
그림자 생성을 위한 개선된 PCF 기법

유영중* · 최진호*

An Improved PCF Technique for The Generation of Shadows

Young-Jung Yu* · Jin-Ho Choi*

요 약

그림자는 사실적인 3차원 장면의 렌더링에 없어서는 안되는 중요한 요소다. 3차원 장면에서 그림자로 인해 물체 사이의 거리를 파악할 수 있고 3차원 효과를 보일 수 있다. 그림자를 생성하는 방법은 크게 이미지 기반 방법(Image Based Method)과 물체 기반 방법(Object Based Method)으로 구분된다. 물체 기반 방법은 정확한 그림자를 계산하는 것이 가능하지만 계산 속도가 장면의 복잡도에 의존하기 때문에 실시간 그림자 생성에는 적합하지 않다. 물체 기반 방법과는 달리 이미지 기반 방법은 그림자 계산 속도가 장면의 복잡도에 의존하지 않기 때문에 빠른 속도로 인해 많이 사용되는 방법이다. 그러나 다른 여러 이미지 영역의 방법과 마찬가지로 이미지 기반 방법은 앨리어싱(Aliasing) 문제가 나타나는 단점이 있다. 이 문제를 극복하기 위한 여러 방법이 제안되었으며 PCF(Percentage Closer Filtering)도 그 중의 한 방법이다. PCF를 이용한 그림자 생성 기법은 렌더맨에 구현되면서 유명해졌는데, 결과적으로 그림자 경계에서 보이는 앨리어싱 문제를 어느 정도 해결하였다. 그러나 사용되는 필터의 크기에 따라 계산 속도가 문제가 될 수 있는 단점이 있다. 본 논문에서는 기존의 PCF 방법을 개선하여, 거의 유사한 결과를 보이면서 PCF보다 더 빠른 시간안에 그림자를 생성하는 기법을 제안한다.

ABSTRACT

Shadows are important elements for realistic rendering of the 3D scene. We cannot recognize the distance of objects in the 3D scene without shadows. Two methods, image-based methods and object-based methods, are largely used for the rendering of shadows. Object based methods can generate accurate shadow boundaries. However, it cannot be used to generate the realtime shadows because the time complexity depends on the complexity of the 3D scene. Image based methods which are techniques to generate shadows are widely used because of fast calculation time. However, this algorithm has aliasing problems. PCF is a method to solve the aliasing problem. Using PCF technique, antialiased shadow boundaries can be generated. However, PCF with large filter size requires more time to calculate antialiased shadow boundaries. This paper proposes an improved PCF technique which generates antialiased shadow boundaries similar to that of PCF. Compared with PCF, this technique can generate antialiased shadows in less time.

키워드

그림자, PCF, 앨리어싱, 필터, 그림자 맵

I. 서 론

3차원 환경에서 그림자의 존재유무는 렌더링된 장면

이 사실적으로 보이는지 여부를 결정하는 중요한 요소라 할 수 있다. 따라서 지금까지 많은 연구자들이 사실적인 그림자를 생성하기 위한 연구를 하였다. 그림 1은 바

탁위에 떠있는 상자를 보여준다. 그림 1(a)는 그림자가 없는 경우로 상자가 바닥위에 놓여 있는지 떠 있는지 여부를 판단하기가 어렵다. 그림 1(b)와 같이 그림자가 있는 경우 상자의 공간상의 위치를 파악하는 것이 가능하다.

그림자를 생성하는 방법은 크게 객체 기반 방법과 이미지 기반 방법의 두 가지로 구분된다. 객체 기반 방법의 그림자 생성 기법[1]은 그림자의 경계 부분을 정확히 계산할 수 있다는 장점을 가진다. 그러나 그림자를 계산하는 시간이 장면의 복잡도에 좌우된다는 단점을 가진다. 따라서 장면이 복잡하면 복잡할수록 그림자를 계산하는 시간이 늘어나고, 따라서 실시간 그림자 계산이 어려워진다. 컴퓨터 그래픽스에서는 정확한 그림자를 생성하는 것도 중요하지만 빠른 시간안에 그림자를 생성하는 것도 중요한 문제이므로 객체 기반 방법은 계산 시간의 문제가 심각한 요소라 할 수 있다. 이에 반해 이미지 기반 방법[2]은 장면의 복잡도와는 상관없이 렌더링되는 장면의 크기에 따라 계산 시간이 결정되는 장점을 가지고 있다. 따라서 최근에는 이미지 기반의 그림자 생성에 관한 많은 연구가 이루어지고 있다. 그러나 이미지 기반의 방법은 생성되는 그림자의 경계 부분에서 앨리어싱이 발생하는 문제가 있다. Reeves 등[3]은 앨리어싱 문제를 해결하기 위한 방법으로 PCF를 제안했다. PCF는 현재 렌더맨에 적용되어 사용중이며, 그림자의 경계에서 발생하는 앨리어싱 문제를 해결하고, 추가로 부드러운 그림자 영역을 생성해준다. PCF가 이미지 기반 방법

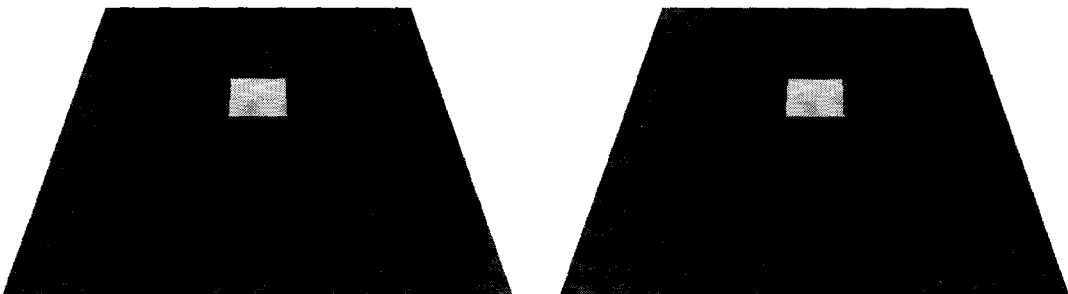
의 문제점인 앨리어싱 문제를 다루고 있지만 완벽히 해결한 것은 아니다. 또한 PCF 방법에서는 그림자 계산을 위해 필터를 사용하는데, 사용하는 필터의 크기에 따라 그림자 계산에 필요한 시간이 증가하게 된다.

본 논문에서는 기존의 PCF를 이용한 그림자 생성 방법을 개선하여, 기존의 PCF를 사용한 그림자와 유사한 결과를 보이면서 좀 더 빠른 시간안에 그림자를 계산하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 이미지 기반 그림자 생성 기법을, 그리고 3절에서는 PCF 기법에 대해 설명한다. 4절에서는 본 논문에서 제안하는 그림자 생성 방법에 대해 설명하고 5절에서 실험 결과를 보여준다. 마지막으로 6절에서 결론을 맺는다.

II. 이미지 기반 그림자 생성

이미지 기반 그림자 생성 기법은 Williams[2]가 최초로 제안한 방법이다. 이 기법의 알고리즘은 2단계로 이루어진다. 먼저, 3D 장면이 조명 시점에서 렌더링되고, 렌더링된 각 픽셀의 깊이 값이 저장된다. 이 저장된 값을 그림자 맵(shadow map)이라 한다. 이 값은 나중에 각각의 픽셀이 그림자 영역에 포함되는지 여부를 결정하기 위해 사용된다. 다음으로, 장면이 카메라 시점에서 렌더링되고, 렌더링된 각 픽셀은 조명 좌표계로 변환된다. 카



(a) 그림자가 없는 경우

(b) 그림자가 있는 경우

그림 1. 그림자 유무에 따른 장면의 사실성
Fig 1. The reality of a 3D scene according to the shadow

메라 좌표계에서 조명 좌표계로 변환된 각 픽셀의 깊이 값은 조명 시점에서 계산되고 저장된 일치하는 픽셀의 깊이 값과 비교된다. 조명 시점에서 계산되고 저장된 깊이 값이 z_l 이고 카메라 시점에서 계산되고 변환된 픽셀의 깊이 값이 z_c 라 하면, 다음 식과 같이 계산된다.

$$\begin{cases} z_l < z_c & \text{그림자 영역} \\ z_l = z_c & \text{경계선} \\ z_l > z_c & \text{조명 영역} \end{cases}$$

Segal등[4]은 텍스처매핑을 이용해서 빠르게 이미지 기반 그림자를 생성하는 기법을 제안했다. 이 기법은 조명과 카메라의 각 시점에서 텍스처 맵을 계산하고, 계산된 텍스처맵을 변환하여 비교함으로써 빠르고 쉽게 이미지 기반의 그림자를 생성할 수 있도록 하는 구현 기법이다. 이 구현 방법은 현재 OpenGL의 확장된 텍스처 기능을 이용해서 구현 가능하다[5]. Fernando등[6]은 Williams[2]가 제안한 그림자맵을 계층적인 구조로 변환하여 그림자 경계 부근에서 발생하는 앨리어싱을 최소화하는 방법을 제안했다. Comba등[7]은 기존의 PCF와 함께, 각 픽셀에 여러개의 깊이 값을 동시에 저장하는 기법을 사용하여 기존의 PCF의 속도를 향상시키는 방법을 제안했다.

본 논문에서는 기존의 PCF 기법을 기반으로 그림자를 생성하는 속도를 향상시키는 방법을 제안한다. 다음 절에서는 기존의 PCF 기법에 대해 설명한다.

III. PCF 기법

Williams[2]가 제안한 이미지 기반 그림자 생성 기법은 두 가지 앨리어싱 문제를 가지고 있다. 하나는 깊이 맵을 생성할 때, 발생하고 나머지 하나는 저장된 깊이 맵을 샘플링할 때 발생한다. 전자는 확률적인 샘플링을 이용해 깊이 맵을 생성하여 해결할 수 있다[8]. PCF는 후자의 문제를 해결하기 위해 제안된 기법이다.

그림 2는 이전의 필터링 방법과 PCF 방법을 비교해 보여주는 그림이다. 현재 선택된 픽셀이 그림자 영역에 포함되는지 여부를 결정하기 위해, 선택된 픽셀과 이웃한 픽셀들의 깊이 값을 일치하는 위치에 있는 픽셀들의 깊이 값과 비교를 수행한다. PCF 이전의 필터링 방법

에서는 필터링되는 전체 픽셀의 깊이 값들의 평균을 계산한 후 비교를 수행한다. 그림 2(a)를 보면 각각의 평균 값이 22.9와 49.8이다. 따라서 선택된 픽셀은 그림자 영역으로 간주된다. 그림 2(a)에서 보는 바와 같이 PCF 이전의 필터링 기법을 사용할 경우, 전체 영역은 그림자 영역 또는 그림자 영역이 아닌 두 경우로만 구분이 가능하며, 그림자 경계 부근에서 앨리어싱이 발생한다. PCF는 이전의 필터링 방법에서 취하던 순서를 반대로 하여 계산한다. 즉, 필터에 속하는 각 픽셀에 대해 비교를 먼저 수행하고, 각각의 비교된 결과를 모아 필터링을 수행하는 것이다. 그림 2(b)를 보면 필터에 속하는 각 픽셀의 비교 결과는 0 또는 1이 되며, 최종 결과는 비교 결과의 평균 값이 된다. 따라서 PCF 방법의 결과는 이전의 필터링 방법이 전체 영역을 그림자 영역과 그림자 아닌 영역으로만 구분하는 것과는 달리 다양한 강도의 그림자 영역으로 구분되는 결과를 얻을 수 있으며, 이 결과를 이용하면 앨리어싱 문제를 해결할 수 있다.

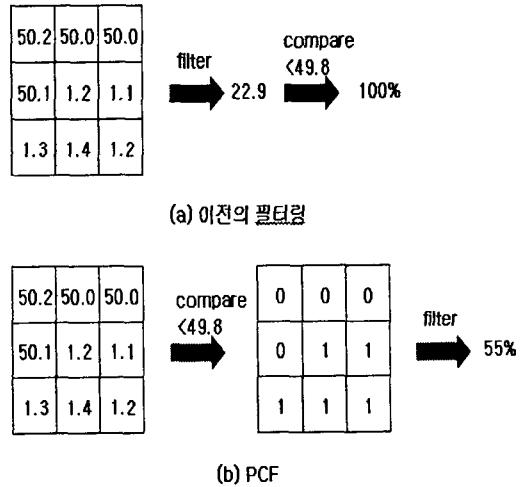
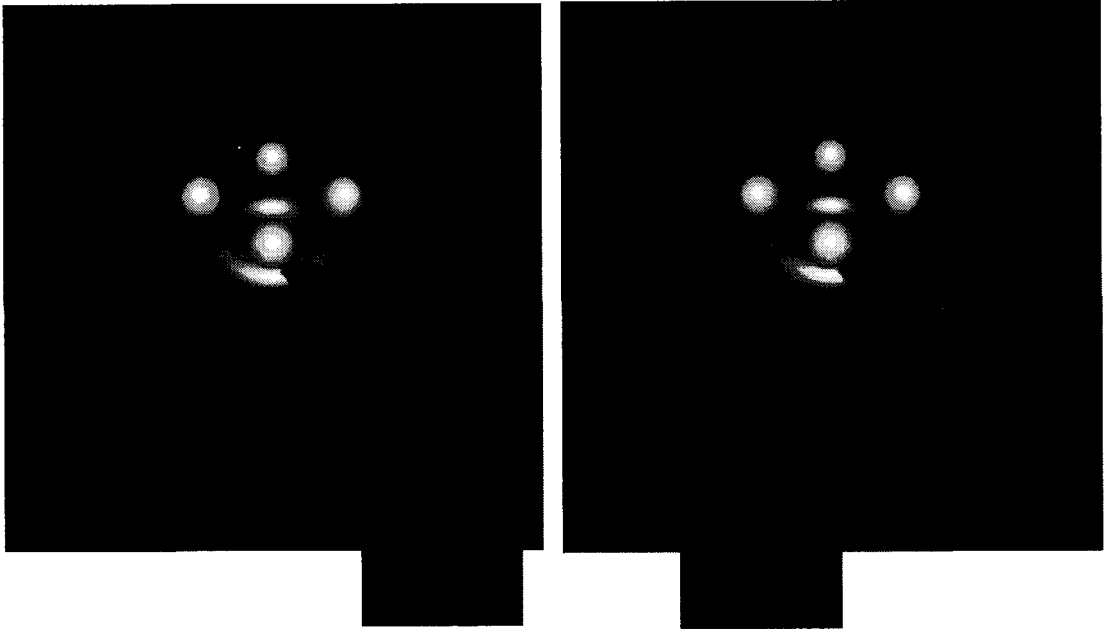


그림 2. 이전 필터링 방법과 PCF 방법의 비교
Fig 2. The comparison between previous filtering method and PCF method

그림 3은 PCF를 사용하지 않은 경우와 사용한 경우의 결과를 보여준다. 그림 3(a)는 PCF를 사용하지 않은 경우의 결과이며, 그림자의 경계 부근에서 앨리어싱이 나타남을 알 수 있다. 그림 3(b)는 PCF를 사용한 경우의 결과로, 그림자 경계 부근이 부드럽게 처리된 것을 볼 수 있다.



(a) PCF 방법을 사용하지 않은 경우

(b) PCF 방법을 사용한 경우

그림 3. 필터링 방법의 결과 비교

Fig 3. The comparison of results of two filtering methods

IV. 개선된 PCF를 이용한 그림자 생성

PCF는 이미지 기반 그림자 생성 기법에서 그림자 경계 부근에 나타나는 앨리어싱을 제거해주는 장점을 가진다. 그 결과로 부드러운 그림자 영역이 만들어진다. 그러나 PCF를 위해 사용되는 필터의 크기에 따라 계산 속도가 문제가 될 수 있다. 필터의 크기가 클수록 더 부드러운 그림자 영역이 생성될 수 있지만 그림자를 계산하기 위해 필요한 계산시간을 더 많이 필요로 한다. 최종적으로 보여지는 이미지의 질 또한 중요하지만, 계산 속도 또한 렌더링에서 중요한 요소이기 때문에 계산에 필요한 시간을 줄이는 것은 컴퓨터 그래픽스에서 중요한 문제로 볼 수 있다.

PCF 방법은 카메라 시점에서 계산된 결과를 조명 시점으로 변환한 후, 조명 시점의 모든 픽셀들과 비교를 수행한다. 이 과정에서 필터가 사용된다. 필터의 크기를 F_s , 이미지 크기를 I_s 라 하면, PCF 방법에 의해 그림자를 계산하기 위해 필요한 계산 시간은 $O(I_s \times F_s)$ 이

다. 따라서, PCF 방법에서 계산 시간은 이미지 크기와 필터 크기에 의존한다.

그러나 그림자를 계산할 때 PCF 방법과 같이 이미지의 모든 위치에서 그림자 영역인지 여부를 계산할 필요는 없다. 장면에서 그림자 영역과 그림자가 아닌 영역을 구분하는 것은 경계선만 알 수 있다면 쉬운 문제가 된다. 본 논문에서 제시하는 알고리즘의 기본적인 생각은 PCF를 위해 사용되는 필터가 모든 픽셀에 적용될 필요가 없다는 것에 착안하고 있다. 즉, 앨리어싱이 나타나는 그림자 경계 부근에 대해서만 필터링을 적용하고 나머지 픽셀에 대해서는 PCF를 적용하지 않음으로 계산 시간을 줄일 수 있다는 것이다. 경계선을 추출하기 위해 이미지 전체를 한번 검사한다. 경계선을 구성하는 픽셀의 수를 B_s 라 하자. 제시하는 알고리즘에서 그림자를 계산하기 위해 요구되는 시간 복잡도는 $O(I_s + B_s \times F_s)$ 가 된다. 전체 장면에서 경계선 픽셀의 수는 많은 부분을 차지하지 않기 때문에 PCF 방법에 비해 계산 시간을 절약할 수 있다.

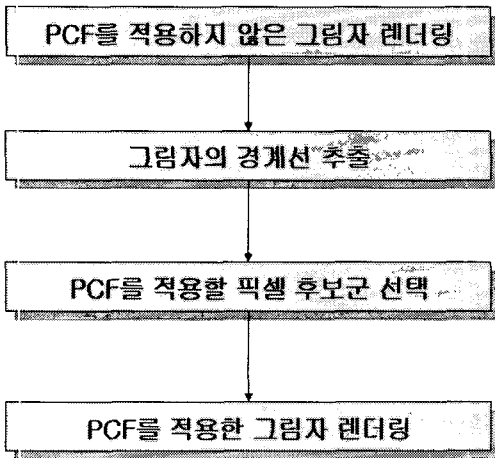


그림 4. 개선된 PCF 알고리즘 단계
Fig 4. Improved PCF algorithm steps

그림 4는 본 연구에서 제안하는 개선된 PCF 알고리즘을 위한 단계이다. 먼저 PCF 방법을 적용하지 않은 그림자를 렌더링하고, 렌더링된 그림자의 경계를 추출한 후, PCF를 적용할 후보 픽셀들을 선택한다. 마지막으로 선택된 후보 픽셀들에 PCF를 적용해서 그림자를 렌더링한다. 그림 5는 본 논문에서 제안하는 알고리즘을 적용한 결과 그림이다. 그림 5(a)는 필터링 없이 그림자를 생성한 결과로, 경계 부근에서 앨리어싱이 발생한 것을 볼 수 있다. 그림 5(b)는 그림자의 경계선을 추출한 결과이고, (c)는 추출된 경계선에서 PCF를 적용할 후보 픽셀들을 선택한 결과로 경계선이 확장된 것을 볼 수 있다. 그림 5(d)는 마지막 결과 그림이다. 그림자 경계선 부근이 부드럽게 처리된 것을 볼 수 있다.

그림 5의 전체 이미지의 크기는 512×512 이고, 3×3 필터를 사용하였다. PCF 방법에 의해 그림자를 생성할 경우, 그림자 테스트를 위해 대략 $512 \times 512 \times 9$ 번의 연산이 필요하다. 그러나 그림 5의 경우 경계선상의 픽셀이 대략 1000개정도이고, 필터링을 위해 선택되는 픽셀의 수는 9000정도이므로, 9000×9 정도의 테스트 연산만으로 PCF를 사용한 결과와 유사한 결과를 얻을 수 있었다.

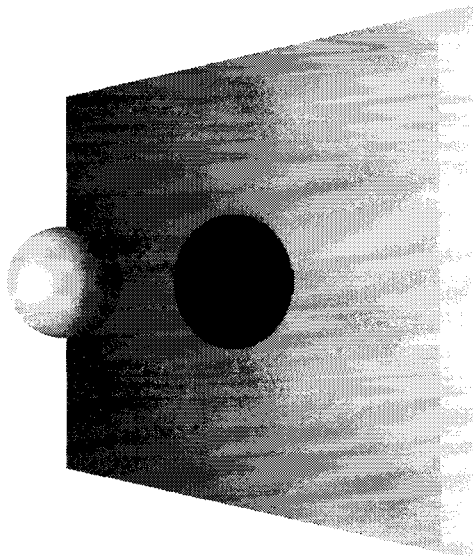
V. 실험 결과

본 논문에서는 제안한 그림자 생성 방법의 실험을 위해 펜티엄 환경에서 마이크로소프트 비주얼 스튜디오 6.0과 OpenGL을 사용하였다. 전체 과정은 하드웨어 처리과정 없이 소프트웨어로만 계산되었다.

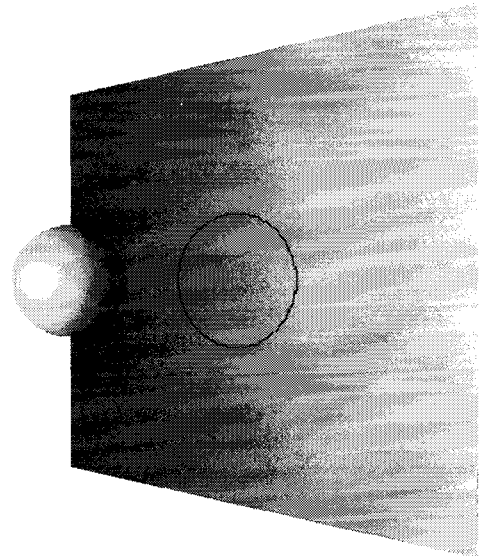
그림 6은 512×512 의 이미지 크기와 3×3 필터를 사용하여 생성한 결과이다. 그림 6(a)는 그림자의 경계선을 추출한 결과이고, (b)는 마지막 렌더링 결과이다. 그림 7은 서로 다른 크기의 필터를 사용할 때의 결과를 보여준다. 그림 7(a)는 5×5 의 크기의 필터를 사용했을 때의 결과이고, (b)는 7×7 크기의 필터에 대한 결과이다. (a)의 결과 더 큰 필터를 사용한 (b)의 결과에서 그림자 경계선 부근의 앨리어싱이 많이 제거되었음을 볼 수 있다. 그림 8은 7×7 크기의 필터를 사용하여 PCF 방법과 본 논문에서 제안한 방법의 렌더링 결과를 보여준다. 두 가지 방법 모두 렌더링 결과는 유사함을 볼 수 있다.

VI. 결 론

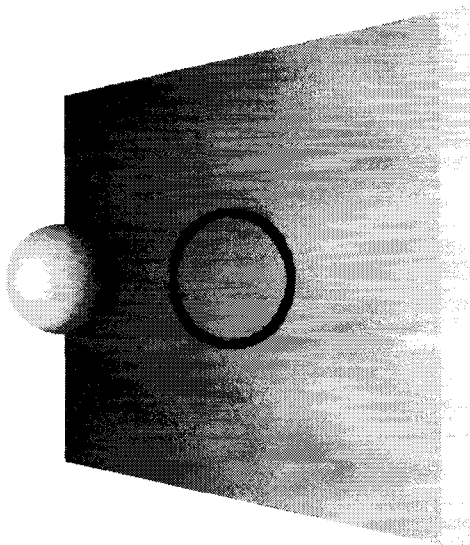
이미지 기반의 그림자 생성 기법은 그림자 경계 부근에서 발생하는 앨리어싱의 문제가 있지만 구현의 용이성과 속도로 인해 최근들어 가장 많이 사용되는 기법이 되었다. 본 논문에서는 이미지 기반 그림자 생성의 앨리어싱 문제를 해결하기 위한 방법으로 제시된 PCF 기법을 개선하여 유사한 결과를 보이면서, 더욱 빠르게 그림자를 계산하는 방법을 제안하였다. 본 논문에서 제안한 방법을 사용한다면 PCF와 유사한 결과를 얻으면서 렌더링시에 더욱 빨리 그림자를 계산함으로써 실시한 렌더링에 기여할 수 있을 것으로 생각한다.



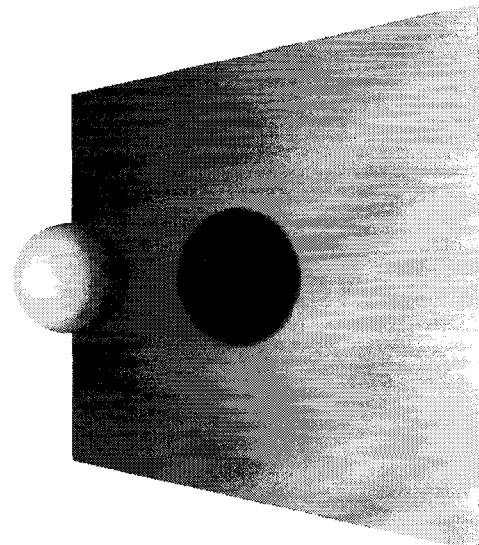
(a) 필터링이 없는 경우



(b) 경계선 추출

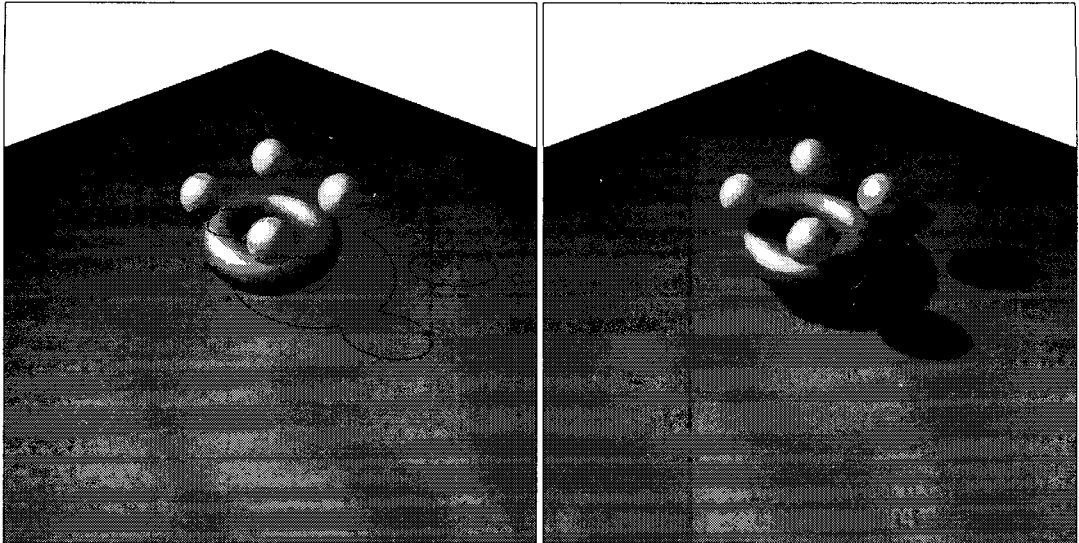


(c) 경계선 확장



(d) 렌더링 결과

그림 5. 제안된 알고리즘의 단계별 결과
Fig 5. The result per step of proposed algorithm

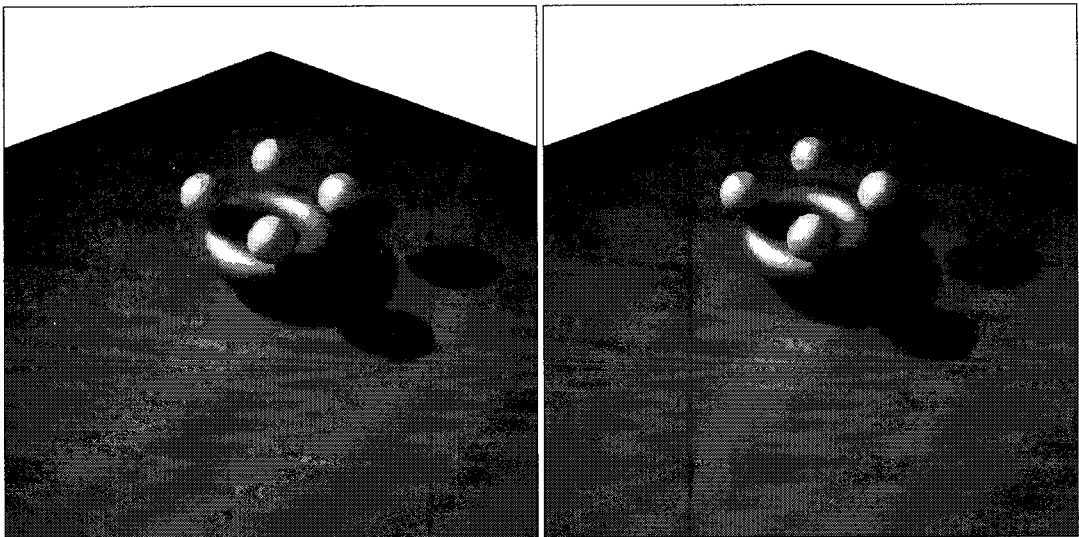


(a) 경계선 추출

(b) 렌더링 결과

그림 6. 추출된 경계선과 렌더링 결과(이미지크기: 512×512 , 필터크기: 3×3)

Fig 6. Extracted boundaries and the rendering result(image size: 512×512 , filter size: 3×3)

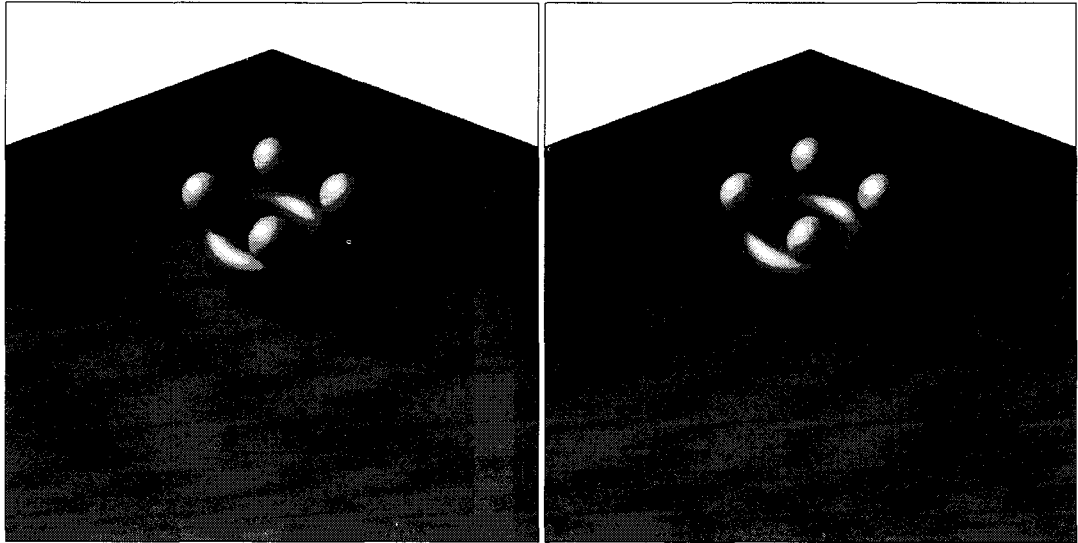


(a) 5×5

(b) 7×7

그림 7. 필터 크기에 따른 렌더링 결과

Fig 7. The rendering results with the different filter sizes



(a) PCF

(b) 제안한 방법

그림 8. PCF 방법과 제안한 방법의 결과 비교(필터크기: 7×7)

Fig 8. The comparison of results between PCF method and the proposed method(filter size: 7×7)

참고문헌

[1] F. C. Crow, Shadow Algorithms for Computer Graphics, Siggraph'77, pp:242-248, July, 1977.
 [2] L. Williams, Casting Curved Shadows on Curved Surfaces, Siggraph'78, pp:270-274, Aug, 1978.
 [3] W. T. Reeves, D. H. Salessin, and R. L. Cook, Rendering Antialiased Shadows with Depth Maps, Siggraph'87, pp:283-291, July, 1987.
 [4] M. Segal, C. Korobkin, R. Widenfelt, J. Foran, and P. Haerberli, Fast Shadows and Lighting Effects using Texture Mapping, Siggraph'92, pp:249-252, July, 1992.
 [5] Shadow Mapping: Casting Curved Shadows on Curved Surfaces, "http://www.paulsprojects.net/tutorials/smt/smt.html".
 [6] R. Fernando, S. Fernandez, K. Bala, and D. Greenberg, Adaptive Shadow Maps, Siggraph'2001, pp:387-390, Aug, 2001.
 [7] C. A. Pagot, J. L. D. Comba, and M. Neto, Multiple- Depth Shadow Maps, SIACG2004, pp:308-315, Oct, 2004.
 [8] R. L. Cook, Stochastic Sampling in Computer Graphics, ACM Transactions on Graphics, Vol:5(1), pp:51-72, Jan, 1986.

저자소개



유 영 중(Yu Young Jung)

1996 부산대학교 전자계산학과 이학사
 1998 부산대학교 전자계산학과 이학 석사

2002 부산대학교 전자계산학과 이학박사
 2002~ 부산외국어대학교 근무
 ※ 관심분야: 컴퓨터 애니메이션, 시뮬레이션



최 진 호(Jin Ho Choi)

1985 부산대학교 전자공학과 공학사
 1987 KAIST 전기 및 전자공학과 공학 석사

1992 KAIST 전기 및 전자공학과 공학박사
 1996~ 부산외국어대학교 근무
 ※ 관심분야: VLSI 설계