

개선된 PCF 기법을 이용한 그림자 생성

유영중*

*부산외국어대학교

Rendering Antialiased Shadows with Improved PCF

Young-Jung Yu*

*Pusan University of Foreign Studies

E-mail : yjyu@pufs.ac.kr

요 약

그림자는 사실적인 장면의 렌더링에 없어서는 안되는 중요한 요소다. 그림자를 생성하는 방법 중 한가지인 이미지 기반 방법은 빠른 속도로 인해 많이 사용되는 방법이다. 그러나 다른 여러 이미지 영역의 방법과 마찬가지로 앨리어싱 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 어느정도 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. 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 boundary can be generated. However, PCF with large filter size requires more time to calculate antialiased shadow boundary. This paper introduces an improved PCF technique which generates antialiased shadow boundary similar to that of PCF. Compared with PCF, this technique can generate antialiased shadows in less time.

키워드

그림자, PCF, 앨리어싱, 필터

1. 서 론

그림자의 존재유무는 렌더링된 장면이 사실적으로 보이는지를 결정하는 중요한 요소라 할 수 있다. 따라서 지금까지 많은 사람들이 사실적인 그림자를 생성하기위한 연구를 하였다. 그림자를 생성하는 방법은 크게 객체 기반 방법과 이미지 기반 방법의 두 가지로 나뉘어진다. 객체 기반 방법의 그림자 생성 기법[1]은 양질의 그림자를 생성할 수 있는 장점을 가진다. 그러나 장면의 복잡도에 따라 그림자를 계산하는 시간이 좌우된다는

단점을 가진다. 양질의 그림자를 생성하는 것도 중요하지만 빠른 시간안에 그림자를 생성하는 것도 중요한 문제이므로 객체 기반 방법은 계산 시간의 문제가 심각한 요소라 할 수 있다. 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. Reeves등[3]은 앨리어싱 문제를 해

결하기위한 방법으로 PCF를 제안했다. PCF는 현재 렌더만에 적용되어 사용중이며, 그림자의 경계에서 발생하는 앨리어싱 문제를 해결하고, 추가로 부드러운 그림자 영역을 생성해준다. PCF가 이미지 기반 방법의 문제점인 앨리어싱 문제를 어느 정도 해결하지만, PCF에서 사용하는 필터의 크기에 따라 그림자 계산에 사용되는 시간이 증가하게 된다. 본 논문에서는 기존의 PCF를 이용한 그림자 생성 방법을 개선하여, 기존의 PCF를 사용한 그림자와 유사한 결과를 보이면서 더욱 빠른 시간안에 그림자를 계산하는 방법을 제안한다.

II. 이미지 기반 그림자 생성

이미지 기반 그림자 생성 기법은 Williams[2]가 최초로 제안한 방법이다. 이 기법의 알고리즘은 2 단계로 이루어진다. 먼저, 장면이 조명의 관점에서 렌더링되고, 렌더링된 각 픽셀의 깊이 값이 저장된다. 이 저장된 값을 그림자 맵(shadow map)이라 한다. 이 값은 나중에 그림자 여부를 결정하기 위해 사용된다. 다음으로, 카메라 관점에서 장면이 렌더링되고, 렌더링된 각 픽셀은 조명 좌표계로 변환되고, 변환된 각 픽셀의 깊이 값이 일치하는 이전에 저장된 깊이 값과 비교된다. 이전에 저장된 깊이 값이 z_i 이고 변환된 픽셀의 깊이 값이 z_c 라 하면, $z_i \leq z_c$ 인 경우, 일치하는 픽셀은 그림자 영역으로 처리된다.

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 구현기법이다. 이 구현 방법은 현재 OpenGL의 확장된 텍스처 기능을 이용해서 구현가능하다[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. 다음절에서 기존의 PCF 기법에 대해 설명하고 4절에서 본 논문에서 제안하는 방법을 설명할 것이다. 그리고 5절에서 실험결과를 통해 기존의 PCF 방법과 본 논문에서 제안한 방법의 결과가 유사함을 보여주며, 마지막으로 결론을 맺을 것이다.

III. PCF

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

그림 1은 이전의 필터링 방법과 PCF 기법을 보여주는 그림이다. 현재 선택된 픽셀의 그림자

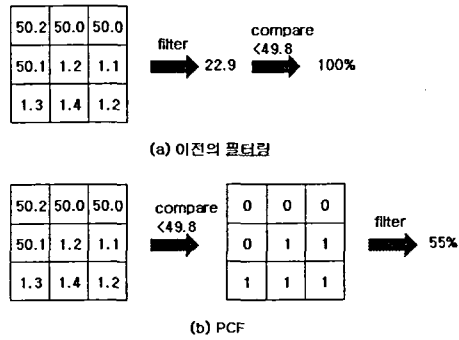


그림 1. 이전의 필터링 방법과 PCF

여부를 결정하기위해 일치하는 위치에 있는 픽셀의 깊이 값과 비교를 수행할 때, 이전의 필터링 방법에서는 필터링되는 전체 픽셀의 깊이 값들의 평균을 계산한 후, 비교를 수행한다. 따라서 이전의 필터링 기법을 사용할 경우, 그림자 영역 또는 아닌 두 경우로만 구분이 가능하며, 그림자 경계 부근에서 앨리어싱이 발생한다. PCF는 이전의 필터링 방법에서 취하던 순서를 반대로하여 계산한다. 즉, 필터에 속하는 각 픽셀에 대해 비교를 먼저 수행하고, 각각의 비교된 결과를 모아 필터링을 수행하는 것이다. 결과는 이전의 필터링 방법과는 달리 다양한 결과를 얻을 수 있으며, 이 결과를 이용하면 앨리어싱 문제를 해결할 수 있다.

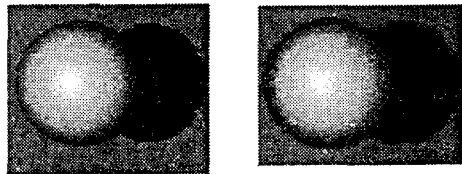


그림 2. PCF를 이용한 결과: 왼쪽 이미지는 PCF 없이 생성한 결과이고, 오른쪽 이미지는 5×5필터를 사용한 결과이다.

그림 2는 PCF를 사용하지 않은 경우와 사용한 경우의 결과를 보여준다. 그림 2에서 왼쪽 이미지는 PCF를 사용하지 않은 경우의 결과이며, 그림자의 경계 부근에서 앨리어싱이 나타남을 알 수 있다. 오른쪽 그림의 경우, 그림자 경계 부근이 부드럽게 처리된 것을 볼 수 있다.

IV. 개선된 PCF를 이용한 그림자 생성

PCF는 이미지 기반 그림자 생성 기법에서 그림자 경계 부근에 나타나는 앨리어싱을 최소화해주는 장점을 가진다. 그러나 PCF를 위해 사용되는 필터의 크기에 따라 계산 속도가 문제가 될 수 있다. 계산 속도는 렌더링에서 중요한 요소이기 때문에 계산에 필요한 시간을 줄이는 것 또한 중요한 문제로 볼 수 있다. 본 연구의 아이디어는 PCF를 위해 사용되는 필터가 모든 픽셀에 적용될 필요가 없다는데 착안하고 있다. 즉, 앨리어싱이 나타나는 그림자 경계 부근에 대해서만 필터링을 적용하고 나머지 픽셀에 대해서는 PCF를 적용하지 않음으로 계산 시간을 줄일 수 있다는 것이다.

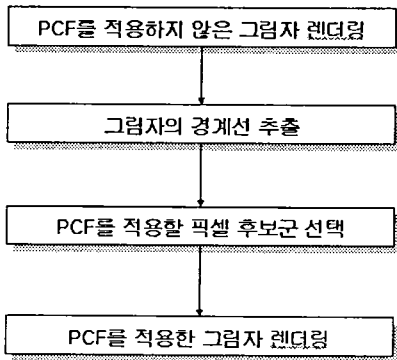


그림 3. 개선된 PCF 알고리즘 단계

그림 3은 본 연구에서 개선된 PCF 알고리즘을 위한 단계이다. 먼저 PCF 적용하지 않은 그림자를 렌더링하고, 그림자의 경계를 추출한 후, PCF를 적용할 후보 픽셀들을 선택한다. 마지막으로 선택된 후보 픽셀들에 PCF를 적용해서 그림자를 렌더링한다. 그림 4는 알고리즘을 적용한 결과 그림이다. 전체 이미지의 크기는 512×512이고, 따라서 5×5 크기의 PCF를 사용할 경우 그림자 테스트를 위해 512×512×25번의 연산이 필요하다. 그러나 그림 4의 경우 경계선상의 픽셀이 대략 1000개정도이고, 필터링을 위해 선택되는 픽셀의 수는 9000정도이므로, 9000×25정도의 테

스트 연산만으로 PCF를 사용한 결과와 유사한 결과를 얻을 수 있다.

IV. 실험 결과

그림 5와 그림 6은 각각 기존의 PCF와 개선된 PCF를 사용해서 생성한 결과 이미지이다. 두 결과는 모두 5×5의 필터를 사용하였으며, 두 결과 이미지가 앨리어싱을 줄이는 유사한 결과를 보임을 알 수 있다. 본 논문에서는 실험을 위해 전체 과정을 하드웨어 처리과정 없이 OpenGL을 이용해 소프트웨어적으로 처리하였으며, 초당 프레임율은 계산하지 않았지만 그 속도차이를 확인할 수 있었다.

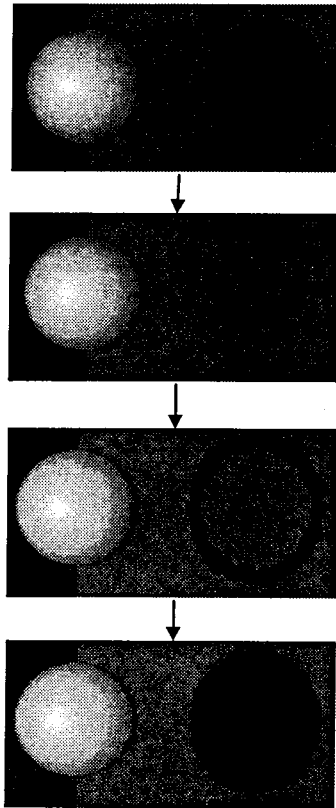


그림 4. 개선된 PCF를 이용한 처리 과정

V. 결론

이미지 기반의 그림자 생성 기법은 그림자 경

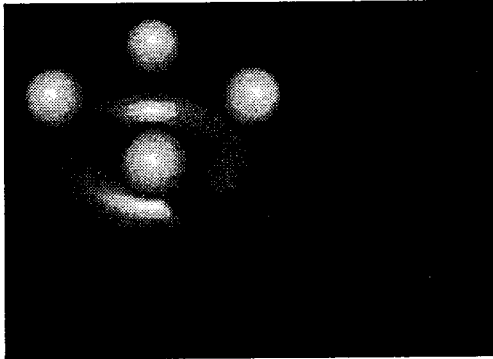


그림 5. 기존의 PCF(5×5)를 이용한 결과



그림 6. 개선된 PCF를 이용한 결과

계 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 실시간 렌더링에 기여할 수 있을 것으로 생각한다.

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.