

# 움직이는 물체의 개수에 비례하는 시간을 가지는 전역조명 알고리즘

정호욱, 오경수, 백두원

송실대학교 미디어학부

tidus-zheng@hotmail.com, {oks, dpaik}@ssu.ac.kr

## A Scalable Global Illumination Algorithm for Animated Graphics

Haoux Zheng, Kyungsu Oh, Doowon Paik

Dept. of Media, Soongsil University

### 요 약

간접조명은 실제감에 많은 도움을 주지만 많은 계산량이 필요하기 때문에 실시간 전역조명에서는 적용되기 힘들었다. 특히 2차광원에서의 가시성을 계산하기 위해서는 각 광원에 대해서 모든 물체들을 처리해야 하기 때문에 계산량이 많다. 본 논문에서는 움직이지 않는 물체들의 가시성을 재활용하여 움직이는 물체가 많지 않을 경우 효율적이면서도 2차광원의 가시성을 활용하는 간접조명 알고리즘을 제안한다. 기존 방법에 비해서 정확하면서도 효율적인 2차광원 가시성을 바탕으로 보다 복잡한 장면에서 사실적인 간접조명을 실시간에 가능하게 한다.

### ABSTRACT

Global illumination is key factor for realistic image but it is not common in realtime rendering due to its complexity. Visibility of 2<sup>nd</sup> lights is hard to determine because we should process all object for each 2<sup>nd</sup> light. In this paper, we reuse visibility of static objects. We present an indirect illumination algorithm which shows good performance and provides exact 2<sup>nd</sup> light visibility when the scene has small number of moving objects. Our method renders correct image while showing good performance.

**Keywords** : Global illumination(간접조명), shadow(그림자)

Received: Mar, 10, 2016      Revised: Apr, 14, 2016  
Accepted: Apr, 18, 2016  
Corresponding Author: Doowon Paik(Soongsil University)  
E-mail: dpaik@ssu.ac.kr

ISSN: 1598-4540 / eISSN: 2287-8211

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1. 서론

전역조명기법은 광원에서 직접 나오는 빛뿐만 아니라 주변 물체에서 반사하는 빛까지 표현하는 그래픽스 기술이다. 전역조명은 1차 광원이 반사되는 물체를 2차 광원으로 보고, 2차 광원의 위치, 방향, 강도 및 색 등을 계산하여 처리한다. 3차 이상의 광원은 계산량대비 화질개선 효과가 미비하기 때문에 본 논문에서는 2차광원까지만 고려한 전역조명을 다룬다.

전역조명을 적용하면 더욱 사실적인 렌더링이 가능하지만 이를 게임에 실시간으로 사용하긴 쉽지 않다. 모든 프레임마다 모든 2차광원에 의한 전역조명기법을 적용하여 렌더링을 수행할 경우 매우 많은 양의 계산을 필요로 하기 때문이다. 더구나 보다 사실적인 효과를 얻기 위해 모든 2차 광원에서 가시성 검사를 하는 경우에는 더욱 그러하다.

본 논문에서는 1차 광원이 고정되어 있고 움직이는 물체가 많지 않은 경우에 2차광원의 가시성까지 고려한 효율적인 전역조명 알고리즘을 제안한다. 대부분의 경우에는 물체는 많지만 움직이는 물체는 많지 않고, 주 광원은 고정되므로 이 경우 본 논문에서 제안하는 알고리즘을 이용하여 실시간 전역조명의 구현이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 기술하고, 3장에서는 광원 생성방법을 설명하며 4장에서 그림자 맵 업데이트 방법을 기술한다. 5장에서 1차 광원과 모든 2차 광원을 이용해서 렌더링 하는 방법을 설명한다.

## 2. 관련연구

Keller[1,2]는 1차 조명이 반사된 위치에 2차 광원을 추가해서 전역조명을 표현하는 방법을 제안하였다. 1차광원으로부터 2차광원들을 만들어 낸후 각 2차광원들에 대해서 여러번 그린후 그 결과를 더하는 방법이다. 2차광원의 개수가 많지 않은 경

우 적용할수 있는 방법이다.

Dachsbacher[3]은 1차 광원으로 부터 보이는 지점들을 이미지 형태로 얻어 낸 후 2차 광원을 생성한다. 1차 광원에 의해 생성된 그림자 맵에 2차 광원의 위치, 노말, 밝기 등을 텍셀 정보로 정의한다. 모든 텍셀에 대해 2차 광원을 생성할 경우 2차 광원의 수가 너무 많아지므로 서로 합쳐서 대표적인 2차 광원들을 생성한다. 2차 광원들을 결정하는 방법으로는 미리 정해진 패턴을 사용하거나[3] 사진 트리를 이용하는 방법[4] 등이 있다. 이 방법에서는 계산량을 줄이기 위해 2차광원의 가시성 여부와 상관없이 조명계산을 하는 문제가 있다.

Ritschel[5]는 장면을 점들로 대치해서 2차광원을 위한 그림자 맵의 생성을 빠르게 한 방법을 제안하였다 이 방법은 그림자 맵의 정확도가 높지 않고 삼각형 외에 점 자료를 추가로 구성해서 사용해야 하는 문제가 있다.

Oh[6]는 시점이 고정될 경우 움직이는 물체의 복잡도에 비례하는 가시성 검사 방법을 제안하였다. 1차광원이 고정되어 있고 많은 물체가 고정된 상황에서 간접조명의 가시성을 검사하는데 적용할 수 있다.

Olsson[7,8,9]은 다수의 점광원이 있는 경우에 실시간으로 가시성이 고려된 조명 방법을 제안하였다. 이 방법은 직접광에만 국한된다는 한계가 있다.

## 3. 움직이는 물체의 개수에 비례하는 시간을 가지는 전역조명

본 논문에서는 Dachsbacher의 방법을 개선하여 2차광원의 가시성을 적용한 전역 조명 방법을 제안한다.

본 논문의 방법에서는 움직이지 않는 물체와 움직이는 물체를 구분해서 처리한다. 움직이지 않는 물체위에 2차광원이 있고 움직이지 않는 물체의 조명을 계산할 경우 이들 사이의 가시성에 변화가 생기지 않으므로 이 사실을 활용할 경우 속도 개

선을 기대할 수 있다.

전처리에서는 움직이지 않는 물체들의 2차 광원과 2차광원들에서의 가시성을 계산하고 실시간 처리에서는 움직이는 물체까지 고려한 2차광원과 전역 조명계산을 처리한다. 물체전처리와 실시간 처리에서 움직이지 않는 물체와 움직이는 물체들을 구분하여 전처리와 실시간 처리에서 처리한다.

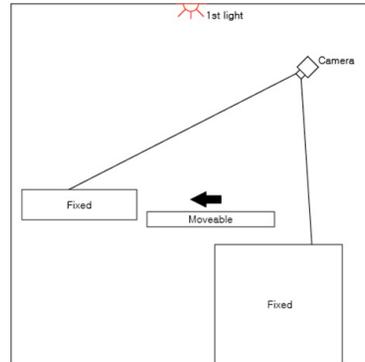
### 3.1 고정된 물체만을 이용한 렌더링

#### 전처리

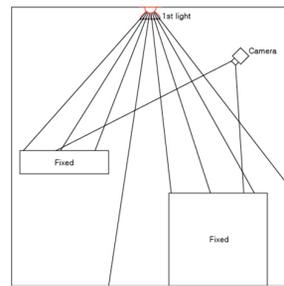
본 논문에서는 장면에 1차 광원 한 개와 많은 움직이지 않는 물체들과 소수 움직이는 물체들이 존재한다고 가정한다.

본논문의 방법에서는 렌더링을 하기 전에 전처리 작업으로 프레임에 따라 움직이지 않는 고정된 물체들에 대한 1차 광원의 그림자 맵(Shadow Map)을 생성하고 Reflective Shadow Map을 이용하여 2차 광원들을 생성한다. 우선 [Fig. 1]처럼 프레임이 변하더라도 움직이지 않는 고정된 물체들과 프레임에 따라 움직이는 물체들을 구분하고, 움직이지 않은 물체만을 대상으로 하는 1차 광원에서의 그림자 맵을 생성한다[Fig. 2]. 1차 광원의 그림자 맵의 모든 픽셀로 2차 광원으로 지정하면 그 수가 너무 많아 계산량이 증가한다[3]. 여러 개의 2차 광원의 위치와 방향의 차이가 크지 않을 경우 여러 개의 2차 광원을 하나의 2차 광원으로 묶어 표현하면 2차 광원의 수를 줄일 수 있다. 본 논문에서는 [4]에서 제안한 사진 트리를 이용하는 방법을 사용하여 다음과 같이 대표 2차 광원을 생성한다. 우선 움직이지 않은 물체만을 대상으로 하는 1차 광원에서의 그림자 맵을 통해 노말 맵을 계산하고, 그림자 맵과 노말 맵을 쿼드트리로 4개의 노드로 나누어서 분산 값을 계산한다. 한 노드의 분산 값이 일정한 값 이상인 경우에는 계속해서 나누고, 일정한 값 미만인 경우에는 그 노드를 2차 광원 노드로 정의한다. 노드 중점의 위치가 2차 광원의 위치로, 중점의 노말은 벡터는 2차광원의 방

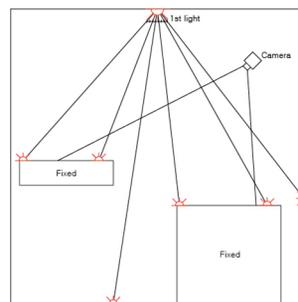
향이 된다[Fig. 3].



[Fig. 1] A scene with light source, moveable objects and fixed objects



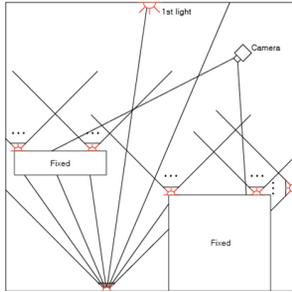
[Fig. 2] Generate shadow map from 1st light source



[Fig. 3] Cluster 2nd light source on fixed objects

Dachsbacher[3]는 속도를 고려하여 렌더링 할 때 가시성 검사를 하지 않는다. 본 논문에서는 보다 사실적인 효과를 얻기 위해 고정된 물체들에 대한 2차 광원에서 가시성 검사를 위해 모든 고정 물체의 2차광원에서 그림자 맵을 생성한다. 이 작

업은 전체의 프레임에 대해 렌더링전에 한번만 하는 작업이므로 각 프레임의 렌더링시점에 수행되지 않는다[Fig. 4].



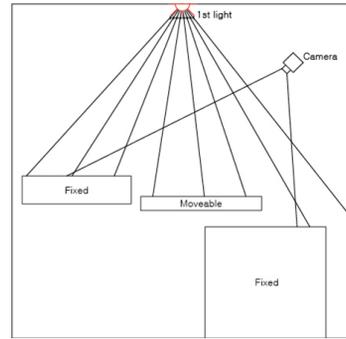
[Fig. 4] Generate shadow map from 2nd lights

### 3.2 렌더링시점의 작업 과정

렌더링시점에는 움직이는 물체들을 고려하여 상기 1차광원의 그림자 맵을 업데이트하고 움직이는 물체들에 대한 2차 광원을 생성한 후, 모든 1차와 2차 광원들을 이용하여 렌더링을 수행한다.

#### 3.2.1 고정된 물체만을 고려한 2차광원에 대한 그림자 맵 갱신

전처리 작업에서는 고정된 물체만을 고려하여 만들어진 1차 광원의 그림자 맵과 고정된 물체상의 2차광원에대한 그림자 맵을 생성했다. 움직이는 물체를 고려하면 렌더링을 위해 프레임마다 그림자 맵을 다시 생성해야 한다. 본 논문에서는 움직이는 물체와 움직이지 않은 물체를 미리 알고 있다고 가정할 때, 그림자 맵을 다시 생성하지 않고 상기 그림자 맵에 움직이는 물체를 업데이트하여 효율적으로 새로운 그림자 맵을 생성한다. 즉, 고정된 물체만을 고려하여 만들어진 상기 그림자 맵의 깊이값과 움직이는 물체의 깊이값을 비교하여 움직이는 물체의 깊이값이 작을 경우에 이 값으로 그림자 맵의 깊이값을 대체한다. 이는 가장 많이 쓰이는 가시성 검사방법인 z버퍼알고리즘으로 수행할 수 있다[Fig. 5].

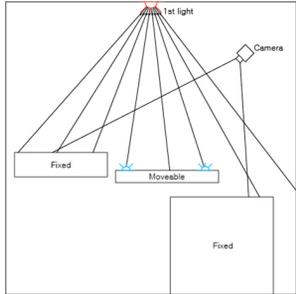


[Fig. 5] Add moveable objects to 1st shadow map

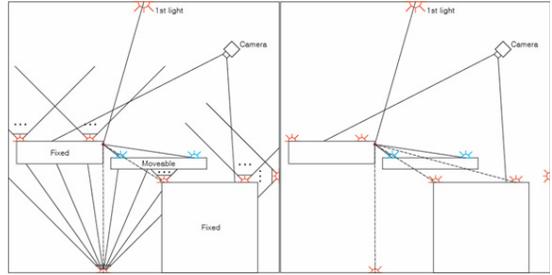
#### 3.2.2 움직이는 물체에 대한 2차광원 생성

움직이는 물체들에 대한 2차 광원을 생성하는 방법은 앞서 3.1에서 기술한 움직이지 않은 물체만을 대상으로 하는 2차 광원을 생성하는 방법과 유사하다. 우선 전처리에서 만든 움직이지 않은 물체만을 대상으로 하는 그림자 맵을 깊이맵 값으로 지정한 상태에서 움직이는 물체들을 그린다. 움직이는 물체들 중 움직이지 않은 물체들 보다 광원에 가까운 물체들만이 그려지게 된다. 이렇게 만들어진 움직이는 물체만을 대상으로 하는 1차 광원에서의 그림자 맵을 통해 노말 맵을 계산하고, 그림자 맵과 노말 맵을 쿼드트리로 4개의 노드로 나누어서 분산 값을 계산한 후, 한 노드의 분산 값이 일정한 값 이상인 경우에는 계속해서 나누고, 일정한 값 미만인 경우에는 그 노드를 2차 광원 노드로 정의한다[그림 7].

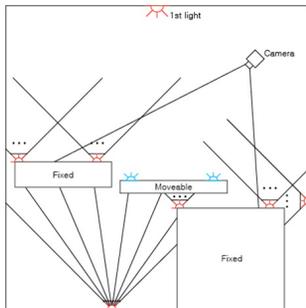
움직이는 물체의 2차 광원에서는 고정 물체의 2차 광원과 달리 가시성 검사를 하지 않는다. 움직이는 물체의 그림자 맵은 움직이지 않은 물체에 비해 그림자 맵 업데이트에 많은 계산을 필요로 하기 때문이다. 그림자 맵 업데이트는 움직이는 물체가 많지 않다는 가정 하에 그림자 맵에 변경된 부분만 수정하는 것이다. 따라서 움직이는 2차광원은 포인트 라이트로 정의하여 렌더링 시 광선 방향과 픽셀의 노말의 각도와 픽셀과 광원의 거리를 참고하여 렌더링 한다[Fig. 7].



[Fig. 6] Cluster 2nd light sources on moveable objects



[Fig. 8] Illuminate with 1st and 2nd light sources



[Fig. 7] Add moveable objects to 2nd shadow map

## 4. 결 과

Inter Core i5 3570 CPU, 8G 메모리와 NVIDIA GeForce GTX 780 그래픽 카드를 탑재한 시스템에서 테스트했다. 그림자 맵을 나누어서 2차 광원 생성 과정은 CPU로 계산했고 렌더링과 그림자 맵 생성 그림자 맵 업데이트 계산을 DirectX 9, shader 3.0으로 계산했다.

실험에는 세 벽으로 구성된 구석에 고정된 모델이 있고 작은 공들이 움직이는 장면을 사용했다.

### 3.2.3 실시간 간접조명 생성 방법

실제 렌더링을 할 때는 계산된 모든 1차, 2차 광원의 영향을 고려해야한다. 고정된 물체의 2차 광원은 움직이는 물체를 그리지 않은 상태에서 생성된 것이다. 움직이는 물체가 추가되면 고정된 물체의 2차 광원이 움직이는 물체에 가려져서 1차광원의 빛이 도착 못 할 수도 있다. 따라서 렌더링하기 전에 고정된 물체의 2차 광원이 1차광원에서 보이는지 확인해야한다.

렌더링 하는 픽셀에 대해 모든 1차 광원과 고정 물체의 2차 광원의 그림자 맵을 이용해서 그림자인지를 확인한다. 움직이는 2차광원은 그림자 맵이 없는 포인트 라이트이므로 빛과 픽셀의 노말 벡터의 각도와 거리만 참고한다[그림 8].

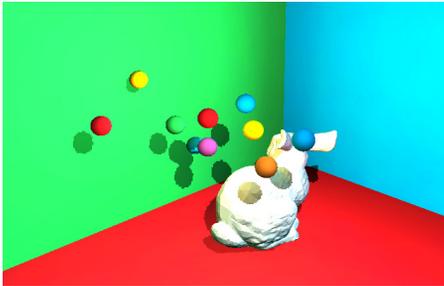
[Table 1] Rendering speed

모델	FPS	고정 물체의 2차 광원	이동 물체의 2차 광원
bunny	50.22	250	0
	15.71	250	13
	10.11	250	41
	23.47	664	0
	7.34	664	13
	4.50	664	41
dragon	30.63	220	0
	15.10	220	13
	8.25	220	41
	14.62	523	0
	6.97	523	13
	4.80	523	41

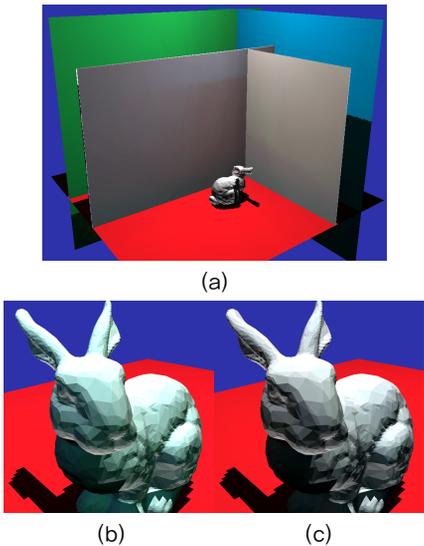
[Table 1]는 정적인 2차광원의 개수와 움직이는 모델 상의 2차광원의 개수에 따른 렌더링 속도를 보여준다. 움직이는 물체상의 광원의 개수가 적은

경우 움직이지 않는 물체 상의 광원의 개수가 늘어 나도 크게 속도 저하가 없는 것을 확인 할 수 있다.

[Fig. 10]은 2차광원이 많이 가려진 장면에서의 결과를 보여준다. reflective shadow map 방법은 2차 광원의 가시성을 검사하지 않기 때문에 파란색 2차광원의 영향이 많이 반영되었다. 반면 본 논문에서 제안하는 방법의 결과는 파란색, 초록색 2차광원의 영향은 가려졌기 때문에 배제되고 가려지지 않은 붉은 색 2차광원들이 많은 영향을 주는 것을 볼 수 있다.



[Fig. 9] Scene with fixed objects and moveable objects



[Fig. 10] A scene with many occluded 2nd light sources)(a), result of reflective shadow map(b), result of our method(c)

## 5. 결 론

본 논문에서는 움직이지 않는 물체들의 광원들에서의 가시성을 재활용함으로써 가시성을 적용한 간접 조명 계산을 실시간에 할 수 있는 방법을 제안하였다. 실험결과 움직이지 않는 물체들의 개수에 렌더링 속도가 영향을 덜 받는다는 것을 확인하였다.

## REFERENCES

- [1] Keller A. "Instant radiosity", Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pp49-56, 1997
- [2] Segovia B. Lehi, J C. Mitanchey, R. Peroche, B. "Bidirectional Instant radiosity", Proceedings of the 17<sup>th</sup> Eurographics conference on Rendering Techniques, pp 389-397, 2006
- [3] Dachsbacher, C.,Stamminger, M. "Reflective shadow maps.", Proceedings of the 2005 symposium on Interactive 3D graphics and games, pp203-231, 2005
- [4] Hyunwoo Ki, Kyoungsu Oh, "A GPU-based light hierarchy for real-time approximate illumination", The Visual Computer, Vol 24 No 7, pp 649-658, 2008
- [5] Ritschel, T., Grosch, T., Kim, M. H., Seidel, H. P., Dachsbacher, C., & Kautz, J. (2008, December). "Imperfect shadow maps for efficient computation of indirect illumination." Transactions on Graphics , Vol. 27, No. 5, Article No 129, 2008.
- [6] Kyoungsu Oh,, Byeongseok Shin, YoungGil Shin "Mobility culling: an efficient rendering algorithm using temporal coherence", The Journal of Visualization and Computer Animation, Vol. 12, No. 3, pp159-166, 2001.
- [7] Olsson, O.Sintorn E. Kämpe V. Billeter M. Assarson U., "Efficient virtual shadow maps

for many lights”, Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp 87-96, 2014.

- [8] Youngjae Chun, Kyoungsu Oh, “A Real-time Soft Shadow Rendering Method under the Area Lights having an Arbitrary Shape”, Journal of Korea Game Society, Vol 14, No 2, pp77-84, 2014
- [9] Hyunwoo Ki, Jihye Lyu, Kyoungsu Oh, “Real-Time Simulation of Single and Multiple Scattering of Light”, Journal of Korea Game Society, Vol 7, No 2, pp 33-44, 2007



정 호 옥(Haoxu Zheng)

2009-2013 연변과학기술대학교 컴퓨터학과 학사  
2013-2015 숭실대학교 미디어학과 석사

관심분야 : 실시간 렌더링, 게임 레벨 자동 생성

---



오 경 수(Kyoungsu Oh)

2001 서울대학교 전기 컴퓨터 공학부 박사  
2001-2002 (주)조이멘트 개발팀장  
2003-현재 숭실대학교 글로벌미디어학부 부교수

관심분야 : 실시간 컴퓨터 그래픽스, 시리얼스 게임

---



백 두 원(Doowon Paik)

1991 Univ of Minnesota 전산학과 박사  
1992-1994 AT&T Bell Labs Member of Technical Staff  
2000-현재 숭실대학교 글로벌미디어학부 교수

관심분야 : 컴퓨터 그래픽스, Computer Aided Design, 병렬처리, 알고리즘

---

— 움직이는 물체의 개수에 비례하는 시간을 가지는 전역조명 알고리즘 —