Machine Learning is the Solution Also for Foveated Path Tracing Reconstruction

Atro Lotvonen[a], Matias Koskela[b] and Pekka Jääskeläinen[c]
Tampere University, Finland

Keywords: Foveated Rendering, Path Tracing, Neural Network.

Abstract: Real-time photorealistic rendering requires a lot of computational power. Foveated rendering reduces the work by focusing the effort to where the user is looking, but the very sparse sampling in the periphery requires fast reconstruction algorithms with good quality. The problem is even more complicated in the field of foveated path tracing where the sparse samples are also noisy. In this position paper we argue that machine learning and data-driven methods play an important role in the future of real-time foveated rendering. In order to show initial proofs to support this opinion, we propose a preliminary machine learning based method which is able to improve the reconstruction quality of foveated path traced image by using spatio-temporal input data. Moreover, the method is able to run in the same reduced foveated resolution as the path tracing setup. The reconstruction using the preliminary network is about 2.9ms per 658 × 960 frame on a GeForce RTX 2080 Ti GPU.

1 INTRODUCTION

High quality real-time graphics rendering remains a major challenge in producing an immersive *Mixed Reality* (MR) experience with contemporary computing platforms. A significant part of the computational cost of photorealistic rendering can be reduced by means of foveated rendering where the quality is reduced in the peripheral parts of the human vision (Guenter et al., 2012). However, most of the foveated rendering work currently focuses merely on reducing the resolution, although peripheral vision has also other understudied aspects which could be utilized for foveated rendering (Albert et al., 2019). Therefore, we believe there is room for research in the area of better foveated reconstruction filters.

Conventionally, the final frame from foveated sparse samples has been reconstructed with linear interpolation. However, machine learning-based methods are an interesting future direction also for reconstruction because they can also take other than resolution reduction factors such as temporal stability of the content into account. However, current machine learning methods suffer from a slow inference runtime and need a loss function that accurately models human vision. These problems are likely going to be solved in the future by the increasingly fast inference hardware available and development of faster deep learning layers with similar performance (Shi et al., 2016). In addition, a recent work shows that the loss function can be a combination of multiple simpler loss functions and it does not have to include a single accurate model of the human visual system (Kaplanyan et al., 2019).

In this position paper we argue that, in the future, machine learning algorithms are going to play a critical role in reconstructing foveated rendering results. We also argue that the reconstruction network could be done only in the reduced foveated resolution. We also show preliminary evidence that machine learning-based reconstruction can be fast enough for wider adaption in the field of foveated rendering.

2 PATH TRACING IN REAL TIME

There are many ways to render content for MR devices. Path tracing (Kajiya, 1986) is an interesting candidate because it naturally supports many effects such as reflections, refractions, and caustics (Pharr et al., 2016) which are harder to achieve with other methods.

[a] https://orcid.org/0000-0002-2985-4339
[b] https://orcid.org/0000-0002-5839-7640
[c] https://orcid.org/0000-0001-5707-8544

Lotvonen, A., Koskela, M. and Jääskeläinen, P.
Machine Learning is the Solution Also for Foveated Path Tracing Reconstruction.
DOI: 10.5220/0009156303610367
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 1: GRAPP*, pages 361-367
ISBN: 978-989-758-402-2; ISSN: 2184-4321

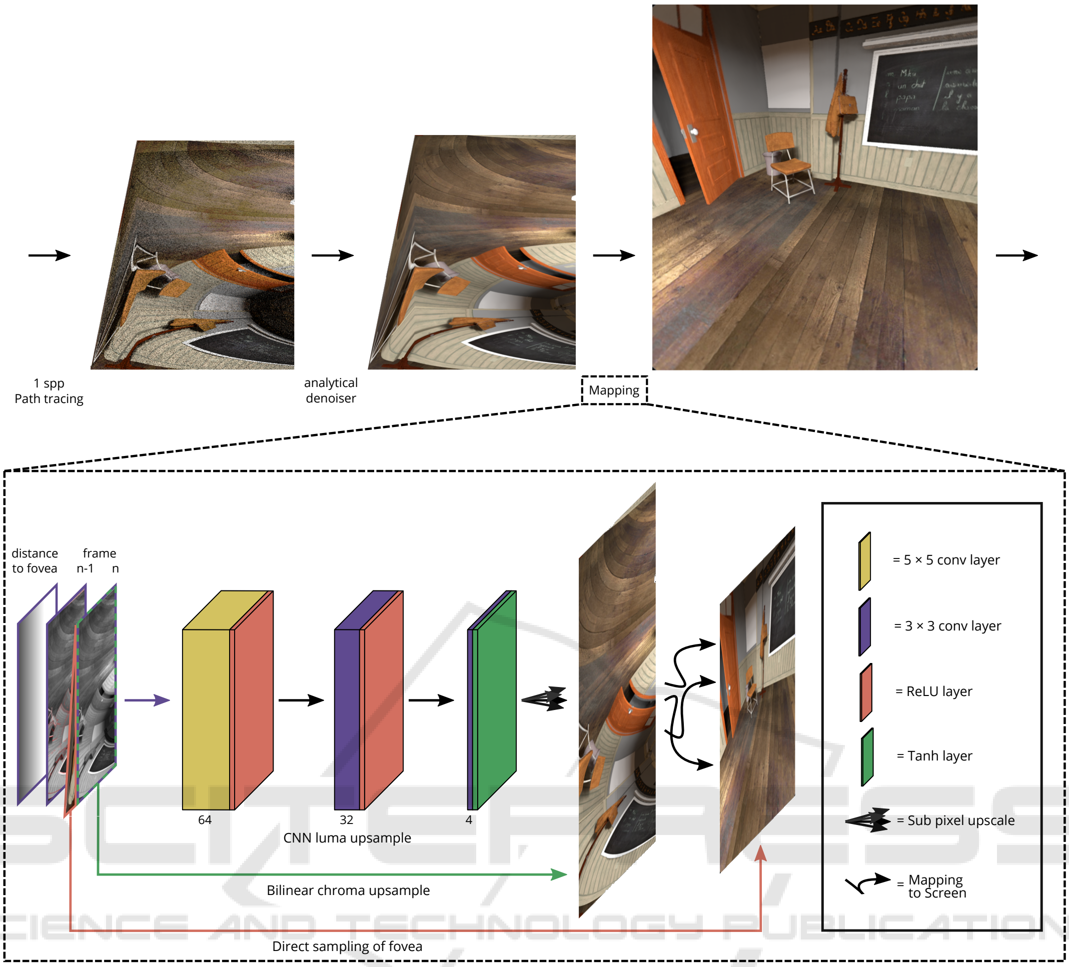

Figure 1: An example of a foveated rendering pipeline similar to Koskela et al. (2019b). In this position paper we claim that the mapping stage should be done with machine learning. A similar mapping could be used with other rendering pipelines that render foveated samples in continuous space like Friston et al. (2019). However, direct sampling of the fovea would be in the center of the input image.

Due to its extreme computational requirements, however, real-time path tracing was for a long time considered unfeasible, but recently dedicated hardware acceleration features for ray traversal, the primitive operation of path tracing have started to appear in state-of-the-art desktop processors (Kilgariff et al., 2018). However, even with the hardware acceleration, the Monte Carlo integration of path tracing can only generate a very noisy estimate of the final image (Barré-Brisebois, 2018). This noise can be removed in real time with analytical filters such as regression-based methods (Koskela et al., 2019a) or fast edge preserving blurs (Schied et al., 2017).

Analytical denoising of path tracing has also been used in foveated rendering (Koskela et al., 2019b). In this method the path tracing and denoising are done in a polar space which is modified to distribute the samples according to visual acuity model of the human eye. Another option instead of polar space would be to use magnified Cartesian screen space rendering (Friston et al., 2019). Typically, reconstruction of the final screen space frame is done with simple filters like linear interpolation. However, the human visual system can detect a presence of a pattern before resolving it and therefore it is a good idea to add contrast to the peripheral parts of the foveated rendering (Patney et al., 2016). Also, an analytical filter can simulate other rendering effects such as depth of field (Weier et al., 2018). However, since data driven methods have been dominating other fields of filling missing image data (Hays and Efros, 2007) and as the commercial processors increasingly add hardware support for inference, in this paper we argue that machine learning can be a solution also to foveated rendering reconstruction.

3 MACHINE LEARNING RECONSTRUCTION

Multiple machine learning based methods have been proposed previously for removing path tracing noise, but their run-time has so far been too slow for real-time applications (Chaitanya et al., 2017; Vogels et al., 2018; Bako et al., 2017). Deep learning has also produced the state-of-the-art results for *Single Image Super Resolution* (SISR) (Ledig et al., 2017). The usual means for creating high resolution images from low resolution ones with deep learning has been to first upsample the low resolution image with an interpolation filter such as bicubic interpolation and then use *Convolutional Neural Networks* (CNN) to improve the result (Dong et al., 2015). However, Shi et al. (2016) suggest that analytical upscaling does not provide additional information for solving the problem. Instead the solution can be found by doing the nonlinear convolutions only in the low resolution space, and finally upscaling the image with an efficient sub-pixel convolutional layer. This way the network learns the upscaling filters within the network, which reduces the computational cost.

Furthermore, with real-time super resolution for video data Caballero et al. (2017) found that including temporal data with motion compensation improves the quality of the result which can be used to reduce the computational cost of the network with similar quality. For temporal denoising and reconstruction of path traced images, Bako et al. (2017) argues that using temporal data shifted with motion vectors is better than using recurrent layers like proposed by Chaitanya et al. (2017).

In addition, different fusing strategies of temporal data in video streams has been tested (Karpathy et al., 2014; Caballero et al., 2017). In early fusing, the temporal data is handled by giving the same input frames to all of the input layers. With slow fusing, different input CNN layers handle different input frames and are collapsed later in the network. It was found that early fusing works better for shallower networks while in deeper networks the slow fusing is more beneficial.

In this position paper, based on these previous works, we show initial evidence to our argument by proposing a preliminary machine learning-based method for reconstructing foveated path tracing. An idea for the architecture of the CNN is to use the efficient sub-pixel convolutional layer. In that case the reconstruction layers are computed in the lower foveated resolution and therefore the network learns upscaling filters instead of just improving the quality of an image that is nearest neighbor upsampled. As a result, we can decrease the spatial and temporal aliasing in the lower resolution rendered periphery, and also take advantage of the lower computational requirements. Furthermore, the proposed method uses temporal data shifted with motion vectors which is fused early in the network because the network architecture is shallow.

4 PRELIMINARY METHOD

In this section, we propose a preliminary machine learning-based method for reconstructing foveated path tracing. We focus on the periphery part of the foveated path traced result since we assume the foveated region already has sufficient quality in the described system. Also, like in previous work we are focusing only on the luminance components of the input images because the human eye is more sensitive to changes in the luminance channel in the YCbCr color space (Shi et al., 2016).

4.1 Path Tracing Setup

In our example setup, the 1 spp path tracing is done the same way as described in Koskela et al. (2019b); we follow the visual acuity function by sampling in the Visual-Polar space. We also use the same foveated resolution reduction of 61% so that the full resolution of a single eye in the used head mounted display (1280×1440) can be rendered in a foveated resolution of 853×960. The used path tracing sample distribution mechanism could also be similar to (Friston et al., 2019), but using the Visual-Polar space allowed us to easily consider only the periphery part of the rendered image: In the case of the Visual-Polar transformation, the fovea is always mapped to the same location in the image. Thus, to focus on the periphery part of the rendering, it is enough to take a slice of the modified polar image on the horizontal ρ axis.

For denoising we can use state-of-the-art fast analytical path tracing denoisers such as Schied et al. (2017) or Koskela et al. (2019a) directly in the reduced foveated Visual-Polar space and then feed the denoised image to our reconstruction CNN.

4.2 Foveated Input and Output

For the input image we focus only on the periphery part of the foveated path traced image. In order to help the convolutional layers to differentiate the spatial resolution shift in the distribution, one of the inputs to the network is the distance to the fovea. In the case of the Visual-Polar space, the distance changes only on the horizontal ρ axis.

Only the luminance channel of the foveated path traced images is considered. This decreases the computation time of the network compared to full RGB values for the current and previous frames. In the case of the Visual-Polar space, the edges of the image are mapped to the middle of the final Cartesian screen space image. Therefore, basic padding, such as zero padding, in the Visual-Polar space leads to poorer quality in the middle of screen space image. This can be handled with a custom wrap-around padding.

Temporal data for the current frame is calculated using the previous frame and motion vectors. The data not present in the previous frame is filled with a value outside of a nominal luminance value. In this case we used a floating point value of -1.0. The idea is that the network learns not to use this data.

We use path tracing resolution of 853 × 960. For the machine learning reconstruction, we take a slice from the periphery of 658 × 960 with a custom padding of 4 on each border, which is fed to the network as an input. Fovea part of the Visual-Polar frame is directly sampled with anisotropic sampling to screen space.

4.3 Convolutional Neural Network Architecture

For our use case we have a similar architecture as is done in the case of (Shi et al., 2016). The foveated CNN is shown fully in Figure 1 with an example input and an output. We use a *hyperbolic tangent* (tanh) activation layer only in the last layer of the network and a *Rectified Linear Unit* (ReLU) in the first two, which we found to work better with our loss function. For each low resolution foveated denoised input we have a fully converged 4096 spp foveated 2× resolution image. Examples of the foveated input and the reference output can be seen in Figure 2. We also employed early fusion for the input data, in that all the three inputs are collapsed at the beginning of the network, which has been demonstrated to improve quality in shallow networks (Caballero et al., 2017; Karpathy et al., 2014).

Given Visual-Polar space images in resolution $W \times H$, the used pixel-wise *Mean Absolute Error* (MAE) loss function can be described as

$$\sum_{y=0}^{H}\sum_{x=0}^{W}|I_{x,y}^{HR} - f_{x,y}(I^{LR})|, \tag{1}$$

where I^{HR} is the high resolution version of the image and I^{LR} is the low resolution image. L1 is the most robust loss function and closest to perceptual difference (Bako et al., 2017) and (Chaitanya et al., 2017) also found that it is effective against outliers.

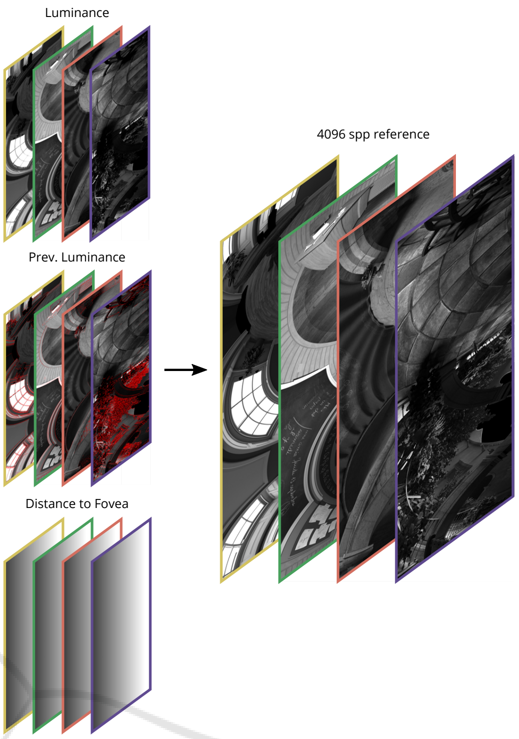

Figure 2: Teaching setup used in our experimental machine learning-based approach. Frames with the same border color are one set of inputs and their corresponding target. The missing data in previous frame is visualized in red pixel values.

5 EXPERIMENT

For training the network for the experiment, 3200 training samples were collected from various scenes, including the experimented ones. The use of separate training frames from the experiment scenes is not a problem in this application since it is difficult for a small network to learn scene specific parameters and similar training could also be used in the final use case. The network was trained using Keras with its TensorFlow backend. The network was trained for 40 epochs with the Adam Optimizer (Kingma and Ba, 2014) and 0.001 learning rate, after which no improvement for the cost function was observed. Furthermore, in this experiment we used the *Spatio-Temporal Variance-Guided* (SVGF) (Schied et al., 2017) filter as a denoiser in the pipeline and the training data for the network was collected with the same denoiser.

For the experiment, four different scenes were

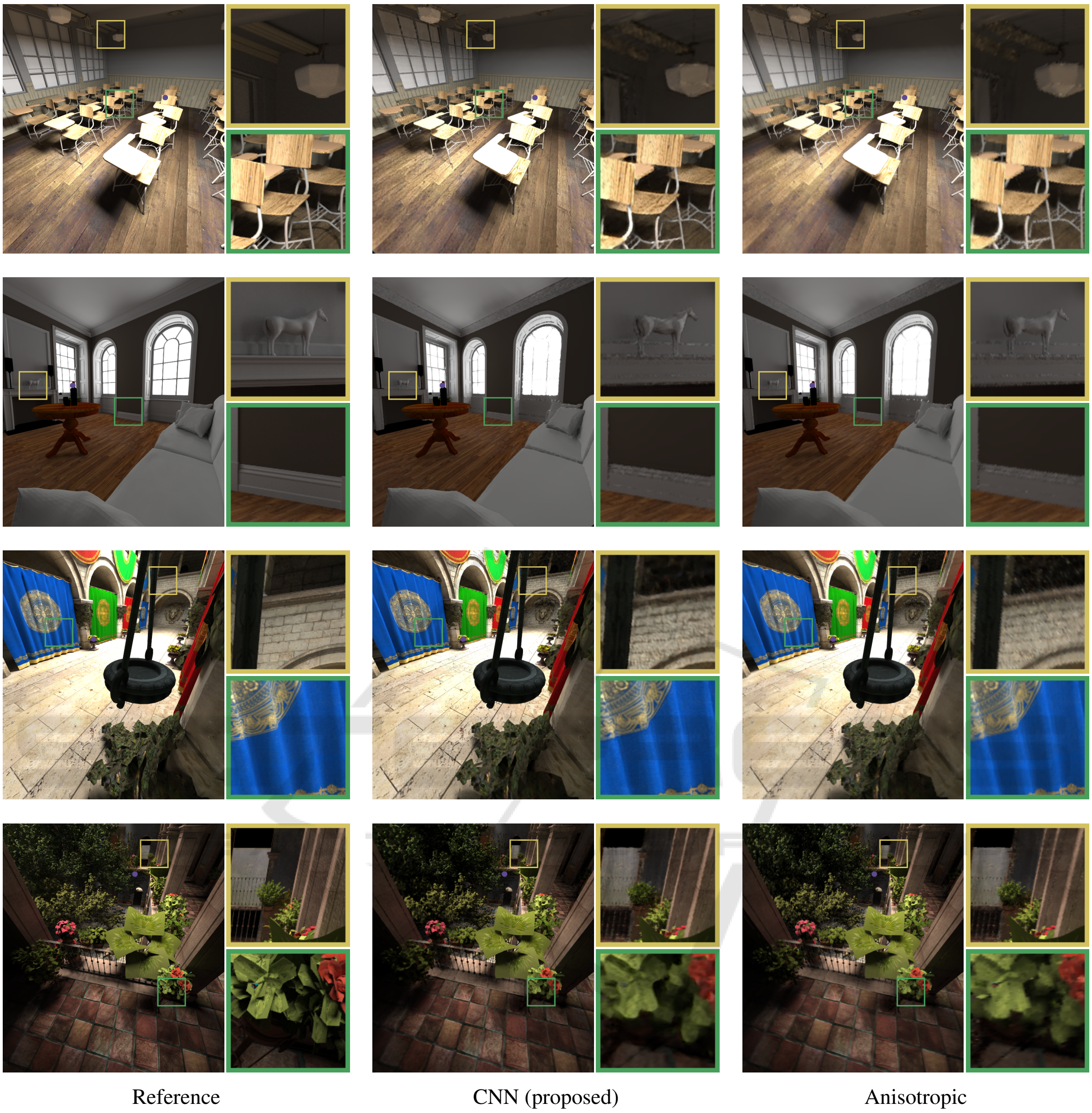

Figure 3: Comparison of different methods. The gaze point is marked with a purple dot. The reference is 4096 spp Cartesian space rendering. This shows how even a simple network learns how to reduce some of the errors produced by a real-time denoiser.

tested. From each scene, a short few second long camera path was recorded. For each camera path two foveated renderings were compared with a fully converged 4096 spp path traced reference. The first foveated rendering was an anisotropic sampling from Visual-Polar space to Cartesian space. The second compared rendering was the upscaled output of the proposed foveated CNN. Examples of the compared renderings can be seen in Figure 3. The results seen in Table 1 show that the proposed CNN gave improvement for *peak signal-to-noise ratio* (PSNR), *structural similarity* (SSIM) and *video multi-method assessment fusion* (VMAF). The improvement in quality could potentially be translated to more reduction for resolution in the foveated rendering space and therefore faster inference.

Running a 658 × 960 frame with the proposed CNN with TensorRT-acceleration (the arithmetics were quantized to 16-bit half floating point precision to fit its TensorCore units), takes on average *2.9ms* per

Table 1: Results for recorded camera path in four path traced scenes.

Metric	Scene	Anisotropic	Proposed
PSNR	Classroom	26.1	**26.6**
	Fireplace	31.8	**32.1**
	Sponza	22.1	**22.4**
	SanMiguel	22.0	**22.1**
SSIM	Classroom	0.826	**0.835**
	Fireplace	0.899	**0.902**
	Sponza	0.695	**0.704**
	SanMiguel	0.613	**0.625**
VMAF	Classroom	26.7	**37.7**
	Fireplace	37.6	**47.3**
	Sponza	11.4	**22.0**
	SanMiguel	10.9	**16.5**

frame on a single GeForce RTX 2080 Ti GPU. The timing is likely fast enough, because it is measured on a single contemporary GPU. This means that either the quality or the speed, or both, can be improved significantly on the next generations as inference hardware acceleration support evolves. Furthermore, in this experiment the network used 32-bit floating-point precision. Using reduced precision would make network faster likely without significant reduction in the quality Venkatesh et al. (2017).

6 CONCLUSIONS

In this position paper we argued that machine learning algorithms are going to play a critical role in reconstructing foveated rendering. Specifically, an interesting option is to perform machine learning-based reconstruction in the reduced foveated resolution to reduce the computational complexity. In order to provide initial proofs, we built a preliminary spatio-temporal super-resolving CNN to reconstruct the periphery part of a foveated rendering pipeline which functions in the reduced foveated resolution. The method was tested with four different scenes in which it improved the PSNR, SSIM and VMAF scores compared to anisotropic filtering. In our opinion, after a more comprehensive architecture and parameter search, a similar idea could be used to greatly reduce the computational complexity and improve the result of the reconstruction in foveated photorealistic rendering, making it usable in realistic mixed reality setups of the future.

We believe that the improved quality of the machine learning based reconstruction can be translated to more reduced foveated resolution and, therefore, less overall rendering workload, but this is yet to prove with more extensive user studies.

ACKNOWLEDGEMENTS

This work was supported by ECSEL JU project FitOptiVis (project number 783162), the Tampere University of Technology Graduate School, the Nokia Foundation and the Emil Aaltonen Foundation.

REFERENCES

Albert, R., Godinez, A., and Luebke, D. (2019). Reading speed decreases for fast readers under gaze-contingent rendering. In *Proceedings of the Symposium on Applied Perception*.

Bako, S., Vogels, T., McWilliams, B., Meyer, M., Novák, J., Harvill, A., Sen, P., Derose, T., and Rousselle, F. (2017). Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Transactions on Graphics (TOG)*, 36(4).

Barré-Brisebois, C. (2018). Game ray tracing: State-of-the-art and open problems. High Performance Graphics Keynote.

Caballero, J., Ledig, C., Aitken, A., Acosta, A., Totz, J., Wang, Z., and Shi, W. (2017). Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Chaitanya, C. R. A., Kaplanyan, A. S., Schied, C., Salvi, M., Lefohn, A., Nowrouzezahrai, D., and Aila, T. (2017). Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG)*, 36(4).

Dong, C., Loy, C. C., He, K., and Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307.

Friston, S., Ritschel, T., and Steed, A. (2019). Perceptual rasterization for head-mounted display image synthesis. *ACM Transactions on Graphics (TOG)*, 38(4).

Guenter, B., Finch, M., Drucker, S., Tan, D., and Snyder, J. (2012). Foveated 3d graphics. *ACM Transactions on Graphics (TOG)*, 31(6).

Hays, J. and Efros, A. A. (2007). Scene completion using millions of photographs. *ACM Transactions on Graphics (TOG)*, 26(3).

Kajiya, J. (1986). The rendering equation. *SIGGRAPH Computer Graphics*, 20(4).

Kaplanyan, A., Sochenov, A., Leimkuehler, T., Okunev, M., Goodall, T., and Gizem, R. (2019). Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. *ACM Transactions on Graphics (TOG)*, 38(4).

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*.

Kilgariff, E., Moreton, H., Stam, N., and Bell, B. (2018). NVIDIA Turing Architecture In-Depth. https://devblogs.nvidia.com/nvidia-turing-architecture-in-depth/ accessed: 2020-01-06.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koskela, M., Immonen, K., Mäkitalo, M., Foi, A., Viitanen, T., Jääskeläinen, P., Kultala, H., and Takala, J. (2019a). Blockwise multi-order feature regression for real-time path-tracing reconstruction. *ACM Transactions on Graphics (TOG)*, 38(5).

Koskela, M., Lotvonen, A., Mäkitalo, M., Kivi, P., Viitanen, T., and Jääskeläinen, P. (2019b). Foveated real-time path tracing in visual-polar space. In *Proceedings of the Eurographics Symposium on Rendering*.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Patney, A., Salvi, M., Kim, J., Kaplanyan, A., Wyman, C., Benty, N., Luebke, D., and Lefohn, A. (2016). Towards foveated rendering for gaze-tracked virtual reality. *Transactions on Graphics*, 35(6).

Pharr, M., Jakob, W., and Humphreys, G. (2016). *Physically based rendering: From theory to implementation*. Morgan Kaufmann.

Schied, C., Kaplanyan, A., Wyman, C., Patney, A., Chaitanya, C. R. A., Burgess, J., Liu, S., Dachsbacher, C., Lefohn, A., and Salvi, M. (2017). Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Venkatesh, G., Nurvitadhi, E., and Marr, D. (2017). Accelerating deep convolutional networks using low-precision and sparsity. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Vogels, T., Rousselle, F., McWilliams, B., Röthlin, G., Harvill, A., Adler, D., Meyer, M., and Novák, J. (2018). Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)*, 37(4).

Weier, M., Roth, T., Hinkenjann, A., and Slusallek, P. (2018). Foveated depth-of-field filtering in head-mounted displays. *Transactions on Applied Perception*, 15(4).