이방성으로 필터링된 밉맵의 보간을 이용한 실시간 필드심도 렌더링

Real-Time Depth of Field Rendering Using Anisotropically Filtered Mipmap Interpolation

이성길, Sungkil Lee*, 김정현, Gerard Jounghyun Kim**, 최승문, Seungmoon Choi***

요약 본 논문은 핀홀 카메라 모델에 의해 렌더링한 컬러와 깊이 이미지의 후처리에 의한 실시간 필드심도 렌더링 방법을 제안한다. 필드심도 렌더링은 최근의 향상에도, 큰 스케일의 블러링에 필요한 계산 때문에 실용적으로 사용되지 못해 왔다. 본 방법은 이방성 가우시안 필터로 생성된 밉맵 이미지들을 비선형으로 보간하여 필드심도 효과에 필요한 블러링을 수행한다. 모든 계산 과정은 GPU로 가속되어, 안정적이고 확장 가능한 실시간 수행 성능을 확보한다. 또한, 후처리 방식의 두 가지 결점인 강도 누출과 블러링 불연속성을 이방성 가우시안 필터와 블러링 정도를 부드럽게 하여 제거한다. 본 방법은 뛰어난 실시간 성능과 함께 고품질의 필드심도 효과를 생성한다.

Keywords: Depth of Field Rendering, Mipmap Interpolation, Anisotropic Filtering.

1. 서론

필드심도(Depth of Field)는 렌즈 시스템에서 초점이 맞아 또렷하게 보이는 깊이(depth)의 범위를 칭한다. 실제 렌즈 시스템은 유한한 조리개(aperture)를 가지고 있어 3차원의 점이 CoC(Circle of Confusion, 착란원)라고 불리는 원형의 블러링(blurring) 영역에 맺히게 된다. [1] 이러한 필드심도 효과는 사실적인 이미지 품질 및 인간의 깊이 인식에 도움이 된다고 알려져 있다. 일반적인 컴퓨터 그래픽스는 핀홀 카메라 모델을 사용하므로, 필드심도 효과를 얻기 위해서는 별도의 이미지 처리나 렌더링 과정이 필요하고, 많은 방법들이 제안되었다.

기존의 필드심도를 얻기 위한 방법들 중, 가장 정확한 방법은 렌즈모델을 이용하여, 유한한 조리개에서 샘플링된 여러 시점을 핀홀 모델을 이용하여 렌더링한 후, 블렌딩 하는 것이다 (multi-pass rendering). [2-3] 이 방법은 사실적인 필드심도 효과를 생성하지만, 이미지 한 장의 렌더링 시간이 몇 분에서 몇 시간에 이르므로 주로 컴퓨터 애니메이션을 위한 오프라인 렌더링에 이용되어 왔다. 이러한 성능을 개선하기 위해 제안된 다른 주류는 핀홀 모델로 렌더링한 컬러, 깊이 이미지를 후처리(post-filtering)하여 필드심도 효과를 근사하는 방법이다. [4-7] 근래에 GPU(Graphics Processing Unit) 기반을 둔 대부분의 실시간 필드심도 렌더링 방법들은 이 방식을 채택하고 있다.

후처리 기반의 방법들은 다중 렌더링(multi-pass rendering)에 비해 빠른 시간 안에 필드심도 효과를 나타낼 수 있지만, 여전히 실시간에 수행되기에는 부족하다. 또한, 정확한 방법들에서 없는 강도의 누출(intensity leakage)과 전경 불연속성(foreground blurring discontinuity)과 같은 결점(artifact)을 보인다. 강도의 누출은 배경 픽셀의 블러링을 위한 필터링(filtering)에서, 초점이 맞은 영역의 색상이 배경으로 새어 나가는 현상을 가리키며, 초점 주변의 영역들이 날카롭게 표시되지 못하여 눈에 거슬리게 한다. [6-7] 전경 블러링의 불연속성은 초점이 맞은 물체보다 앞쪽에 있는 물체들의 경계가, 실제 필드심도 이미지의 부드러운 경계와 달리 날카롭게 끊어지는 것을 가리키며, 역시 이미지 품질을 저하시키는 큰 요인이다. [6]

본 논문은 2007년 과학재단 특정기초연구(R01-2006-000-11142-0)의 지원에 의하여 연구되었음.

*주저자 : POSTECH 컴퓨터공학과 박사과정 e-mail: yskill@postech.ac.kr
**교신저자 : 고려대학교 정보통신학부 교수 e-mail: gjkim@korea.ac.kr
***공동저자 : POSTECH 컴퓨터공학과 교수; e-mail: choism@postech.ac.kr

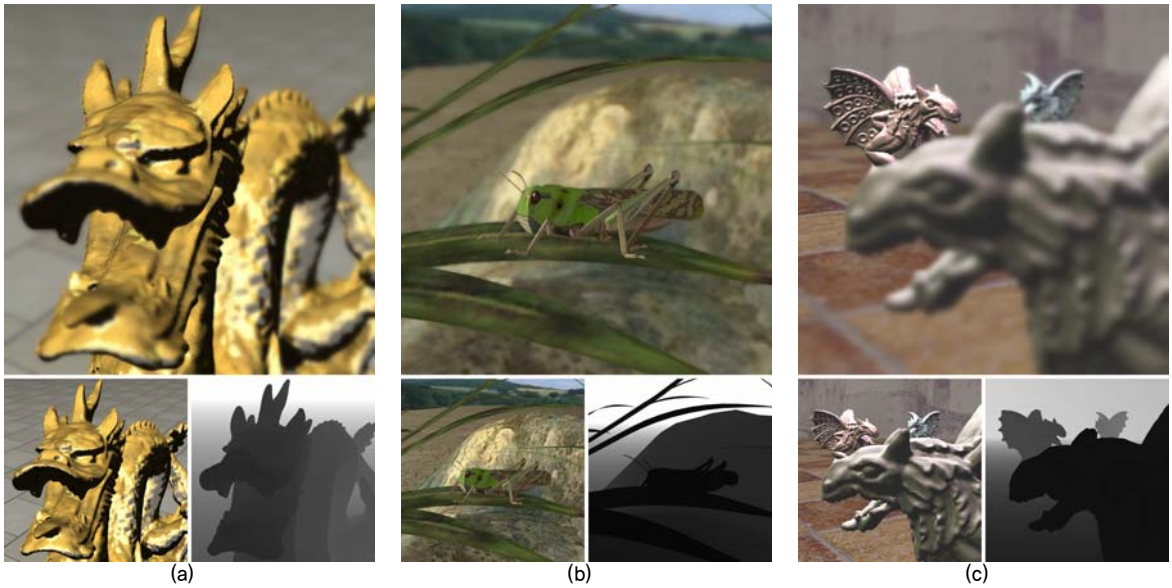


그림 1. 본 논문에 제시된 방법을 이용하여 생성된 필드심도 이미지. 각 이미지 아래 작은 이미지는 핀홀 모델을 이용하여 렌더링된 컬러(왼쪽)와 깊이(오른쪽)를 나타낸다. 각 이미지는 드래곤의 귀(a), 메뚜기(b), 가운데 Gargoyle(c)에 초점을 맞추고 있다. 본 논문의 방법은 강도 누출과 전경 불연속성 없이 고품질의 필드심도 이미지를 생성한다.

이에, 본 논문은 기존 후처리 방법들의 두 가지 결점을 개선하여 만족할 만한 고품질을 가지는 동시에, 품질을 보장하는 기존 방법 중 가장 빠른 방법(예, Scheuermann [5])보다 우수한 성능를 가지는 필드심도 렌더링 방법을 제안한다. 렌더링 속도는 밉맵의 비선형 보간(nonlinear interpolation)을 이용하여 확연히 개선되고, 강도 누출과 전경 불연속성은 이방성 필터링과 더불어 픽셀에서의 블러링 정도를 부드럽게 하여 제거된다. 그림 1은 본 논문에서 제안하는 방법을 이용하여 렌더링한 이미지 결과의 예이다.

2. 일반적인 필드심도 모델

본 논문에서는 필드심도 렌더링에 필요한 블러링의 정도를 결정하기 위해 Potmesil[1]에 의해 제안된 필드심도 모델을 사용 한다 (그림 2). 이 모델을 이용하여, 핀홀 모델에 의해 렌더링된 깊이 이미지 픽셀의 CoC 지름을 구하고, 이를 블러링 정도(DoB: degree of blurring)에 대응 시킨다.

그림 2에서 3차원 위치 **P**에 있는 객체(object)의 이미지 깊이 V_d와 초점이 맞은 위치/깊이 $\mathbf{P}_f$/ V_f는 렌즈의 특성과 함께 다음과 같이 쓸 수 있다.

$$V_d = \frac{Fd}{d-F} \quad \text{and} \quad V_f = \frac{Fd_f}{d_f - F} \tag{1}$$

수식에서 F는 렌즈의 초점거리(focal length)이고, d와 d_f는 렌즈로부터 물체와 초점이 맞은 영역까지의 거리이다. 그림 2에서 이미지 평면 상의 CoC, C_r은 수식 (1)과 유효 렌즈 크기 E를 이용하여 수식 (2)와 같이 나타낼 수 있다.

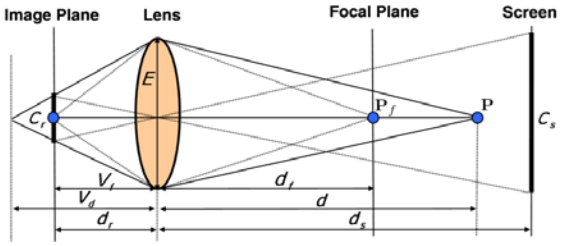


그림 2. Potmesil의 필드심도 모델[1]

$$C_r = |V_d - V_f|\frac{E}{V_d} = \left(\frac{EF}{d_f - F}\right)\frac{|d - d_f|}{d} \tag{2}$$

C_r과의 비례관계를 이용하여, 스크린의 CoC 지름 C_s는 렌즈에서 이미지 평면과 스크린까지의 거리 d와 d_s를 이용하여 다음과 같이 쓸 수 있다.

$$C_s = \frac{d_s}{d_r}C_r \tag{3}$$

지름 C_s의 착란원은 해당 픽셀에 대해서 스크린 상에서 블러링되어야 할 영역을 가리킨다.

3. 밉맵의 비선형 보간을 이용한 실시간 필드심도 렌더링 방법

밉맵(mipmap)은 입력 이미지로부터 1/4씩 크기를 줄여 나가며 렌더링한 텍스쳐(texture) 이미지의 집합(set)이다. 이렇게 축소된 밉맵 이미지를 입력 이미지의 크기로 확대했을 때, 그 이미지는 필드심도에 쓰일 수 있는 블러링된 효과를 보인다. 각 픽셀의 블러링 정도에 해당하는 밉맵의 레벨

을 확대하여 블러링 효과를 픽셀별로 따로 얻는 것이 본 논문에서 제안하는 필드심도 렌더링 방법의 주안점이다.

본 논문의 필드심도 렌더링 방법은 3 단계로 이루어진다. 먼저 주어진 핀홀 컬러 이미지에 대해 밉맵을 생성 후, 각 픽셀에서의 CoC는 연속적인 밉맵 레벨과 대응된다. 밉맵은 불연속적으로 정의되므로, 연속적인 밉맵 레벨과 가까운 두 불연속 레벨의 이미지를 비선형(nonlinear)으로 보간(interpolation)하여 최종 필드심도 이미지를 생성하게 된다. 본 방법은 기존의 방법(미리 블러링한 이미지와 입력 이미지 두 장을 블렌딩 [5])과 달리, CoC에 해당하는 블러링의 정도를 연속 범위(continuous domain)에서 정확히 얻을 수 있을 뿐 아니라, 그 수행 성능 또한 매우 우수한 장점을 지닌다. 각 단계의 자세한 과정은 아래에 자세히 기술한다.

3.1 밉맵 피라미드의 생성

일반적으로, 밉맵은 GPU 자체의 내장된 능력을 이용하여 생성될 수 있지만, 이렇게 생성된 밉맵은 확대하였을 때, 그 품질이 많이 저하된다. 이에, 본 방법은 이미지 확대 품질을 향상하기 위해 이미지의 다운 샘플링(down-sampling)을 GPU상의 프래그먼트 쉐이더(fragment shader)에서 가우시안 필터링을 하여 수행한다.

$I(l)$와 $\mathcal{G}$를 레벨 l에서의 밉맵 이미지와 3×3의 가우시안 컨벌루션 연산자(Gaussian convolution operator)라고 할 때, 입력 이미지 $I(0)$로부터 시작하여 다음 레벨(1/4 크기)의 이미지는 다음과 같이 생성된다.

$$I_{\mathbf{p}}(l) = \mathcal{G} * I_{\mathbf{p}}(l-1) = w_G \sum_{\mathbf{q}\in\Omega} G(\mathbf{q}-\mathbf{p}) I_{\mathbf{q}}(l-1) \tag{4}$$

수식에서 $\mathbf{p}$는 중심 픽셀 위치, $\mathbf{q}$는 $\mathbf{p}$에 인접한 8 픽셀이고, $G(x)$는 x에서의 가우시안 오프셋(offset), w_G는 정규화 계수((normalization weight)를 가리킨다.

3.2 CoC와 밉맵레벨의 연관

생성된 밉맵을 필드심도의 블러링에 이용하기 위해서는 스크린상의 CoC(C_s)는 밉맵 피라미드의 레벨과 관계되어야 한다. 여기서의 난제는 C_s가 연속적으로 정의되는 블러링의 정도인데 반해, 밉맵은 불연속적으로 정의되는 점이다. 이 둘을 연관시키기 위해서, 먼저 CoC의 반지름을 가우시안 확률 분포(Gaussian distribution)를 보이는 IDF(intensity distribution function [8])의 표준편차(standard deviation)와 연관시킨다. σ 를 IDF의 standard deviation이라고 할 때, C_s와 σ 의 비례 관계는 단위 길이 당 픽셀 수 DPI를 이용하여 다음과 같이 정의 된다.

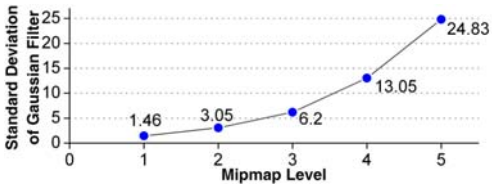


그림 3. 최소 RMS 오차를 가지는 가우시안 필터의 표준편차(standard deviation)과 밉맵 레벨의 관계.

$$\sigma = DPI\frac{C_s}{2} \tag{5}$$

다음은 σ 를 밉맵 레벨에 대응시키는 것이다. 저자는 σ 의 표준편차를 가지는 가우시안 필터링과 밉맵의 특정 레벨을 확대한 이미지를 정량적으로 비교하여, 각 밉맵 레벨에 대해서 최소의 RMS(root-mean-square) 오차(error)를 가지는 σ 를 구하였다 (그림 3). 본 비교에는 "Lena" "Baboon" 이미지가 사용되었다. 찾아진 결과로부터 σ 와 밉맵 레벨 m 사이의 관계를 다음 수식과 같이 근사할 수 있다.

$$\sigma = \frac{3}{2}\cdot 2^{m-1} \tag{6}$$

수식 (6)을 이용하여 σ 과 m 사이 일반적인 관계는 다음과 같이 정의될 수 있다.

$$\begin{aligned} m &= \begin{cases} 0 & \text{if } k_\sigma\sigma < 0.5 \\ 1+\log_2(k_\sigma\sigma) & \text{otherwise} \end{cases} \quad \text{and} \\ l &= \lfloor m \rfloor, \end{aligned} \tag{7}$$

k_σ는 수식 (6)에서 유도된 비례 상수이며, 본 논문에서는 2/3을 쓴다. 수식에서 $k_\sigma\sigma < 0.5$ 은 가우시안 필터링의 표준편차가 픽셀의 블러링에 영향을 주지 못함을 의미한다. 수식 (7)에서 m은 연속적인 값이므로, 연속적인 범위에서 블러링 결과를 생성하기 위해서, m에 인접한 두 레벨 l과 l+1에 해당하는 두 이미지를 보간함이 필요하다.

3.3 밉맵의 비선형 보간

픽셀 $\mathbf{p}$에 대해 m이 결정되면, 수식 (7)을 이용하여 다음과 같이 직관적인 비선형 보간 관계를 추론할 수 있다.

$$I_{\mathbf{p}}(m) = I_{\mathbf{p}}(l)^{1-\beta} I_{\mathbf{p}}(l+1)^{\beta} \tag{8}$$

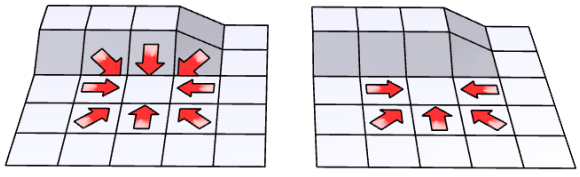
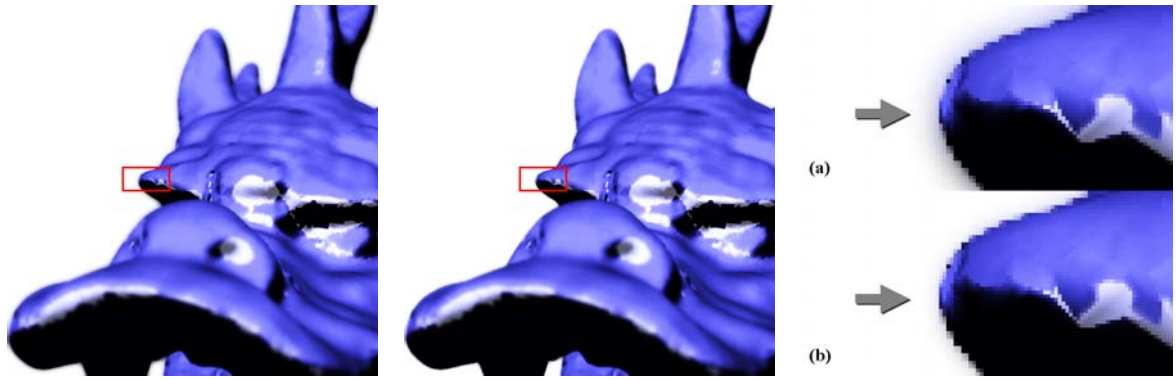


그림 4. 이방성 필터링의 개념 도시. 왼쪽 이미지는 등방성 필터링, 오른쪽은 이방성 필터링을 나타낸다. 픽셀의 높이는 날카로움의 정도를 나타낸다. 이방성 필터링에서는 초점이 맞은 날카로운 영역의 컬러가 배경으로 흘러나오지 않게 한다.



(a) (b) (c)

그림 5. 등방성과 이방성 가우시안 필터링을 이용하여 생성된 밉맵 보간의 예. (c)는 (a)와 (b)의 적색 사각형 영역을 확대한 것이다. (b)에서 강도의 누출이 대폭 감소한 것을 볼 수 있다. 본 비교에서는 평균 32 픽셀의 CoC 지름이 사용되었다.

수식에서 $\beta = m-1$ 이다. 이와 같이, 불연속으로 정의된 밉맵을 연속적인 블러링/CoC의 정도에 대응시킬 수 있고, 연속적인 블러링 정도를 가지는 결과 이미지를 얻어낼 수 있다. 수식 (8)을 이용하여 확대된 이미지 역시, 대부분의 이미지 확대에서 보이는 이중선형 결점(bilinear artifact)을 보이는데, 한 레벨 아래의 이미지를 확대한 후 그 이미지들을 이용하여 가우시안 필터링을 적용하면, 확연하게 이 결점을 r마소시키고, 고품질의 블러링 이미지를 생성할 수 있다.

4. 이방성 가우시안 필터링을 사용한 강도 누출 제거

3장에서 제안한 방법은 각 픽셀의 CoC의 지름에 따라 필드심도 효과에 필요한 블러링을 매우 효율적으로 생성하지만, 후처리 방식의 공통적인 문제점인 강도 누출과 전경 블러링의 불연속성을 보인다. 이에, 본 장에서는 강도 누출을 막기 위한 이방성(anisotropic) 필터링을 소개하고, 다음 장에서는 전경의 불연속성을 제거하는 방법에 대해 설명한다.

강도의 누출은 각 픽셀의 CoC에 따라 등방성(isotropic) 가우시안 필터링을 적용하기 때문에 일어난다. 주변 픽셀이 초점이 맞는 영역에 속하면, CoC가 1 픽셀에 가까울 것이므로 그 영향이 중심 픽셀이 미치지 않아야 하지만, 등방성 필터링은 이를 고려하지 않으므로 초점이 맞은 픽셀이 배경 픽셀에 그 강도를 누출되게 된다. 이러한 결점은 필터링에 기반을 둔 대부분의 후처리 방법들에서 보이므로, 강도 누출을 방지하기 위해 몇몇 방법이 제안되었지만 그 부하 때문에 방법이 실시간에 실행되기 어렵거나, 또 다른 결점을 생성하였다. [6-7] 이에, 본 논문에서는 간단하고 효과적이면서, 그 수행성능을 감소시키지 않는 방법을 제안한다.

배경에서 강도 누출을 방지하는 방법의 핵심은 자신보다 초점이 맞은 영역 쪽으로 가까이 있는 주변 픽셀을 필터링하는 동안 섞지 않는 것이다. 그림 4는 이방성 필터링의 착안점을 도시한다. 이방성 필터링을 위해, 수식 (4)에서 밉맵을 생성하기 위해 적용하였던 가우시안 필터 계수 G를 다음과 같이 정의되는 이방성 계수 H로 바꾸게 된다.

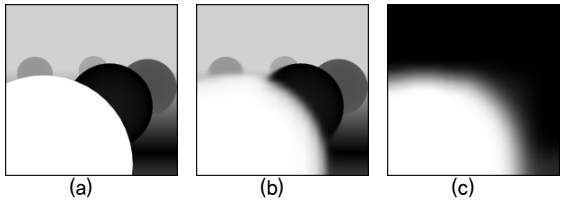
$$H(\mathbf{p},\mathbf{q},l) = \begin{cases} 0 & \text{if } \sigma_{\mathbf{p}}(l) > \sigma_{\mathbf{q}}(l)+\eta \text{ and } Z_{\mathbf{p}}(l) > Z_f \\ G(\mathbf{q}-\mathbf{p}) & \text{otherwise} \end{cases} \quad (9)$$

수식에서 $Z_{\mathbf{p}}(l)$과 Z_f는 밉맵 레벨 l에서 중심 픽셀과 초점의 깊이를 나타낸다. σ 는 블러링의 정도 즉, DoB 값이며, η 는 비슷한 DoB 값을 가지는 픽셀들이 필터링에 포함되도록 하는 편차(offset) 값이다. G는 수식 (4)에서 쓰인 가우시안 필터 계수를 가리킨다. 본 식을 이용하여 강도의 누출에 기여하는, 초점 거리보다 멀리 위치하며 중심픽셀보다 가까운 픽셀들은 블러링에 영향을 미치지 않게 되고, 따라서 초점 영역 바깥의 날카로운 경계가 잘 유지된다. 그림 5는 등방성과 이방성 필터를 적용한 이미지를 비교한 예를 나타낸다.

5. 전경 블러링 불연속성의 제거

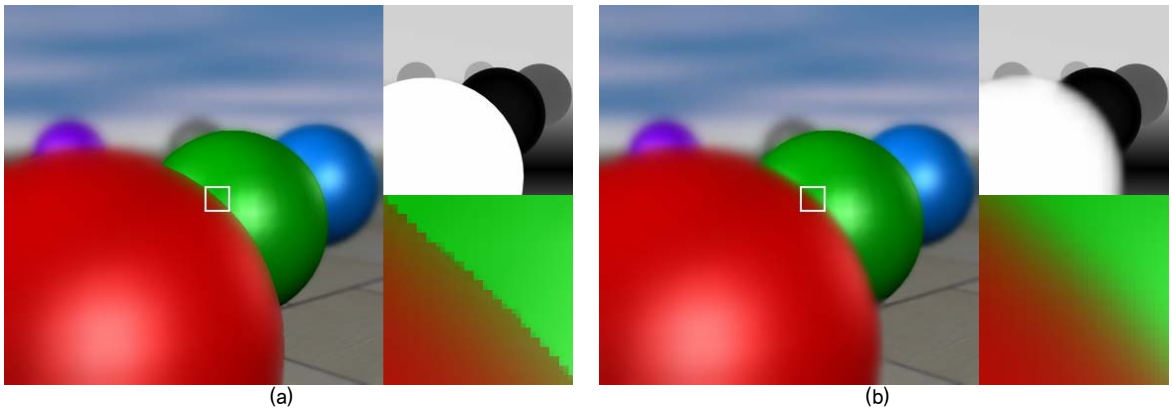
본 장에서는 4장에서 언급한 바와 같이 불연속적인 블러링 경계를 제거하는 방법을 기술한다. 불연속 블러링은 전경(foreground)에 위치한 물체의 경계에서 블러링의 정도가 급격히 변해, 눈에 띄게 부자연스러운 현상을 지칭한다. 고품질의 필드심도 효과를 위해서는 이 결점은 반드시 제거되어야 하지만, 대부분의 후처리 방법들은 이를 간과하였다.

이에 본 논문에서는 렌더링 속도를 저하시키지 않으며 불연속 블러링을 효과적으로 제거하는 방법을 제안한다. 불연



(a) (b) (c)

그림 6. (a) 원래 DoB 이미지, (b) 전경이 부드럽게 된 DoB 이미지, (우) 최고 레벨 M에서의 DoN 이미지



(a) (b)

그림 7. 불연속적인 DoB(a)와 전경을 부드럽게 한 DoB(b)를 이용하여 생성된 필드심도 이미지의 예. 오른쪽 위 이미지는 해당 DoB이미지이고, 아래는 백색 사각 영역을 확대한 이미지이다. (b)에서 불연속성이 효과적으로 제거되었음을 알 수 있다.

속 블러링이 일어나는 이유는 각 픽셀에서의 블러링이 단일(중심) 픽셀의 깊이에 의해 결정되기 때문이다. 실제 필드심도 효과에서의 블러링은 주변 픽셀들과 함께 결정되므로, 블러링 정도(DoB)를 주변 픽셀들을 고려하여 결정한다면, 그 불연속성은 효과적으로 제거될 수 있다.

3장의 수식 (7)에서 DoB σ (즉, σ (0))는 가장 고해상도의 깊이 이미지에서 결정되었다. 이 DoB 이미지를 블러링시킴으로서, 그 주변 픽셀들의 DoB값을 가져오는 효과를 낼 수 있다. 불연속성을 다루는 유일한 기존 방법은 DoB 이미지 전체에 블러링을 적용하지만, 이 방법은 초점 영역에서 블러링되지 말아야 할 부분들이 블러링되는 단점을 가지고 있다. [6] 이에, 본 논문에서는 각 픽셀별로 DoB의 블러링 정도를 다르게 주기 위해 픽셀별로 상이한 밉맵 레벨 n에서 DoB를 블러링함으로써, 그러한 단점을 제거한다.

각 픽셀에서 DoB를 블러링하기 위한 밉맵 레벨, n은 자신보다 가까운 주변 전경 픽셀들의 블러링 정도(DoN; Degree of Neighbor's Blurring)를 이용하여 결정한다. 주변 픽셀들의 블러링이 자신보다 많이 크다면, 그 픽셀들이 전경에 해당한다는 것이고, 주변 픽셀들의 블러링이 적다면 그 픽셀들이 자신과 비슷한 깊이에 있거나, 배경에 해당한다는 의미이다. 수식 (10)은 수식 (7)과 동일한 원리를 적용하여 DoB를 블러링하는 과정을 나타낸다.

$$n = \begin{cases} 0 & \text{if } k_\sigma \rho(M) < 0.5 \\ 1+\log_2(k_\sigma \rho(M)) & \text{otherwise} \end{cases} \quad (10)$$

수식에서 M은 현재 컬러 밉맵에서 가장 높은 레벨(가장 작은 크기의 밉맵 레벨)을 의미한다. $\rho(M)$은 DoN 밉맵에서 레벨 M에서 읽어온 값을 의미한다.

그림 6은 본 과정을 도시한다. 현재 가장 높은 밉맵 레벨 M에 해당하는 DoN 텍스쳐 값을 읽어와(그림 6(c)), 이를 DoB의 블러링을 위해 사용하면 전경에 해당하는 부분만 블러링된 효과를 볼 수 있다(그림 6(b)). 이렇게 읽어온 DoB n을 이용하여, 수식 (7)의 σ (실제로 σ (0))을 σ (n)으로 대체하면, 전경 경계가 부드럽게 처리된다. 그림 7은 전경의 불연속성을 제거한 이미지의 예를 도시한다.

6. 결과

6.1 렌더링 성능

저자는 본 논문에서 제안한 방법을 Intel 2.67 GHz Core2Duo 기반의 PC에서 nVidia GeForce 7900GTX를 이용하여 구현하였다. 렌더링을 위한 소프트웨어는 OpenSceneGraph와 OpenGL Shading Language를 이용하였다. 이렇게 구현된 필드심도 렌더링 방법의 성능을 측정하고, 근래에 제안된 GPU 기반 방법인 Scheuermann과 Zhou의 성능과 비교하였다(그림 8). 본 방법의 수행 성능(적색

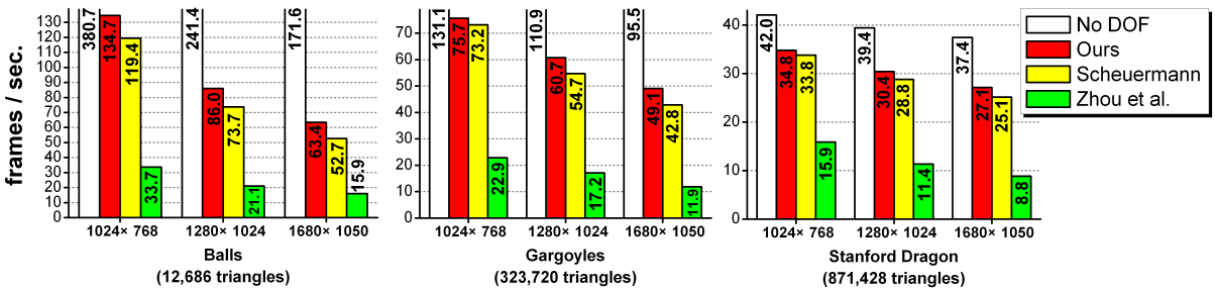
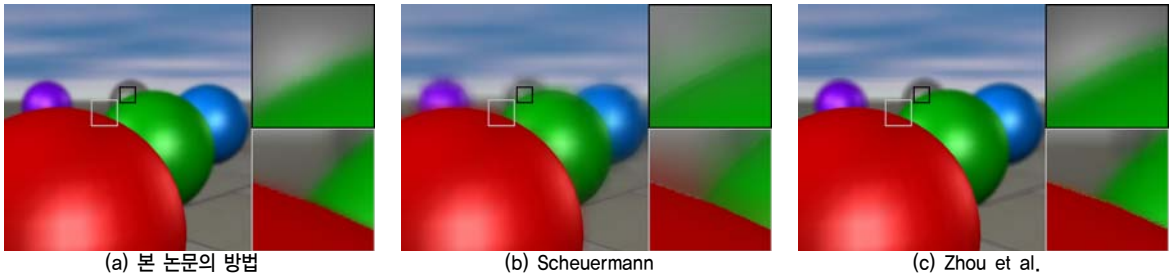
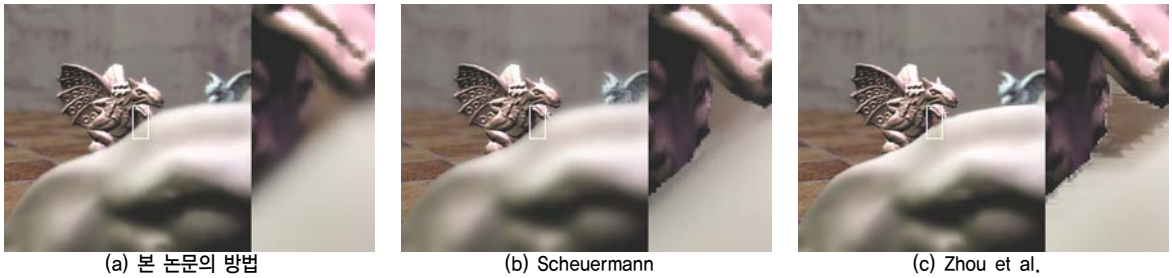


그림 8. Scheuermann과 Zhou의 방법과의 필드심도 렌더링 성능 테스트 비교 결과.



(a) 본 논문의 방법 (b) Scheuermann (c) Zhou et al.

그림 9. 근래 GPU 기반의 방법[5,7]과의 결과 비교. 초점은 전경의 적색 구에 맞춰져 있다. 흑/백 사각형은 오른쪽에 확대되어 있다.



(a) 본 논문의 방법 (b) Scheuermann (c) Zhou et al.

그림 10. 근래 GPU 기반의 방법과의 결과 비교. 본 논문의 방법만이 유일하게 부드럽게 처리된 전경의 경계를 보인다.

바)은 기존의 두 방법들에 비해 향상된 매우 뛰어난 실시간 성능을 보인다. 예를 들어 1024×768의 해상도에서 75.7 frame/s로, 실시간 응용프로그램에서 다른 특수효과와 함께 쓰이기에 충분한 성능을 보인다. 나아가 근래에 제안된 방법들 중, 가장 빠른 Scheuermann 방법보다도 더 나은 렌더링 성능을 보여줌을 알 수 있다.

본 방법의 또 다른 큰 장점은 수행 성능이 CoC의 크기에 관계없이 단지 해상도에만 연관되어 있다는 것이다. 대부분의 기존 방법들은(Scheuermann 제외) CoC에 의존적으로 수행되기 때문에, 큰 스케일의 블러링이 필요한 경우에 그 성능이 급격히 감소하여, 실시간 응용분야에서 그 사용이 제한되어 왔다. 이에 반해, 본 논문의 방법은 CoC의 스케일에 관계없이 매우 안정적인 성능을 확보할 수 있다.

6.2 렌더링 품질

그림 9와 10은 제안된 방법을 이용하여 전경과 배경에 초점을 두고 렌더링한 결과 이미지이다. 백/흑색으로 표시된 영역은 원 이미지의 오른쪽에 확대되어 있다. 전체적으로, 본 논문의 방법은 두 결과 이미지 모두에서 자연스러운 필드심도 효과를 생성하였다. 그림 9는 전경의 적색 구에 초점을 맞춘 이미지로서, Scheuermann의 이미지에서 심각한 강도 누출이 관찰되는데 반해, 본 논문과 Zhou의 방법에서는 그러한 결점이 거의 없음을 볼 수 있다. 그림 10은 배경의 Gargoyle에 초점을 맞춘 이미지로서, 전경에서 다른 두 방법(Scheuermann과 Zhou)이 심각한 전경 불연속성 및 다른 결점을 보이는데 반해, 본 방법은 연속적으로 부드럽게 이어진 경계를 자연스럽게 생성하였음을 볼 수 있다. 이렇듯이, 근래까지의 후처리 방식 중에 강도 누출과 전경 블러링의 불연속성을 모두 제거하며, 또 다른 결점이 없는 방법은 본 논문에서 제안된 방법이 유일하다.

7. 결론

본 논문에서 제안한 방법은 높은 품질의 필드심도 효과를 렌더링할 뿐만 아니라, 매우 빠른 실시간 수행성능을 보이는 방법으로, 근래의 필드심도 렌더링 방법들 중에서 유일하게 품질과 성능을 동시에 만족하는 방법이다.

참고문헌

[1] M. Potmesil and I. Chakravarty, "A lens and aperture camera model for synthetic image generation". Proc. SIGGRAPH, Vol. 15, No. 3, ACM, pp. 297-305, 1981.

[2] R. Cook, T. Porter, and L. Carpenter, "Distributed ray tracing" Computer Graphics, vol. 18, no. 3, pp. 137-145, 1984.

[3] P. Haeberli and K. Akeley, "The Accumulatin Buffer: Hardware support for high-quality rendering". Proc. SIGGRAPH, pp. 309-318, 1990.

[4] P. Rokita, "Fast Generation of depth of field effects in computer graphics", Computers and Graphics, vol. 17, no. 5, pp. 593-595, 1993.

[5] T. Scheuermann, "Advanced depth of field" in Game Developers Conference, 2004.

[6] M. Bertalmio, P. Fort, and D. Sanchez-Crespo, "Real-time, accurate depth of field using anisotropic diffusion and programmable graphics card" in Proc. 3DPVT, pp. 767-773, 2004.

[7] T. Zhou, J. Chen and M. Pullen, "Accurate depth of field simulation in real time", Computer Graphics Forum, vol. 26, no. 1, 2007.

[8] Y. C. Chen, "Lens effect on synthetic image generation based on light particle theory", The Visual Computer, vol. 3, no. 3, pp. 125-136, 1987.