

# 대규모 전장공간의 실시간 렌더링 기법 Real-Time Rendering of Massive Battlefields

김영현 김태준 윤성의  
Young-Hyun Kim Tae-Joon Kim Sung-Eui Yoon  
한국과학기술원 전산학과(KAIST)  
(010-6424-0535, yhkim0535@gmail.com)

## Abstract

We introduce several techniques from the graphics field to accelerate the rendering process of massive scenes and show their benefits with massive battlefields, which have large/complex terrain and many highly detailed objects. These techniques include view frustum culling, occlusion culling, and levels-of-detail (LODs). In our tests, view frustum and occlusion culling improve performance up to two orders of magnitude and four times, respectively than the brute-force method without any quality degradation. LOD improves 5.5 times without noticeable quality degradation. We observe at least 10 times and up to 65 times performance improvement depending on various scene setting.

**Keywords :** real-time rendering, levels-of-detail, occlusion culling, frustum culling, clustering

## 1. 서 론

전장환경을 실시간으로 시뮬레이션하기 위해서는 전장공간정보(전장을 구성하는 정보로 지형정보, 객체정보, 전술 정보 등의 총칭)의 효율적인 처리 기술 개발이 절실히 요구 된다. 육군의 경우만 해도, '전투21' 모델이 '창조21' 모델에 비해 보다 세밀한 전장환경을 표현하고 있는데, 개별 병사수준에서, 여단, 사단급 이상의 통합된 수준까지의 정보를 교환하고 적절히 표현하는 기술이 마련되어 있지 않은 형편이다. 뿐만 아니라 다중 해상도의 위게임 시뮬레이션을 가시화 할 때, 실제 전장 환경과 같은 장면을 만들어 내기 위해서는 수많은 오브젝트들을 가시화해야 하는데 기존의 일반적인 방법을 사용하면 처리속도가 느려 실제 전장환경과 같은 고화질 장면을 만들어 내는데 어려움이 있다. 이러한 문제들을 해결하기 위해서는 오브젝트들의 효율적인 가시화 방법에 대한 연구가 필요하다.

장면(scene)의 고성능 가시화, 특히 고화질의 사실적 렌더링(rendering)은 그래픽스 분야의 가장 중요한 연구 주제 중 하나이다. 영화에서 사용되는 초고화질 렌더링에서부터 최근 눈부시게 발전하는 게임 영상을 위한 실시간 렌더링까지 보다 복잡한 장면을 고속으로 가시화하기 위해 수많은 연구가 진행되고 있다. 앞에서 제시한 대규모 전장환경의 실시간 가시화의 문제를 해결하기 위해 본 논문에서는 그래픽스 분야에서 오랜 기간 연구되어온 가시화 기술들을 소개하고, 이러한 기술들이 전장 환경에서 어느 정도의 성능향상을 이룰 수 있는지 실험하였다.

## 2. 관련 연구

해외의 여러 대학들과, 연구소, 산업계 에서 국방 M&S (Modeling and Simulation) 에 대하여 많은 연구가 이루어지고 있다. 미국의 버지니아 대학에서 군관련 aggregate / disaggregate 기술을 중점적으로 연구하고 있고, 미 국방성은 합성전장환경 정보를 표현하고 교환하기 위해 SEDRIS (Synthetic Environment Data Representation and Interchange Specification) 기반 기술을 개발하고 표준화 작업<sup>[1,2]</sup>을 진행 중이며, 합성전장 환경(지형 및 무기체계)를 실제로 구현하기 위한 SceneGraph 기술이 활발하게 개발되고 있다. 산업계에서는 지형과 무기체계간의 상호작용을 위해 DARPA STOW 프로그램에서 Dynamic Terrain이란 개념을 소개하고, 이를 HLA/RTI 분산환경에서 운영가능한 기술을 개발하였다<sup>[3,4]</sup>. 국내의 경우 지형과 같은 환경정보 표현 및 교환을 위한 국제표준인 SEDRIS 작업에 국내 업체인 Codic Communications가 참여하여 지형 정보 표현에 관한 높은 수준의 기술이 개발되었다. 이와 같이 전장공간 정보의 효율적인 가시화 및 지형 표현을 위한 연구는 국내외로 활발히 진행되고 있지만 수많은 오브젝트가 포함된 대규모 전장공간의 고화질 렌더링에 관한 연구는 아직 많은 발전의 여지가 있다.

대규모 장면의 효율적인 렌더링은 그래픽스분야에서 오랜 기간 동안 연구되어 왔다. 가장 많이 연구되고 사용되는 기술 중 하나는 카메라 시점에 따라 오브젝트를 간략화 하여 렌더링 기본개체(primitive)의 수를 줄여 성능을 높이는 것이다<sup>[8]</sup>. 카메라로부터 멀리 떨어진 오브젝트의 경우 간략화 하여도 최종 결과가 크게 차이나지 않기 때문에 이러한 방법은 화질은 유지하면서 성

능은 높이는 방법으로 각광받고 있다. 이는 더 발전되어 오브젝트의 각 부분을 다른 수준으로 간략화 하기에 이르렀다<sup>[6]</sup>.

렌더링 기본개체를 줄이기 위한 또 다른 접근 방법은 각 개체가 실질적으로 렌더링 될지 여부를 미리 판단해서 그려지지 않을 부분을 사전에 잘라내는 것(culling)이다. 카메라의 가시 영역, 일명 절두체(view frustum)를 벗어나는 부분을 사전에 판별하여 컬링하는 방법(view frustum culling)<sup>[9]</sup>, 다른 물체에 가려지는 부분을 판별하여 컬링하는 방법(occlusion culling)<sup>[7]</sup> 등이 가장 많이 사용된다. 일정 크기를 감싸는 바운딩 볼륨(bounding volume)을 이용해 보존적(conservative)으로 컬링하는 방법으로 성능을 가속하기도 한다.

### 3. 효율적인 전장환경 가시화 방법

전장환경을 가시화 하는 방법은 상당히 중요하다. 데이터를 어떻게 관리하고 가시화 하느냐에 따라 시스템의 성능이 크게 다르다. 예를 들어, 1201x1201 크기로 샘플링 된 지형데이터에 사용되는 삼각형의 개수는 288만개((가로사이즈-1)\*(세로사이즈-1)\*2)인데 매 프레임마다 288만개의 삼각형을 모두 화면에 그린다면 상당한 자원을 필요로 하며 지형의 크기가 커질수록 부하(overhead) 또한 선형적으로 증가한다. 즉 지형을 가시화 할 때 지형의 삼각형을 모두 렌더링 하는 것은 시스템 성능 저하의 주된 원인이다. 이를 해결하기 위해서는 데이터의 복잡도를 감소시켜야 한다. 데이터 복잡도 감소방법으로는 지형을 일정 크기로 나누어 데이터 처리영역을 감소시키고 지형을 관리하는 clustering 기법, 시야에 보이지 않는 불필요한 삼각형을 가시화목록에서 제거하여 복잡도를 감소 시켜 주는 culling 기법이 있다. 컬링은 컬링하는 방식에 따라서 크게 view frustum culling과 occlusion culling 두 가지로 나뉘어진다.

#### 3.1 지형 clustering

지형데이터의 크기가 커질수록 데이터의 관리방식에 따라 시스템 성능에 큰 차이가 생긴다. 데이터를 관리하는 구조를 잘 설계 한다면, 데이터 처리 시 불필요한 데이터의 접근을 줄여 접근성을 높일 수 있다. 지형 clustering이란 거대한 지형의 정점들을 일정한 크기로 나누어 관리하는 것으로써, 특히 지형과의 연산을 많이 필요로 하는 작업을 할 때, 높은 성능 향상을 가져올 수 있다<sup>[6]</sup>. 또한 cluster 크기에 따라 다른 성능을 보이는데, 본 시스템에서는 [그림1]과 같이 cluster 크기가 33x33인 경우 가장 좋은 성능을 나타내었다.

#### 3.2 Culling

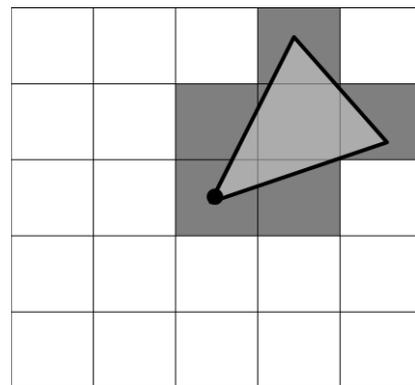
컬링은 가시화의 성능을 올려주는 방법으로, 보이지 않는 영역을 잘라내어 불필요한 정보를 가시화 시키지

Cluster Size	FPS	Number of Cluster for Vertices
5x5	14	300x300
9x9	70	150x150
17x17	242	75x75
33x33	251	37x37
65x65	240	18x18
129x129	217	9x9

[그림1] 1201x1201개의 정점이 있는 지형(2,880,000 triangles)에 Clustering 크기를 다르게 하여, View Frustum Culling 실험결과. 해상도 800x600, Geforce GTS250.

않도록 하여 성능 향상을 가져오는 방법이다.

**View Frustum Culling(절두체컬링)** 전체 3차원 공간에는 수많은 폴리곤과 오브젝트가 있지만 [그림2]와 같이 실제로 카메라 시야범위에 포함되는 영역만 가시화시켜주는 기법을 말한다. 절두체는 근평면(near plane), 원평면(far plane), 좌평면(left plane), 우평면(right plane), 상평면(top plane), 하평면(bottom plane) 총6개의 평면으로 이루어지고 절두체 컬링을 하기 위해서는 평면 방정식  $\vec{n} \cdot \vec{x} + d$  ( $\vec{n}$ : 법선벡터,  $\vec{x}$ : 평면상의 임의의 점,  $d$ : 평면과 원점의 거리)을 사용한다.  $\vec{x}$  대신 임의의 점을 대입했을 때 위 방정식의 값이 6개 평면 모두에 대해서 양수(또는 모두 음수)이면 이 점은 절두체 속에 포함된다.

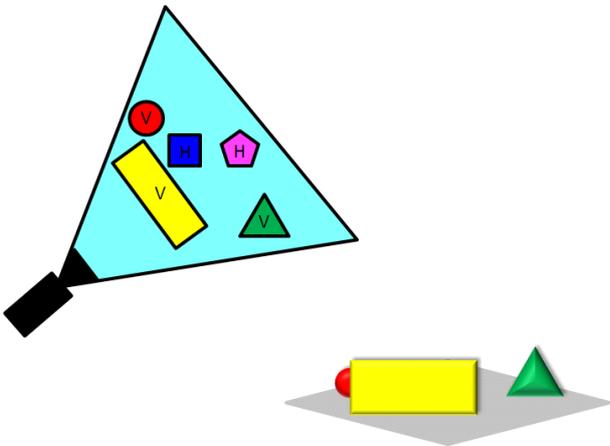


[그림2] View Frustum Culling시 흰 부분(Frustum의 영역)은 가시화목록에서 제거하고, 회색음영이 있는 부분(카메라 Frustum 영역)만 가시화 하여 성능을 향상 시킨다.

절두체컬링을 사용할 때 각각의 정점에 평면방정식을 대입하게 되면 1201x1201해상도 기준으로 1,442,401만큼의 정점들을 모두 계산해야 한다. 이는 카메라의 움직임이 있을 때 마다 계산되어야 하기 때문에 1,442,401만큼의 연산은 시스템 성능에 크게 영향을 줄 수 있다.

하지만 지형 clustering과 접목하여 사용한다면 cluster 크기 33x33개의 cluster 기준으로 1,369 만개의 연산으로 절두체컬링을 하게 된다. 지형 clustering 구성에 따른 절두체컬링의 성능차이를 비교한 [그림1]을 참고.

**Occlusion Culling** 오브젝트를 가시화 할 때 해당 오브젝트가 다른 오브젝트에 완전히 가려진 경우 가시화 하지 않도록 하는 culling방법이다[그림3]. Occlusion culling 을 사용하기 위해선 occlusion map을 구성해야 한다. 구성과정은 전 프레임에서 가시화된 모델을 z-buffer에 그리고, GPU 및 Graphic Library 에서 제공하는 API를 이용하여 오브젝트의 가시성을 구해 시야에 가려진 여부를 판별하고, 판별된 값으로 시야에 가려진 오브젝트를 가시화목록에서 제외 한다<sup>[7]</sup>. [그림5]의 (1)번과 같이 위에서 보았을 경우 다른 오브젝트에 가려지는 오브젝트가 없어 Occlusion Culling이 속도향상을 이루지 못하였고(가시 모델의 수: 150, fps 11), (2)번과 같이 옆에서 보았을 경우 오브젝트에 완전히 가려지는 상황이 많이 발생하여(가시 모델의 수: 22, fps 41) (1)번의 상황에 비해 약 4배의 성능향상이 있었다. 이처럼 Occlusion Culling기술은 깊이 복잡도가 높은 환경에서 많은 속도 향상을 얻을 수 있다.



[그림3] 카메라 Frustum에서 가려지는 오브젝트 검출(좌). 가려지는 오브젝트를 Culling하고 실제 가시화된 화면(우).

### 3.3 Levels-Of-Detail (LODs)

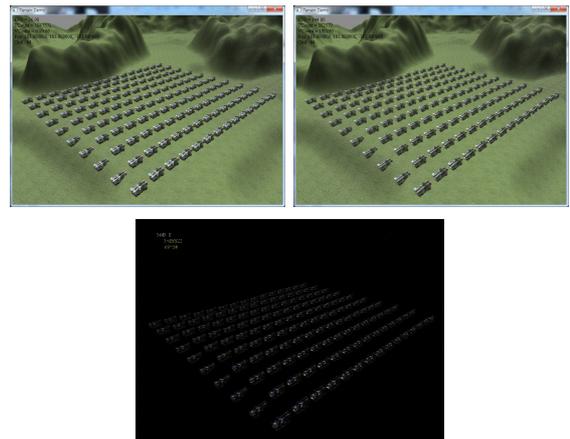
다중해상도 위게임 시뮬레이션을 가시화 과정에서 실제 전장환경과 유사한 장면을 만들기 위해서는 보다 정밀한 수준의 오브젝트를 사용해야 하는데 이는 시스템의 큰 부하가 된다. 그러나 LOD를 사용하면 실질적 정밀도는 유지하면서도 성능은 크게 향상시킬 수 있다. 많은 수의 삼각형을 포함한 정밀한 오브젝트는 카메라에 가까이 있는 경우(클로즈업) 화질 개선에 중요한 역할을 하지만 카메라와 멀리 떨어진 경우 조악한 오브젝트를 사용해도 최종적으로 렌더링 되는 결과는 크게 다

르지 않다. 따라서 다양한 수준의 정밀도의 오브젝트를 카메라와의 거리에 따라서 사용하면 적절한 화질은 유지하면서도 성능을 극대화 할 수 있다.

오브젝트를 간략화 하는 방법에는 미리 오브젝트의 간략화정도를 단계별로 저장하여 해상도에 따라 저장된 오브젝트의 모델을 런타임시 적당한 단계를 불러와 렌더링하는 정적 LOD 방식과, 실시간으로 카메라의 위치 관계에 따라 오브젝트(또는 오브젝트의 각 부분)의 간략화를 진행하여 다양한 LOD를 구성할 수 있는 동적 LOD가 있다.

### 3.3.1 Progressive Mesh

Progressive Mesh는 동적LOD의 한 종류로써, 반복적인 Edge Collapse방법을 사용하여 입력된 Mesh로부터 연속적인 해상도의 Mesh를 만들어내는 방법으로, 효율적인 데이터의 저장 및 전송, 가시화시간 단축의 장점을 갖는다<sup>[8]</sup>. Edge Collapse방법은 각 단계마다 하나의 정점(Vertex)과 하나 혹은 두 개의 면(Face)을 제거 하는데, 각각의 Edge Collapse를 되돌릴 수 있으므로 단순화과정을 거친 Mesh를 원래의 상태로 되돌릴 수도 있다. [그림4]처럼 장사포모델 150개를 Progressive Mesh를 적용시킨 결과. 원본 Mesh의 경우(1,047,572개의 삼각형) 초당 26프레임의 결과 볼 수 있었으며, Edge Collapse방법을 적용시킨 저해상도의 Mesh의 경우(282,572개의 삼각형) 초당 144프레임의 결과가 측정되어 약 5.5배의 성능향상을 보인다.



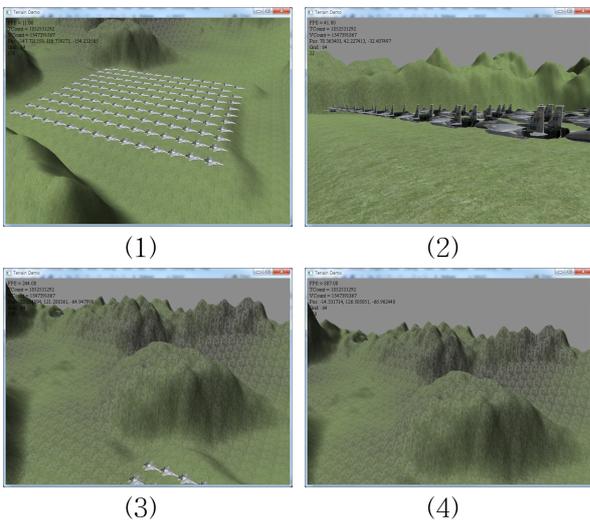
[그림4] 원본 메쉬(좌), Progressive Mesh(우), 원본메쉬와 Progressive Mesh의 이미지 차이(아래). Progressive Mesh의 경우 원본 메쉬에 비해 약 5.5배의 성능이 향상 되었다.

## 4. 결과

실험 환경은 Intel Quad-Core 3.00GHz CPU, Geforce GTS250, 4GB RAM, Windows 7 32bit OS를 사용 하였으며 800x600의 해상도로 실험을 하였다. 실험에 사용한 모델은 장사포모델 (6,110개의 삼각형,

640KB)과 F-15모델 (112,096개의 삼각형, 11.7MB) 이다.

Clustering기술과 View Frustum Culling을 함께 사용할 경우 [그림5]의 (1)번 기준으로 288만개의 렌더링할 삼각형(지형전체의 삼각형 수)을 1369번의 culling연산으로 33만개(culling되지 않은 삼각형)로 단축시키며 사용전과 후의 이미지의 차이가 없고, occlusion culling은 [그림5]의 (1)번과 (2)번과 같이 F-15모델이 카메라의 frustum 영역에 포함되어 frustum culling이 되지 않지만, occlusion culling에 의하여 약 4배의 성능향상이 이루어지고 있다. (3)번과 (4)번의 경우는 F-15모델이 카메라 frustum영역에 일부 있는 것과, 완벽히 culling된 것의 성능향상 차이를 보인다.



[그림5] F-15모델(112,096개의 삼각형) 150개를 위에서 보았을 경우: 11fps, 150개의 가지 모델 (1), 옆에서 보았을 경우: 41fps, 22개의 가지 모델 (2), 일부만 보았을 경우 244fps, 5개의 가지 모델(3), 보지 않을 경우 807fps (4).

## 5. 결론

진장환경시물레이션을 실시간으로 처리하기 위한 대규모 진장공간 실시간 렌더링기술을 연구 수행 중, 그래픽스분야의 기술들을 진장환경시물레이션에 활용하고 그 효과를 실험하였다.

Clustering기술의 경우, 정적인 모델(지형, 구조물)에 사용하기 적합하고 view frustum culling과 함께 사용할 시 높은 성능향상을 나타내며, occlusion culling은 [그림5]의 (1)번과(2)번의 경우처럼 오브젝트의 객체수가 많은 환경에 사용하기 적합한 것으로 나타났다. Levels-Of-Detail은 정적, 동적인 모델 모두에 사용가능하며 큰 성능향상을 볼 수 있다. 하지만 Edge Collapse 방법을 사용함에 따라 원본이미지와 약간의 차이가(이미지왜곡)생기고, 구현 또한 다른기술에 비해 어려운 편이다.

이 연구를 통하여 대용량의 진장공간정보를 3차원 가시화 함에 있어 실시간 처리가 가능함을 보였고 각 기술들의 성능개선이 어떠한 상황에 뛰어난 효과를 보일 수 있는지 확인 하였다.

## 참고문헌

- [1] SEDRIS organization., www.sedris.org
- [2] Tim Gifford, Farid Mamaghani., "An Introduction to SEDRIS", ITEC 2005, Amsterdam, 26/4/2005.
- [3] Michel D. and Brock D. L., "A Multiresolution 3D Environment for an Observer Based Multiresolution Architecture", In Proceedings of the Spring SIW, 1997.
- [4] Dale D. Miller, Steve Adelson, Tim Moore, Annette Janett, " Dynamic Terrain Implementation using the RTI", 15 th Distributed Interactive Simulation Workshop, 1999.
- [5] Sung-Eui Yoon, Brian Salomon, Russell Gayle and Dinesh Manocha., "Quick-VDR: Interactive View-Dependent Rendering of Massive Models", IEEE Transaction on Visualization and Computer Graphics 2005.
- [6] David Luebke, Carl Erikson., "View-dependent simplification of arbitrary polygonal environments", SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp. 199 ~ 208.
- [7] Hansong Zhang, Dinesh Manocha, Tom Hudson, Kenneth E. Hoff, III., "Visibility culling using hierarchical occlusion maps", SIGGRAPH '97 Proceedings of the 24th annual conference on Computer graphics and interactive techniques. pp. 77 ~ 88.
- [8] Hugues Hoppe., "Progressive Mesh", SIGGRAPH '96 Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, pp. 99~108.
- [9] Sekulic D., Frustum Culling, GPU Gems, R.Fernando, Ed. Addison-Wesley, Chapter 29.6.4, 2004.