게임진행시 5개의 캐릭터가 필요하며, [그림3]의 경우에는 2개의 캐릭터가 필요하다. 이처럼 일인칭 관점의 3D게임에서는 캐릭터가 필요로 하는 동기화의 대상은 일정한 가시영역내의 캐릭터들 중에서도 장애물로 인해서 가시관계를 방해받지 않는 캐릭터들이다.

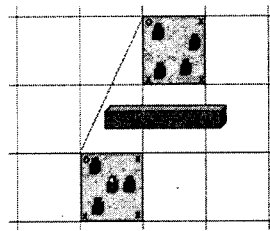
따라서 서버에서는 컴파일 시 전체 MAP을 이루는 각각의 셀을 대상으로 현재 셀과 가시영역내의 셀들 중 서로 가시관계에 있는 셀들을 동기화 대상으로 저장한 후 게임 진행시에는 현재 셀 속의 클라이언트들에는 지정된 셀 속의 클라이언트들만을 동기화시켜주는 방법을 제안한다.

#### 3.1 셀 사이의 Visibility 결정

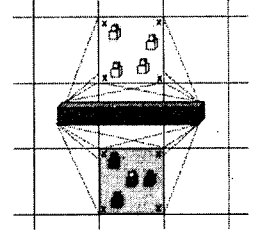
평면 또는 공간상의 두 점을 연결하는 선분이 어떤 장애물(점, 선, 다각형 등)과 만나지 않으면, 그 두 점은 서로 가시관계(visibility)에 있다고 한다. 두 점이 주어졌을 경우 두 점의 가시성 판단은 두 점을 잇는 선분과 주변 장애물을 대표할 수 있는 직선의 교차여부로 판별할 수 있다.[5]

grid-based filtering을 이루는 셀들의 가시성 여부는 각각의 셀을 이루는 4점들 사이의 가시성여부로 판별이 가능하다. [그림4]와 같이 A,B두 셀이 존재할 경우 A의 4꼭지점 각각에서 B의 4꼭지점 각각을 대상으로 가시성을 판단하여 어느 한점이라도 가시성의 범위에 들어온다면 두 셀은 가시성의 범위에 놓이게 된다. 그러나 [그림5]처

럼 A의 4꼭지점 각각에서 B의 4꼭지점 모두의 쌍이 비가시관계에 있으면 두 셀은 비가시 관계에 놓이게 된다.



[그림4]



[그림5]

컴파일 시 각각의 셀에는 이러한 가시성 판단을 통하여 가시관계에 있는 셀들을 추출한 후 셀의 참조변수를 통해 가시관계에 있는 셀들의 목록을 저장 시킨다.

#### 3.2 Visibility를 고려한 grid-based filtering

본 논문에서 제안하는 visibility를 이용한 3D게임에서의 grid-based filtering 개선방법은 다음과 같은 과정에 의해서 진행된다.

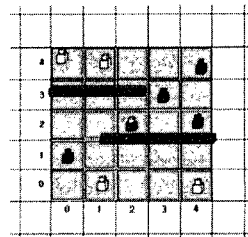
첫째, 3D MMORPG게임에서 게임이 시작되기 전에 서버에서는 접속하는 클라이언트 사이의 동기화를 위해서 기본적인 grid-based filtering을 적용하여 게임의 맵을 분할한다.

둘째, 캐릭터들 사이의 가시성을 불가능하게 할 수 있는 장애물들의 좌표를 추출하여 이들 좌표로 이루어진 장애물의 벡터 정보들을 저장시켜 놓는다.

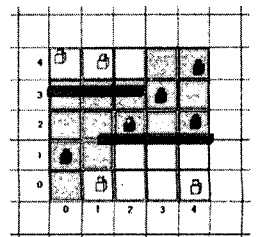
셋째, 맵의 나누어진 모든 셀을 기준으로 해당 셀의 4모서리 점과 주변 셀의 4모서리 점의 가시성을 판별한다. 만일 해당 셀과 주변 셀의 어느 한 점의 쌍이라도 가시성으로 판별되면 두 셀에 존재하는 캐릭터들은 서로 가시영역에 있을 수 있는 가능성이 있으므로 해당 셀의 동기화 목록에 주변 셀을 설정시킨다.

게임의 진행시에는 사전에 위의 과정을 거쳐서 각각의 셀에 설정된 가시영역 셀 속의 클라이언트만을 동기화시켜 준다.

이러한 일인칭 관점 게임의 3D MMORPG게임에서 기존의 grid-based filtering의 방법으로 하나의 동기화에 사용되는 주변영역의 셀의 개수를 N이라 하고, 본 논문에서 제안한 방법으로 하나의 동기화에 사용되는 주변영역의 셀의 개수를 M이라 할 때 본 논문에서 제안한 방법으로 클라이언트 사이의 동기화를 구현 할 경우 동기화에 따르는 기존의 연산량에 비해서  $((1-M/N)*100)\%$ 의 성능향상을 가져올 수 있다.



[그림6]



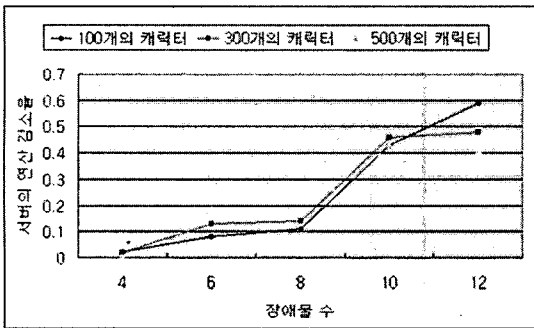
[그림7]

[그림6]과 [그림7]은 똑같은 일인칭 관점의 3D게임의 상황이라고 가정할 때 기존의 grid-based filtering과 본 논문에서 제안한 방법을 나타내고 있다. 하나의 셀의 가시영역을 만족시키는 주변 셀이 24개라고 가정할 때 가운데 셀을 기준으로 기존의 grid-based filtering에서는 [그림6]과 같이 가운데 셀을 포함한 25개 셀을 모두다 동기화의 대상으로 한다. 그러나 [그림7]과 같이 본 논문에서 제안한 방법을 적용할 경우에는 가운데 셀을 기준으로 장애물을 고려한 후 최종적인 동기화 대상의 셀의 개수는 15개로 줄어들게 된다.

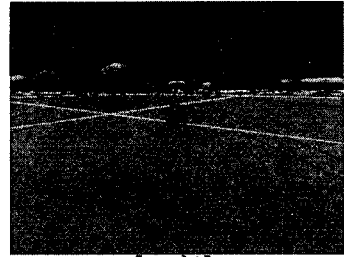
4. 실험 및 성능평가

본 논문의 실험을 위해서 OpenGL을 기반으로 캐릭터 사이에 가시관계를 방해할 수 있는 장애물 4, 6, 8, 10, 12개를 각각 설정한 5개의 3D게임 환경을 만들었다. 각각의 캐릭터들은 메시로 이루어진 지형위에서 서로의 목표점을 향해 이동 한다. 캐릭터의 개수는 각각 100, 300, 500개를 대상으로 5개의 환경에서 실험하였으며 전체 맵은 가로세로 20개의 400개의 셀로 분할하였다. 기존의 grid-based filtering과 본 논문에서 제안한 방법을 적용하여 서버가 일정시간 동안 클라이언트의 동기화를 위해서 처리하는 데이터 송신량을 측정한다 뒤 기존의 방법에 대해 본 논문에서 제안한 방법으로 인한 연산량의 감소를 백분율로 나타내었다. 실험은 cpu 1.84G 와 512RAM을 갖는 PC상에서 c++로 시뮬레이션 프로그램을 작성하여 수행 하였다.

[표1]



실험결과 평균적으로 기존의 방법에 비해서 가시영역내의 셀들만을 동기화시켜주는 본 논문의 방법이 서버의 연산량을 22% 줄여준다는 것을 알 수 있다. 또한 캐릭터 사이의 가시관계를 방해하는 장애물이 증가함에 따라서 서버의 연산량 감소가 증가하는 것을 알 수 있다. 또한 장애물의 위치나 종류에 따라서 연산량의 증가폭이 다른 것을 알 수 있다. 즉 캐릭터들이 자주 모이는 부분의 장애물이나 큰 장애물들은 캐릭터사이의 가시관계에 많은 영향을 주므로 이러한 장애물은 서버의 연산량을 크게 줄여주며, 캐릭터들이 잘 모이지 않는 지형에 존재하는 장애물이나 작은 크기의 장애물들은 서버의 연산량 감소에 작은 영향을 주게 된다.



[그림8]

[그림8]은 본 실험환경 상에서 하나의 캐릭터를 중심으로 한 일인칭 관점형태의 게임진행 데모를 보여주고 있다.

5. 결론

본 논문에서는 일인칭 관점 게임의 3D MMORPG게임에서 컴파일 시 각 셀을 중심으로 가시성 테스트를 수행하여 가시영역내의 셀들만을 동기화 대상으로 지정한 후 게임 진행 시 현재 셀 속의 캐릭터들에게 동기화 대상으로 설정된 셀 속의 클라이언트들만을 동기화 시켜줌으로서 서버의 데이터 송신량을 줄여주는 방법을 제안하였다. 실험결과 일인칭 관점의 3D게임에서 캐릭터사이의 가시관계를 방해할 수 있는 장애물이 존재할 경우 본 논문의 방법을 적용함으로써 기존의 grid-based filtering의 방법에 비해서 22%의 성능향상을 확인할 수 있었다.

[참고문헌]

- [1] Diot, C., Gautier, L, "A distributed architecture for multiplayer interactive applications on the Internet", *IEEE*, 1999
- [2] D.J. Van Hook, et al., "Approachs to relevance filtering", *11th Workshop on Standards for the Interoperability of Distributed Simulations*, 1994
- [3] Lety, E., Turletti, T., Baccelli, F, "Cell-based multicast grouping in large-scale virtual environments", *Measurement and modeling of computer systems*; *ACM*, 2000
- [4] Morse, K. L., Zyda, M, "Multicast grouping for data distribution management", *Simulation practice and theory*, 2002
- [5] J. O'Rourke, "Computational Geometry in C", *Cambridge Press*, 1998
- [6] Zou, L., Ammar, M., Diot, C, "An evaluation of grouping techniques for state dissemination in networked multi-user games", *Modeling, analysis and simulation of computer and telecommunication systems*, 2001
- [7] 이승욱, "MMORPG 게임엔진의 성능개선을 위한 분할공간에서의 충돌검출", *한국정보처리학회 논문지 B*, 2003
- [8] 전일곤, 오삼권, "대용량 킬-플레이밍 게임을 위한 AOI에 관한 연구", *한국정보처리학회 2004년 춘계학술대회*, 2004