

## 그림자 아틀라스를 이용한 부드러운 그림자 생성 방법

박선용<sup>o</sup>                      양진석                      오경수

숭실대학교 미디어학과

{aknyong, ispio, oks}@ssu.ac.kr

### Real-time Soft-shadow using Shadow Atlas

Sunyoung Park<sup>o</sup>                      Jinsuk Yang                      Kyoungsu Oh

Dept. of Media, Soongsil University

### 요 약

컴퓨터 그래픽스에서 그림자는 그림자 자체로서 뿐 아니라 장면 내 물체들 간의 거리감에 대한 단서를 제공함으로써 사실감 제고 측면의 매우 중요한 역할을 한다. 그림자를 표현하기 위한 전통적인 방법으로 그림자 매핑이나 그림자 볼륨 등의 기법들이 사용되지만 점광원(point light)을 가정하므로 결과가 자연스럽지 못하다. 반면, 면광원(area light)을 사용할 경우 부드러운 그림자를 생성하므로 좀 더 사실적인 그림자 표현이 가능하지만 광원면(light source surface) 전체에 대한 적분을 요구하기 때문에 계산비용이 매우 비싸다. 이러한 단점을 극복하기 위해 차폐물(occluder)의 광원으로의 역투사(back-projection)나 반영(penumbra)의 크기 계산을 통한 필터링 등 여러 방안들이 소개되었지만 낮은 수준(order)의 근사로 인한 누광(light bleeding)이나 물결현상(ringing effect), 그리고 성능저하 등의 문제가 발생한다. 본 논문에서는 그림자 아틀라스(shadow atlas)를 이용하여 이러한 문제들을 개선하는 방법에 대하여 기술한다.

### Abstract

In computer graphics, shadows play a very important role as a hint of inter-object distance as well as themselves in terms of realism. To represent shadows, some traditional methods such as shadow mapping and shadow volume have been frequently used for the purpose. However, the rendering results are not natural since they assume the point light. On the contrary, an area light can render soft-shadows, but its computation is too burdensome due to integral over the whole light source surface. Many alternatives have been introduced, back-projection of occluder onto the light source to get visibility of light or filtering of shadow boundary by calculating size of penumbra. But they also have problems of light bleeding or ringing effects because of low order approximation, or low performance. In this paper, we describe a method to improve those problems using shadow atlas.

키워드: 그림자 맵, 그림자 아틀라스, 컨벌루션 그림자 맵

Keywords: Shadow Map, Shadow Atlas, CSM(Convolution Shadow Map)

# 1. 서론

그림자는 빛이 물체에 가려져 도달하지 못하는 영역을 의미하며, 그림자를 통해 장면 속 물체들 간의 상대적인 위치 관계 파악 및 사실감을 제고시킨다.

대표적인 그림자 생성 알고리즘으로 그림자 볼륨(shadow volume)[1], 그림자 텍스처(shadow texture)[2], 그림자 맵핑(shadow mapping)[3] 등을 들 수 있다. 이상의 방법들은 모두 구현이 간단하고 빠르기 때문에 게임이나 애니메이션 등에서 많이 사용되어지고 자연 상태에서는 존재하지 않는 점광원(point light source)을 가정하기 때문에 경계가 뚜렷한 그림자(hard shadow)를 생성한다. 하지만 실제 모든 광원은 면적이거나 부피를 가지고 있기 때문에 부드러운 그림자(soft shadow)를 만들어내게 된다.

부드러운 그림자를 위한 기존의 연구들은 면광원의 한 지점에 점광원을 가정하고 만들어진 뚜렷한 그림자(hard shadow)를 부드럽게 필터링 하는 방법과 접영점(接影點, shadow receiving point)에서의 면광원에 대한 가시정도(visibility)를 계산하는 방법의 두 가지로 나눌 수 있다. 첫 번째는 면광원(area light source)과 차폐물(occluder), 그리고 접영면(receiver)의 위치관계에 따라 반영(penumbra)의 크기를 결정하고 면광원의 중심으로부터 생성한 그림자를 반영 크기의 필터로 필터링한다[5][7]. 이러한 방법들은 필터크기가 ‘광원-차폐물-접영점’ 간의 거리에 따라 적응적으로 변하기 때문에 부드러운 그림자를 좀 더 사실적으로 표현할 수 있는 반면 복잡한 구조를 가지는 차폐물의 경우 평균깊이의 계산과 최종 반영처리에 시간이 많이 소요되며 면광원의 한 지점에서 만들어진 그림자를 사용한다는 문제가 있다. Annen et al.[5]은 차폐물의 평균깊이를 효율적으로

계산하기 위해[8]에서 제안한 방법을 응용하여 그래픽스 하드웨어를 이용하는 효율적인 방법을 제안하였지만 구현이 다소 복잡하고 위치에 따라 낮은 차수의 푸리에 근사를 적용할 경우 그림자에 물결현상(ringing effect)이 발생할 수 있다.

두 번째는 셰이딩할 픽셀에 해당하는 지점에서 면광원으로 차폐물을 역투사(back-projection)하고 광원이 보이는 정도를 0-1사이의 값으로 정규화하여 그림자를 표현하는 방법으로 더 정확한 그림자의 표현이 가능하지만 픽셀단위 역투사에 따른 많은 계산량과 면광원의 중심점에 가상카메라를 가정하고 만들어진 하나의 그림자맵을 사용함으로써 발생하는 그림자 겹침(overlapped), 누광(light bleeding) 등의 단점을 가진다[4].

본 논문에서는 위에서 언급한 기존 연구들이 가지는 문제점 중 성능개선에 중점을 둔 부드러운 그림자 생성 방법을 제안한다. 알고리즘은 다음 3단계 과정으로 구성된다.

- 1) 면광원의 중심으로부터 그림자 맵을 만든 후, 각 객체마다 그림자 아틀라스(shadow atlas)를 생성한다.
- 2) CSM(Convolution Shadow Map) 또는 PCF(Percentage Closer Filter) 기법을 이용하여 차폐물(occluder)의 평균깊이를 계산한다.
- 3) 광원-차폐물-접영점의 위치관계를 이용하여 필터(반영)의 크기를 계산하고 그림자 아틀라스를 필터링하여 부드러운 그림자를 표현한다.

이 논문의 2장에서는 주요 알고리즘을, 3장에서는 실험결과에 대해 언급하며 마지막으로 4장에서 결론 및 향후 과제에 대해 기술한다.

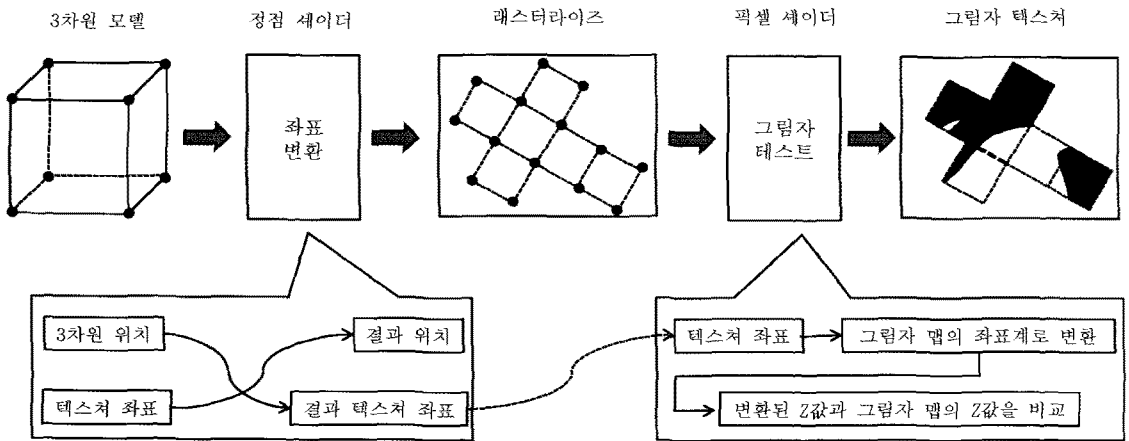


그림 1. 그림자 텍스처 생성 방법. 입력된 3차원 모델링 데이터는 정점 셰이더에서 텍스처 공간으로 좌표 변환한다. 그런 다음 픽셀 셰이더에서 그림자 테스트를 통해 계산된 그림자 정보를 그림자 텍스처에 저장한다.

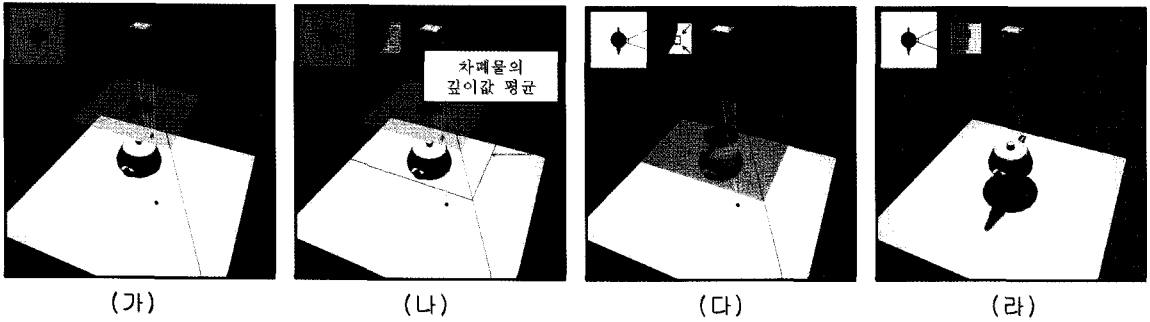


그림 2. 그림자 텍스처를 이용한 부드러운 그림자 생성 방법. (가) 먼저 셰이딩 지점에서 면광원까지 절두체를 만든 후 그림자 맵과 교차영역을 구한다. (나) 교차영역 내부에 존재하는 차폐물의 깊이값 평균을 구한다. (다) 차폐물의 깊이 값 평균과 면광원을 이용해 그림자 텍스처를 필터링할 필터크기를 구한다. (라) (다)에서 구한 필터크기를 이용하여 그림자 텍스처를 필터링한 후 부드러운 그림자를 계산한다.

## 2. 부드러운 그림자 생성 알고리즘

본 절은 최종 그림자 필터링 과정에서 그래픽 하드웨어의 mip-map 기능을 활용할 수 있는 부드러운 그림자 생성 알고리즘을 설명한다. 2.1절에서는 Oh et al.[2]의 그림자 아틀라스 생성 방법을, 2.2절에서는 Annen et al.[5]의 광원에서 차폐물까지의 평균거리 및 반영(필터)의 크기를 계산하는 방법에 대하여 기술한다. 2.3절에서는 그림자 아틀라스 필터링하여 장면에 삽입될 부드러운 그림자를 생성하는 과정을 설명한다.

### 2.1 그림자 아틀라스 생성

그림자 아틀라스란 텍스처 공간상에 펼친 메시의 모든 지점에서 광원에 대한 그림자 검사(shadow test) 결과를 기록한 그림자 텍스처를 의미한다. 게임 등에 사용되어 지는 거의 대부분의 3차원 모델들은 텍스처 좌표를 가지며 그렇지 않은 경우는 Levy et al.[6]의 방법을 이용하여 최적화된 UV 좌표를 생성할 수 있다.

그림자 아틀라스를 만들기 위해서는 먼저 그림자 맵(depth map)을 생성한다. 그림자 맵에는 광원으로부터 장면까지의 최단거리가 저장된다. 메시 표면의 특징지점이 그림자에 포함되는지 여부는 광원까지의 거리 A와 그림자 맵에 저장된 거리 B를 비교하여  $A > B$ 일 경우 그림자(=0)로 판단한다(그림자 검사, shadow test). 일반적인 그림자 매핑(shadow mapping)은 스크린 공간의 가시(visible) 부분에 대해서만 그림자 검사를 수행하지만 그림자 아틀라스에서는 메시의 모든 지점에 대해 검사를 수행한다는 것이 다르다. Oh et al.[2]은 이렇게 만들어진 그림자 아틀라스를 mip-map으로 만들어 그림자 경계에 대한 반앨리어싱(anti-aliasing) 효과를 빠르게 적용하는 방법을 제안하였다.

그림 1은 그림자 아틀라스 생성 과정을 보여준다. 모든 과

정은 셰이더에서 수행되며 장면 내 모든 객체는 각각의 그림자 아틀라스를 가진다.

### 2.2 차폐물의 깊이 평균 계산

Fernando는 [7]에서 삼각 방정식에 의해 반영(penumbra) 크기를 계산하여 그림자 경계를 부드럽게 하는 필터크기로 사용하였다. [7]에서는 그림자 맵에서 광원을 가리는 차폐부분을 찾아내기 위해 주변영역 모든 텍셀에 대해 그림자 검사를 수행하므로 속도가 매우 느리다. 이를 개선하기 위해 [5]에서는 [8]의 CSM(Convolution Shadow Map) 기법을 응용하여 그래픽스 하드웨어를 통해 성능향상을 도모하였다. 본 논문에서는 [5]에서 제안한 방법을 준용하였다.

그림 2는 [5]과 [7]의 일반적인 알고리즘을 보여준다. 먼저 (가)에서는 셰이딩 지점과 면광원과 절두체(frustrum)를 만들어 그림자 맵과 교차하는 영역을 구한다. [7]에서는 교차하는 영역 내의 모든 텍셀에 대해 그림자 검사를 수행하기 때문에 속도가 매우 느린 반면, [5]에서는 구형파(square wave) 형태의 그림자 경계를 (0·1) 푸리에 급수(Fourier Series)로 근사하여  $d(x)$ 와  $z(p)$ 간 비선형적(non-linear)인 그림자 검사식  $f(d(x), z(p))$ 를 (식 1)의 선형적인 식으로 변환함으로써 그림자 맵에 대해 사전 필터링(pre-filtering)이 가능하도록 하였다.

$$f(d(x), z(p)) = \sum_{i=1}^M a_i(d(x))B_i(z(p)) \quad (\text{식 1})$$

단,  $x$ 는 셰이딩할 지점,  $d(x)$ 는 광원으로부터  $x$ 까지의 깊이,  $z(p)$ 는  $x$ 에 대응하는 그림자 맵에서의 텍셀값,  $a_i$ 는 깊이( $d(x)$ )에 따른 계수,  $B_i()$ 는 그림자 맵을 표현하는 기저(basis) 이미지, 그리고  $M$ 은 근사항의 개수를 의미한다.

그림자 맵에 대한 사전 필터링할 수 있다는 것은 밍맵 등의 그래픽스 하드웨어를 이용한 계산이 가능하기 때문에 성능 면에서 많은 이점을 얻을 수 있다. 다만 위 근사식에서 항(term) 개수가 적을 경우 그에 따른 시각적 문제가 발생할 수 있으나 근사결과를 그림자에 직접 적용하는 [8]과 달리 여기서는 필터 크기 계산을 위한 간접적인 요소로 사용되기 때문에 실제 구현에서 크게 문제되지 않았다.

푸리에 급수를 이용하여 차폐영역의 평균 깊이를 구하는 식의 세부유도 과정은 부록(Appendix)에 포함하였다.

### 2.3 그림자 아틀라스 필터링 및 최종 렌더링

이번 절에서는 (식 6)에 의해 계산된 차폐영역의 평균 깊이와 면광원의 기하정보를 이용하여 최종적으로 그림자 아틀라스에 적용할 필터(반영)의 크기를 구하는 방법에 대하여 설명한다. 면광원은 정사각형을 가정하며 필터크기  $f_w$ 는 다음의 삼각방정식으로 유도된다.

$$f_w = \frac{\Delta}{d} \cdot \frac{(d - z_{avg})}{z_{avg}} \quad (\text{식 } 2)$$

$$l = \log(r * \frac{f_w}{2z_n \tan(fovy/2)}) \quad (\text{식 } 3)$$

위 식에서  $\Delta$ 는 면광원 너비,  $d$ 는 세이딩될 지점의 깊이 값,  $z_n$ 은 근평면(near plane)까지의 거리,  $fovy$ 는 시야각,  $r$ 은 그림자 맵의 해상도, 그리고  $f_w/2z_n \tan(fovy/2)$ 는 텍스처 공간으로 투영된 필터의 크기를 나타낸다.

최종 단계에서는 (식 2)에 의해 계산된 크기의 필터로 그림자 아틀라스를 필터링(평균) 한 후 모델에 적용하면 된다. 필터링은 [7]에서와 같이 필터의 크기에 해당하는 텍셀을 모두 읽어 평균을 계산하거나 필터 크기를 밍맵 레벨로 환산( $l$ )하여 삼선형(tri-linear) 필터인 밍맵을 이용할 수 있다. 이때 3차원 공간에서 텍셀이 차지하는 영역의 크기가 그림자 아틀라스와 그림자 맵에서 서로 다르기 때문에 계산된 밍맵 레벨을 그대로 적용할 수 없다. 우리는 3차원 공간에서 두 텍스처의 텍셀이 차지하는 영역의 비율을 얻기 위해 [9]에서 소개된 방법을 사용하였다. 즉, 밍맵의 각 부표면(sub-surface)에 자신의 레벨 값을 가지는 텍스처로 장면을 광원시점에서 렌더링하면 각 텍셀 값이 우리가 원하는 비율( $l_{lod}$ )을 가지게 되고 (식 3) 및 (식 4)와 같이 보정된 밍맵 레벨( $l_{corrected}$ )과 텍셀의 개수  $N_{corrected}$ 를 얻을 수 있다.

$$l_{corrected} = l + l_{lod} \quad (\text{식 } 4)$$

$$N_{corrected} = (2^{l_{corrected}})^2 \quad (\text{식 } 5)$$

### 3. 실험결과

본 논문의 구현에 있어 그래픽 API는 DirectX 9.0, 하드웨어는 Nvidia Geforce GTX 480 그래픽 카드와 Intel CPU 3.0GHz를 사용하였다.

그림 3과 표 1은 본 논문에서 제안하는 방법과 [7]의 PCSS(Percentage Closer Soft Shadow) 및 SAT(Summed Area Table)의 방법으로 그림자를 처리한 결과와 프레임율(frame per second)을 비교하고 있다. 그림 3의 (a)에서 밍맵핑의 특성으로 인해 렌더링 품질은 다소 문제를 보이거나 성능(프레임율)에서는 획기적으로 향상되었음을 확인할 수 있다.

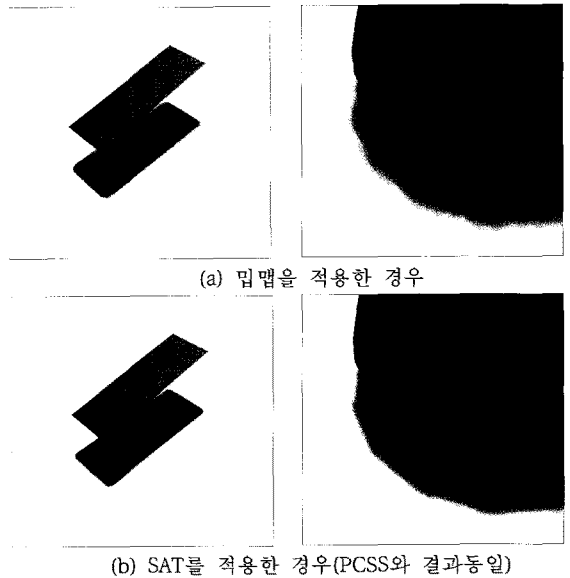


그림3. 그림자 아틀라스를 밍맵을 사용하여 필터링 한 경우와 SAT를 사용하여 필터링한 경우의 비교, (a)의 경우 복잡한 기하구조(오른쪽)에서 다소 문제를 보일 수 있으나 성능 면에서는 많은 개선을 확인할 수 있다.

필터링 방식	PCSS	SAT	밍맵
프레임율	27~45	132~140	170~176

표 1. 필터링 방식별 성능(performance)비교 (frame per second)

1) 표 1에 기재된 프레임율은 구현 시 근평면(near plane), 시야각(FOV), 그리고 광원의 크기(light width) 등의 기하설정에 많은 영향을 받으므로 여기서는 비율만을 참조하면 된다.

#### 4. 결론 및 향후과제

본 논문은 그림자 아틀라스를 이용한 부드러운 그림자 생성 방법을 설명하고 있다. 논문에서 제안한 방법은 면광원과 차폐물, 세이딩할 위치의 기하적 위치관계에 따라 다른 크기의 반영을 가지는 부드러운 그림자를 그림자 아틀라스를 사용해 구현함으로써 하드웨어 가속이 가능하도록 하였다. 반영의 크기를 계산하는데 가장 중요한 광원으로부터 차폐물까지의 평균 거리는 [5]의 방법을 그대로 적용하였으며 그림자 아틀라스 최종적인 필터링은 mip매핑을 적용하였다. 그 과정에서 그림자 맵과 그림자 아틀라스의 각 텍셀이 3차원 공간상에서 차지하는 비율을 보정하기 위해 레벨이 반영된 mip(mip-mapped texture)을 활용하는 방안을 제안하였다.

본 논문에서 제안하는 방법은 성능에 있어 많은 향상을 얻었지만 mip매핑으로 인한 화질상의 문제가 반론으로 제기될 수 있다. 또한 차폐물의 깊이계산에 [5]에서 제안한 방법을 수정 없이 사용하였는데 이 역시 푸리에 급수의 근본적인 문제인 낮은 차수의 근사에서 나타날 수 있는 물결무늬효과가 깊이변화가 심한 곳에서 발생할 수 있다. 추후 보완해야 할 문제로 지속적인 연구가 필요하다.

#### Appendix

다음은 푸리에 급수를 이용하여 차폐영역의 평균 깊이를 구하는 식의 세부유도 과정이다.

$$\bar{f}(d(x), z(p)) \approx \frac{1}{2} + 2 \sum_{k=1}^M \frac{1}{c_k} \sin[c_k(d(x) - z(p))], \quad (\text{식 } 6)$$

$\bar{f}()$ 는 그림자 검사  $f()$ 의 보수(complementary)함수이며  $c_k$ 는  $\pi(2k-1)$ 이다. 그러면 차폐부분의 평균깊이  $z_{avg}$ 는 (식 7)로 유도되어 진다.

$$\begin{aligned} z_{avg} &\approx \frac{1}{1-s_f(x)} [w_{avg} * (\frac{1}{2} + \sum_{k=1}^M \frac{2}{c_k} \sin[c_k(d(x) - z)])z](p) \\ &\approx \frac{1}{1-s_f(x)} [w_{avg} * \frac{\pi}{2} + \frac{2}{c_k} \sum_{k=1}^M \sin(c_k d(x))(w_{avg} * z \cos(c_k z)) - \\ &\quad \frac{2}{c_k} \sum_{k=1}^M \cos(c_k d(x))(w_{avg} * z \sin(c_k z))](p) \end{aligned}$$

(식 7)

위 식에서  $s_f(x)$  그림자 테스트 결과 값,  $w_{avg}$ 는 평균을 구하는 박스필터를 의미한다.

#### 감사의 글

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2011-0012214)

#### 참고 문헌

- [1] Crow F. C, "Shadow algorithms for computer graphics", Computer Graphics (Proceedings of SIGGRAPH '77), pp. 242-248, 1977.
- [2] Hyunwoo Ki, Taegy Ryu, Kyoungsu Oh, "An Efficient Shadow Texture Algorithms with Shadow Maps", In Proceedings of Computer Graphics International, 2008.
- [3] Williams L, "Casting curved shadows on curved surfaces.", Computer Graphics (Proceedings of SIGGRAPH '78), pp. 270-274, 1978.
- [4] Guennebaud G., Barthe L., Paulin M., "Real-time soft shadow mapping by backprojection", In Proceedings of Eurographics Symposium on Rendering, pp. 227-234, 2006.
- [5] Thomas Annen, Zhao Dong, Tom Mertens, Philippe Bekaert, Hans-Peter Seidel, Jan Kautz, "Real-time, all-frequency shadows in dynamic scenes", Proceedings of ACM SIGGRAPH 2008.
- [6] Bruno Levy, Sylvain Petitjean, Nicolas Ray, Jerome Mailhot, "Least squares conformal maps for automatic texture atlas generation", Proceedings of ACM SIGGRAPH 2002, Vol. 21, pp. 362-371, 2002.
- [7] Fernando, R. "Percentage-Closer Soft Shadows", In ACM SIGGRAPH 2005 Sketches, 35, 2005.
- [8] Annen, T., Mertens, T., Bekaert, P., Seidel, H.-P., and Kautz, J. "Convolution Shadow Maps", In Rendering Techniques 2007 (Proc. of EGSR), pp. 51-60, 2007.
- [9] Fernando, R., Fernandez, S., Bala, K., and Greenberg, D. P, "Adaptive shadow maps", In Proceedings of ACM SIGGRAPH 2001, Computer Graphics Proceedings, Annual Conference Series, pp. 387-390, 2001.

## 〈저자 소개〉



박선용

- 1995년 동국대학교 컴퓨터공학 학사
- 2006년 연세대학교 경영정보학과 석사
- 2010년 숭실대학교 미디어학과 박사
- 관심분야: 실시간 렌더링, 증강현실, 시각화



양진석

- 2009년 숭실대학교 미디어학부 학사
- 2009년~현재 숭실대학교 미디어학과 석박사통합과정
- 관심분야: 실시간 컴퓨터 그래픽스, 게임 프로그래밍



오경수

- 1994년 서울대학교 계산통계학과 학사
- 1996년 서울대학교 전산학과 석사
- 2001년 서울대학교 전기컴퓨터공학부 박사
- 2001년~2002년 (주)조이멘트 개발팀장
- 2003년~현재 숭실대학교 글로벌미디어 학부 부교수
- 관심분야: 실시간 컴퓨터 그래픽스, 게임 프로그래밍