Indirect Illumination Using Convolutional Neural Networks

Emil Georgiev

Introduction

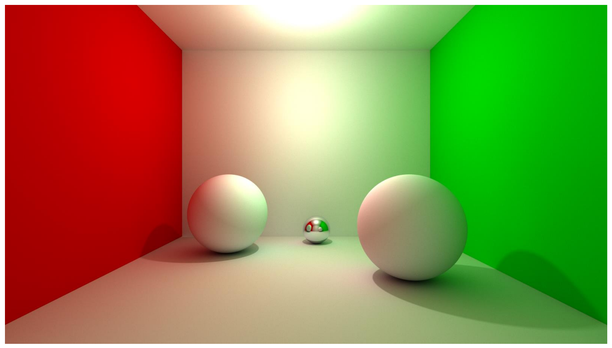

Figure 1: *Cornell box with 3 spheres[1]*

- Indirect Illumination is very computationally expensive
- Demand for real-time photorealistic lighting
- Could convolutional neural networks provide fast and realistic results?
- Prior work:

Figure 2: *Deep Shading Indirect Illumination[2]*

Problem Statement

- Research and develop an indirect illumination solution using convolutional neural networks in Keras.
- Demonstrate graphical quality and computation speed

Design

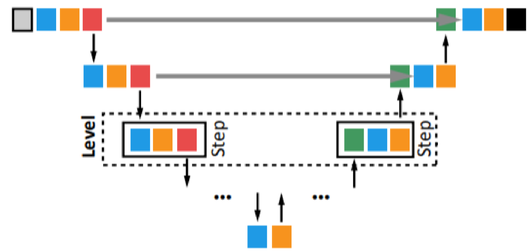

Figure 3: *U-shaped convolutional neural network[2]*

- Autoencoder-like network
- Concatenates the output of each downsampling stage with the input of the corresponding upsampling stage
- Better generalization compared to normal convolutional neural networks

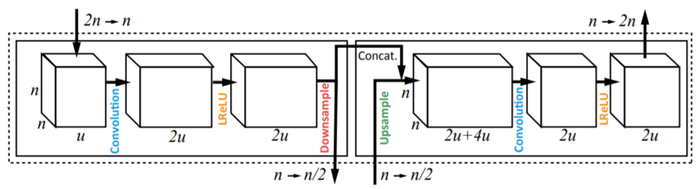

Figure 4: *Convolution-deconvolution modules[2]*

- Dataset contains 6006 training samples
- Each sample contains a per-pixel position, normal and world-space normal image
- Flipped and rotated sample augmentations are already included in the dataset
- 90/10 dataset split for training and testing, respectively

Evaluation and Conclusion

- Prediction speed: 10ms on average, real-time speeds
- Training done for 10 epochs, batch size of 16 on a GTX 1060 Max-Q
- Results approximate indirect illumination well at a high level, though they're a bit blurry

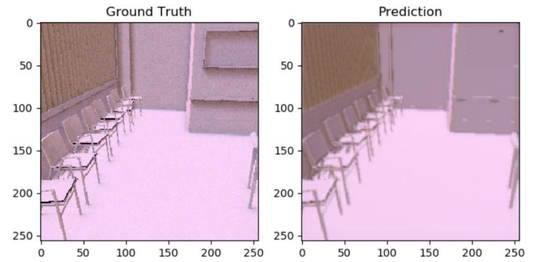

Figure 5: *Ground truth and predicted indirect illumination*

epoch	acc	loss	mse	val_acc	val_loss	val_mse
0	0.638109	0.044259	0.0442587	0.6710758	0.0349066	0.0349066
1	0.66767	0.020324	0.0203237	0.6512966	0.0172496	0.0172496
2	0.673508	0.015186	0.0151859	0.6878428	0.0111325	0.0111325
3	0.678208	0.012914	0.012914	0.6858813	0.0125183	0.0125183
4	0.680011	0.011734	0.0117337	0.6853667	0.0087281	0.0087281
5	0.684416	0.011305	0.0113049	0.6845878	0.0098130	0.0098130
6	0.684996	0.011179	0.0111788	0.6941132	0.0093401	0.0093401
7	0.689735	0.009982	0.0099823	0.6890625	0.0084884	0.0084884
8	0.691956	0.009422	0.0094224	0.6913835	0.0060989	0.0060989
9	0.692405	0.008430	0.0084301	0.6958721	0.0054813	0.0054813

Figure 6: *Training, validation loss and metrics*

References

- [1] JakBB (2015). Global Illumination in Minecraft. [image] Available at: https://imgur.com/gallery/5iE3m [Accessed 12 May 2019]
- [2] Nalbach, O., Arabadzhiyska, E., Mehta, D., Seidel, H.-P., and Ritschel, T. (2017). Deep Shading: Convolutional Neural Networks for Screen Space Shading. Available at: http://deep-shading-datasets.mpi-inf.mpg.de/deep-shading-preview.pdf [Accessed 10 May 2019]