ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
논문번호 19-04-36

j.inst.Korean.electr.electron.eng.Vol.23,No.4,1366~1372,December 2019
http://dx.doi.org/10.7471/ikeee.2019.23.4.1366

몬테칼로 렌더링 노이즈 제거를 위한 듀얼 신경망 구조 설계

Design of a Dual Network based Neural Architecture for a Cancellation of Monte Carlo Rendering Noise

이 광 엽*★

Kwang-Yeob Lee*★

Abstract

In this paper, we designed a revised neural network to remove the Monte Carlo Rendering noise contained in the ray tracing graphics. The Monte Carlo Rendering is the best way to enhance the graphic's realism, but because of the need to calculate more than thousands of light effects per pixel, rendering processing time has increased rapidly, causing a major problem with real-time processing. To improve this problem, the number of light used in pixels is reduced, where rendering noise occurs and various studies have been conducted to eliminate this noise. In this paper, a deep learning is used to remove rendering noise, especially by separating the rendering image into diffuse and specular light, so that the structure of the dual neural network is designed. As a result, the dual neural network improved by an average of 0.58 db for 64 test images based on PSNR, and 99.22% less light compared to reference image, enabling real-time race-tracing rendering

요 약

본 논문에서는 레이 트레이싱 그래픽에서 사용되는 몬테칼로 렌더링에 포함되는 잡음을 제거하기 위해 개선된 신경망구조를 설계하였다. 몬테칼로 렌더링은 그래픽의 실감을 높이는데 가장 좋은 방법이지만 픽셀마다 수천 개 이상의 빛 효과를 계산해야 하기 때문에 렌더링 처리시간이 급격히 증가하여 실시간 처리에 큰 문제를 갖고 있다. 이 문제를 개선하기 위해 픽셀에서 사용되는 빛의 수를 줄이게 되는데 이때 렌더링 잡음이 발생하게 되고 이 잡음을 제거하기 위해 다양한 연구가 진행되어 왔다. 본 논문에서는 렌더링 잡음을 제거하는데 딥러닝을 사용하며 특히, 렌더링 이미지를 확산광과 집중광으로 분리하여 이중 신경망 구조를 설계하였다. 설계결과 단일구조 신경망에 비하여 듀얼구조 신경망은 PSNR기준으로 64개 테스트 이미지에 대하여 평균 0.58db가 개선되었으며 reference image에 비하여 99.22% 빛의 수를 줄여 실시간 레이 트레이싱 렌더링을 구현하였다.

Key words : Ray Tracing, Denoising, MonteCarlo rendering, Autoencoder, Neural Network, Graphics

* Dept. of Computer Engineering, Seokyeong University
★ Corresponding author
E-mail : kylee@skuniv.ac.kr, Tel : +82-2-940-77245.
※ Acknowledgment
This work was supported by Seokyeong University in 2019.
Manuscript received Dec. 18, 2019; revised Dec. 21, 2019; accepted Dec. 26, 2019.
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.

Ⅰ. 서론

최근 Nvidia 뿐만 아니라 Intel에서도 컴퓨터 그래픽의 실감능력을 높이기 위해 광선 추적(ray tracing)기법을 적극 개발하고 제품화 하고 있다. 광선 추적은 그래픽을 구성하는 모든 픽셀(pixel)에서 빛의 경로를 계산하고 그 점의 광도로 색채를 정하여 화면을 합성하는 방식이다. 장점은 매우 고품질의 자연스러운 그래픽 화면을 합성할 수 있으나, 단점은 광원에서부터 빛을 하나하나 추적하여 이미지를 만들어 내기 때문에 연산 량이 굉장히 많다. 이를 해결하기 위해 이미지의 픽셀에서부터 역으로 빛을 추적해 광원까지 추적하는 방법을 path tracing이라한다. 현재는 path tracing이 ray tracing을 의미한다.

픽셀에서의 모든 빛을 추적하여 이미지를 만드는 것은 현실적으로 불가능하다. 따라서, 몇 개의 빛을 픽셀에서부터 출발하여 추적하게 되는데 이들의 평균을 내어 픽셀 값을 정하게 된다. 이 방법을 Monte Carlo Rendering[1]이라고 한다.

픽셀당 많은 빛을 추적 할 경우 좀 더 현실과 같은 이미지가 생성된다. 빛을 추적하는 단위는 sample per pixel(spp)라고 한다. spp가 감소하면 연산량은 줄어들지만, noise가 많은 이미지를 출력한다. 반대로 spp 증가하면 현실적인 이미지를 출력하지만, 연산량이 매우 많이 증가한다. spp에 따른 연산 량 차이는 아래 표 1과 같다[2].

Table 1. The difference of calculation amount according to spp.
표 1. spp 변화에 따른 연산 량 비교

Image Resolution	8spp	128spp	1024spp	8196spp
720p	7,373	117,965	943,718	7,553,434
1080p	16,589	265,421	2,123,366	16,986,931
4K	70,779	1,132,462	9,059,697	72,477,573

(unit = 1000)

표 1에서 보듯이 가장 높은 수준의 영상인 4K기준으로 잡음이 거의 없는 8196spp로 ray tracing할 때 1개의 sample당 1번의 연산이 실행한다고 가정하면 대략 724억번의 연산이 필요하다. 이러한 연산 량은 실시간으로 처리하기가 매우 어려운 연산량이다. 따라서 연산량을 줄이기 위해서는 spp수를 줄여야 하는데, 표 2와 같이 spp 수가 감소하면 상대적으로 이미지에 포함되는 잡음량이 증가하고, 이미지의 품질이 낮아진다. 본 논문에서는 적은 spp를 사용하면서도 많은 spp를 사용한 이미지와 동일한 수준의 이미지 품질을 갖도록 이미지에서 잡음을 제거하는 방법을 제안한다. 또한 잡음제거에 효율이 높은 인공지능 기반의 신경망(Neural Network)을 적용하고 적용하는 신경망의 최적 구조를 제안한다. 특히, ray tracing에서 픽셀에 작용하는 빛의 성분을 분석하여 성분에 따라 잡음을 제거하는 신경망을 별도로 설계함으로 써 잡음의 효율을 개선하는 구조를 제안한다.

Table 2. PSNR and SSIM according to the spp variance,
표 2. spp 변화에 따른 신호대비 잡음 비(Peak Signal Noise Ratio)와 구조적 유사도(Structural SIMilarity) [3][4]

	8spp	128spp	1024spp
PSNR	13.6834	20.1158	25.1086
SSIM	0.1839	0.3987	0.6321

Ⅱ. Ray tracing 픽셀 파라메터 분석 및 PreProcessing 설계

1. 광선효과

Ray tracing은 그림 1(a)와 같이 입사광과 가까이 있는 물체의 반사광이 ray casting되면서 발생하는 수 많은 빛의 ray들이 교차점에서 만들어 내는 수 많은 색을 반영하는 기술이다. Casting되는 reflection ray는 diffuse와 specular로 크게 나누어 볼 수 있다. Diffuse reflection은 입사광(incoming light)이 물체의 표면에서 반사될 때 모든 방향으로 동일하게 퍼져 나가지 않고 그림 (b)와 같이 표면의 거칠기에 따라 밝기와 방향이 달라진다. 반면, specular reflection은 그림 (c)와 같이 좁은 beam 형태로 빛이 같은 방향으로 동시성을 갖고 반사되어 물체의 표면이 더욱 밝게 나타난다. 이때, beam의 angle은 물체 표면의 smoothness에 의존한다.

2. 렌더링 출력이미지 광선 파라메터 분석

본 논문에서는 ray tracing에서 Monte Carlo 알고리즘으로 구현된 렌더링 출력 이미지에 포함된

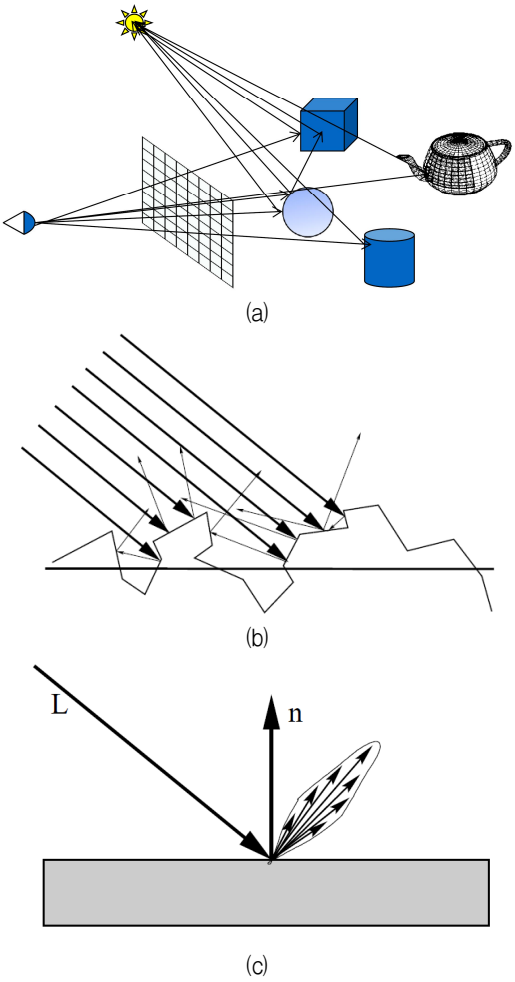

Fig. 1. (a) Ray casting (b) Diffuse reflection (c) specular reflection
그림 1. (a) Ray casting (b) 확산 반사 (c) 집중 반사

다양한 광선 파라메터를 분석하였다. Monte Carlo 렌더링 테스트 이미지에서 주요 성분이 위에서 설명된 확산광과 집중광임을 확인하였다.

Ray tracing의 모든 픽셀에서 광선의 성분량을 분석하기 위해서는 다음과 같이 식(1)을 적용한다. 식(1)은 법선 방향의 입사광에 대한 반사광 함수(f_r)로 입사zenith angle(θ_i)와 시점zenith angle(θ_r) 그리고 specular 방향각(φ), 표면경사도(σ)로 구성된다. 그러나 거친 표면에 대한 specular peak 성분을 잘 포함하면서 모델을 단순화한 식(2)을 사용한다[5].

$$f_r(\theta_i,\theta_i,\phi) \propto \frac{L_i\cos\theta_i}{L_i\cos\theta_i} \cdot \frac{1}{\cos\theta_r} \cdot \frac{e^{-\frac{\varphi^2}{2\sigma^2}}}{\sqrt{2\pi\sigma}} \propto \frac{e^{-\frac{\varphi^2}{2\sigma^2}}}{\sigma\cos\theta_r} \quad (1)$$

$$f_r \propto \frac{F(\theta_i,\theta_r,\phi,\hat{n}) \cdot G(\theta_i,\theta_r,\phi) \cdot e^{-c^2\alpha^2}}{\cos\theta_i \cdot \cos\theta_r} \quad (2)$$

Monte Carlo 렌더링을 통하여 만들어진 테스트 이미지 64장을 바탕으로 식(2)에 근거하여 이미지의 모든 픽셀에서 확산광과 집중광의 평균 분포확률을 구해보면 다음과 표 3과 같다. 다른 빛의 성분과 확산광, 집중관을 분리하기 위한 픽셀별 밝기는 0.001 이상의 차별이 발생하는 경우로 결과를 만들었다. 또한, 테스트 이미지별로 확산광과 집중광의 픽셀별 성분을 보면 확산광의 경우 모든 이미지에서 균일하게 분포되어 있으나 집중광의 경우 이미지 별로 성분이 다양하기 때문에 ray tracing 렌더링의 결과 이미지에서 확산광과 집중광을 분리하여 잡음을 제거하는 것이 더 높은 효율을 얻을 수 있음을 알아냈다.

Table 3 Diffuse light and Specular light Distribution
표 3 확산광과 집중광 분포

Test Image No.	*Diffuse light Average Distribution ratio*	*Specular light Average Distribution ration*
64	92.81%	62.38%

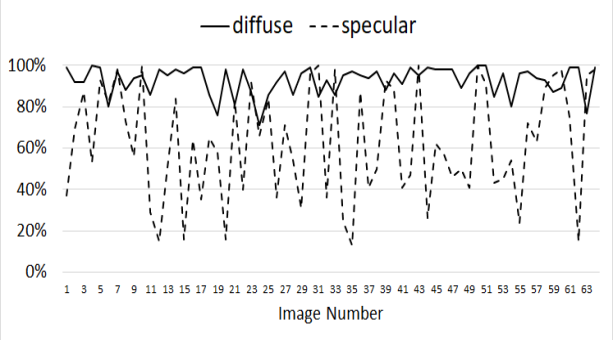

Fig. 2. Diffuse light and Specular light distribution chart in 64 test images.
그림 2. 64테스트 이미지에 포함된 확산광과 집중광의 분포도

III. Dual CNN 설계

본 논문에서는 앞장에서 실험한 결과를 반영하여 Monte Carlo 랜더링 출력 이미지를 확산광 이미지와 집중광 이미지로 분리하고 각각의 이미지를 사용하는 Dual CNN을 설계하였다.

1. Dual Layer

제안하는 신경망은 다음 그림 3과 같이 Diffuse Network과 Specular Network으로 분리하고 잡음

이 포함된 랜더링 이미지를 두 개의 신경망이 동시에 입력 이미지로 받아 dual로 처리하게 된다. Dual Network의 출력 이미지는 output layer에서 합을 통하여 잡음이 제거된 이미지로 최종 출력된다. Dual Network은 신경망 가운데 이미지 처리에 가장 효율이 높은 Convolution Neural Network구조로 설계하는데 diffuse와 specular network은 동일한 구조를 갖기 때문에 하드웨어 크기가 두 배로 증가하는 단점을 갖게 된다. 이를 해결하기 위해 본 논문에서는 은닉층에 해당하는 convolution layer의 크기를 대폭 줄일 수 있는 autoencoder[6]를 적용하고 출력단을 1 x 1 convolution 구조로 설계하였다.

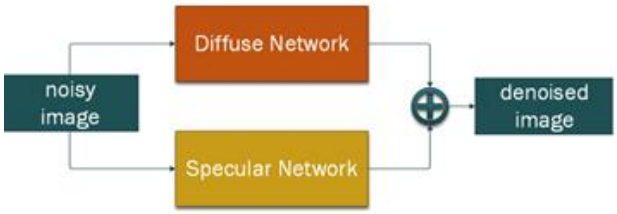

Fig. 3. Dual Nueral Network Architecture.
그림 3. 이중 신경망 구조

2. Autoencoder CNN 설계

구현된 Autoencoder는 Encoder를 사용하여 입력 이미지 크기를 점점 줄여가며, feature map의 크기는 늘려간다. 이때 가장 많은 feature map을 가지고 있는 Encoder와 Decoder사이의 Layer를 Bottleneck Layer라하며, 총 512개의 feature map을 가진다. 이후 Decoder를 통과하며 원본이미지 크기를 복구한다. 이때, noise는 Decoder를 통해 제거되어 noise가 없는 영상으로 출력된다. 추가적으로 Reconstruction Layer를 통해 미처 제거되지 못한 지 않은 noise와 filter의 경계에서 발생할 수 있는 checked pattern

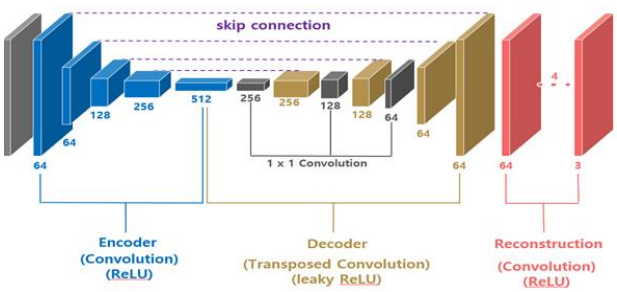

Fig. 4. Denoising Auto Encoder.
그림 4. 잡음제거 Auto Encoder

을 제거한다. 제안하는 Neural Network 구조는 그림 4와 같다.

3. 1×1 Convolution

본 논문에서는 Decoder의 연산량을 감소시키기 위해 1×1 convolution[7]를 설계에 적용하였다. 1×1 convolution을 활용하면 featurmap의 크기가 감소할 때, 연산량이 매우 많이 감소하는 기법이다.

Ⅳ. Autoencoder CNN기반 Denoising 구현

설계된 Autoencoder CNN구조에서 Monte Carlo Rendering Noise를 제거하는 과정은 Preprocessing, Encoding / Decoding, Reconstruction으로 이루어진다. Preprocessing을 통해 Neural Network의 입력으로 사용할 수 있게 noisy image를 변환하고 Encoding Layer를 통과하면서, feature(특징점)들이 추출된다. 추출된 feature들은 다시 Decoding Layer를 통과하며, 원본 이미지 크기로 돌아오게 된다.

마지막으로 Reconstruction Layer를 통과하여 Denoising Image가 생성된다. 학습에서 이미지는 patch단위로 무작위로 영상에서 뽑아낸 64×64 크기의 작은 이미지를 단위로 한다. 원본 이미지를 이용해 학습을 할 경우 너무 큰 용량의 메모리를 필요로 하므로, patch단위로 나누어 저장한 뒤에 batch단위로 patch들을 모아서 사용한다.

1. Preprocessing

Preprocessing은 renderer에서 나오는 출력 이미지를 Neural Network의 입력으로 사용할 수 있도록 하는 작업이다. Renderer에서 생성되는 이미지는 Color(R, G, B) 뿐만아니라, color값을 계산하기 위해 사용되었던 값들을 출력 시킬 수 있다. 이를 활용하여 Neural Network의 입력으로 사용한다.

기본적으로 생성되는 Color 3ch, Specular 3ch, Diffuse 3ch, Normal 3ch, Albedo 3ch, Depth 1ch를 사용한다. 추가적으로, Color, Specular, Diffuse, Normal, Albedo의 분산 2ch과 Depth의 분산 1ch 총 22ch를 이용한다. 이때, 편차가 큰 값들을 Neural Network의 입력으로 이용할 경우 모델이 발산할 가능성이 있다. 따라서, 분산의 제곱근을 취해 표준편차로 변경하여 사용한다.

추가적으로, noise의 후보들과 edge들을 검출하기

위해, 모든 pixel값들의 gradient를 계산한다. gradient는 x방향으로 한번, y방향으로 한번 진행한다. 따라서, Color 3ch, Specular 3ch, Normal 3ch, Albedo 3ch, Depth 1c 총 16ch 씩 2개 총 32개가 생성된다. 마지막으로 Noise가 조금 제거된 Median값들을 이용하여 Neural Network에 추가적인 힌트를 주게 된다. 이는 noise가 존재하는 color, specular, diffuse에만 적용한다. Preprocessing을 통과한 이미지는 총 63ch를 가지게 된다.

2. Encoding/Decoding

Encoding은 이미지의 사이즈는 줄이면서 feature map의 개수를 늘리는 작업이다. 실제로 이미지가 1024×1024의 크기일 경우 Encoding을 거치며 1024×1024, 512×512, 256×256, 128×128 크기까지 줄어든다. 중간에 Encoding과 Decoding사이의 bottle neck Layer에서는 총 512개의 feature map을 가진다. Activation function으로 Rectified Linear Unit (ReLU)[8]를 사용한다.

Encoding에서의 각 층의 정보는 저장되고 Decoding에서 Skip Connection[9]으로 사용된다. Decoding은 특징점을 다시 줄이며 원본 이미지를 복구하는 작업이다. 각 층은 Encoding Layer와 Skip Connection을 통해 연결 되어있다. 이 Skip Connection은 Encoding을 통과하며 찾아낸 feature map들을 원본 이미지를 복원할 때 더해준다. 이를 통해, 학습할 때 좀 더 빠르게 원본 이미지에 가까운 이미지를 생성하는 가중치 값을 찾아낸다. 또한, 실제 이미지의 detail한 부분을 더 잘 살려준다. Decoder의 Activation function은 Leaky Rectified Linear Unit(leaky ReLU)[10]를 사용한다. 추가적으로, 이미지가 원본을 복원하면서, feature map들을 줄이게 되므로, 1×1 convolution을 사용할 수 있다. 1×1 convolution은 필터를 통과하기 전에 우선적으로 줄임으로서 연산 량을 대략 1/10까지 줄일 수 있다.

3. Reconstruction

Reconstruction은 Encoding/Decoding을 완료한 이미지의 후처리를 위한 층이다. Encoding/ Decoding 층에서 미처 제거되지 못한 noise와 Encoding/Decoding을 통과하며 필터의 경계에서 발생하는 체크무늬 패턴을 제거하는 용도로 사용된다. 이 층을 통과한 이미지는 feature map들로 이루어진 이미지에서 R, G, B로만 구성된 3채널의 이미지로 변환된다.

Ⅴ. Experiment Result

제안하는 CNN구조의 잡음제거 성능을 측정하기 위해 최소화된 spp를 이용하여 랜더링 시간과 inference time을 측정하였다. 또한, 제안하는 dual 구조의 우수성을 입증하기 위해 64개의 테스트 이미지를 reference로 하여 단일 CNN구조와 잡음 제거 성능을 비교 하였다.

1. Rendering time test

우선 64개 테스트 영상에 대하여 8196spp와 64spp 렌더링을 실행하고 잡음 대비 동일한 품질을 보일 때 실행시간을 비교하였다. 64spp에는 본 논문에서 제안하는 autoencoder neural network을 적용하였다. 실험 환경은 AMD's 2990wx (32core) and tungsten renderers에서 실행하였으며 그 결과는 표 4와 같다. rendering time을 비교할 때 본 논문의 renderer가 240배의 실행시간 단축을 보인다.

Table 4. Rendering time comparison.
표 4. 렌더링 시간 비교

	64 spp	8196 spp
Rendering Time	20sec	1hour 20min 48sec

2. Inference time test

본 논문에서 제안하는 1×1 convolution 기반 autoencoder의 효과를 측정하기 위해 실험 영상에서 inference time을 측정하였다. 앞에서 설명한 측정 환경에서 표 5와 같은 결과를 보였다. 1×1 convolution을 사용하는 경우에 30% 처리시간이 단축되었음을 알 수 있다.

Table 5. Inference time average in 64 scenes.
표 5. 64scene을 사용한 평균 inference time

	without 1x1	with 1x1
Inference Time	0.1930sec	0.1534sec

3. Single CNN vs. Dual CNN

본 논문에서 제안하는 Dual CNN 구조의 성능을 동일한 64개 테스트 이미지에 대하여 Single CNN 구조와 비교 하였다. 다음 그림 5에서 두 구조의 비

교 결과를 보면 Dual CNN이 PSNR기준으로 평균 0.58db 개선이 되었다.

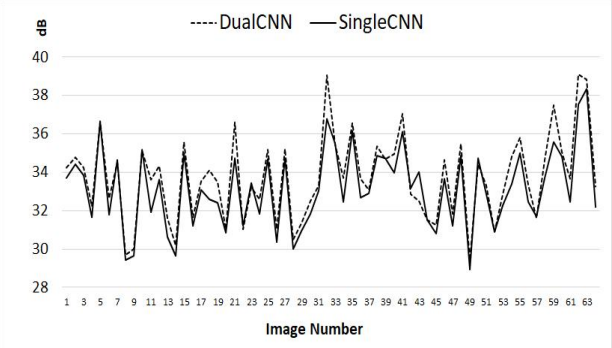

Fig. 5. Dual CNN vs. Single CNN in PSNR(dB)
그림 5. 이중구조 CNN과 단일구조 CNN의 PSNR성능비교 (단위 : dB)

Ⅵ. Conclusion

본 논문에서는 Ray tracing의 결과 이미지를 만들기 위해 Monte Carlo randering 알고리즘을 사용하여 이미지의 픽셀 데이터를 생성하게 된다 이때, 64spp의 ray수를 갖고 랜더링을 실행하여도 8196spp의 ray로 랜더링한 결과와 거의 동일한 이미지 품질을 보장하기 위해 랜더링 이미지에서 잡음을 제거 하게 된다.

Ray의 수를 줄여 처리속도를 개선하면 실시간 Ray tracing graphics를 구현이 가능하여 모바일 기기에서 고품질 컨텐츠 구현이 가능하게 되었다. 특히, 본 논문에서 제안하는 1×1 convolution구조의 autoencoder는 다양한 neural network에 적용가능하여 mobile neural network 연구에도 이바지 할 수 있다.

References

[1] VEACH, Eric, GUIBAS, Leonidas J. “Optimally combining sampling techniques for Monte Carlo rendering,” *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, pp.419-428, 1995.
DOI: 10.1145/218380.218498

[2] H. H. Lee, K. Y. Lee, C. W. Jo. “Implementation of Autoencoder based Neural Network for Realtime Ray Tracing,” *Proceedings of 2019 3rd Domestic and International Integration conference.* SoCoRI, p.419-428, 1995.

[3] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P.. “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing,* Vol.13, No.4, pp.600-612, 2004.
DOI: 10.1109/TIP.2003.819861

[4] Hore, A., Ziou, D., “Image quality metrics: PSNR vs. SSIM,” *20th International Conference on Pattern Recognition,* IEEE, pp.2366-2369, 2010.
DOI: 10.1109/ICPR.2010.579

[5] G. Meister, R. Wiemaker, R. Monno, H. Spitzer, A. Strahler, “Investigation on the Torrance-Sparrow Specular BRDF Model,” *International Geoscience and Remote Sensing Symposium,* IEEE, 1998. DOI: 10.1109/IGARSS.1998.703752

[6] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of machine learning research,* pp.3371-3408, 2010.

[7] Veach, E., & Guibas, L. J., “Optimally combining sampling techniques for Monte Carlo rendering,” *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques,* 1995.

[8] Nair, V., & Hinton, G. E., “Rectified linear units improve restricted boltzmann machines,” *Proceedings of the 27th international conference on machine learning,* ICML, pp.807-814, 2010.

[9] Mao, X. J., Shen, C., & Yang, Y. B., “Image restoration using convolutional auto-encoders with symmetric skip connections,” arXiv preprint arXiv:1606.08921, 2016.

[10] Maas, A. L., Hannun, A. Y., & Ng, A. Y., “Rectifier nonlinearities improve neural network acoustic models,” Proceeding of ICML, p.3, 2013.

BIOGRAPHY

Lee Kwang-yeob (Life Member)

1985 : BS degree in Electronics Engineering, Sogang University
1987 : MS degree in Electronics Engineering, Yonsei University.
1994 : PhD degree in Electronics Engineering, Yonsei University.

1989~1995.2 : Senior Researcher, Hyundai Electronics Inc.
1995.3~present : Professor, Dept. of Computer Engineering, Seokyeong University