DeepFovea++: Reconstruction and Super-Resolution for Natural Foveated Rendered Videos

Christoph Reich
TU Darmstadt
christoph.reich@stud.tu-darmstadt.de

Marius Memmel
TU Darmstadt
marius.memmel@stud.tu-darmstadt.de

Jonas Henry Grebe
TU Darmstadt
jonas.grebe@stud.tu-darmstadt.de

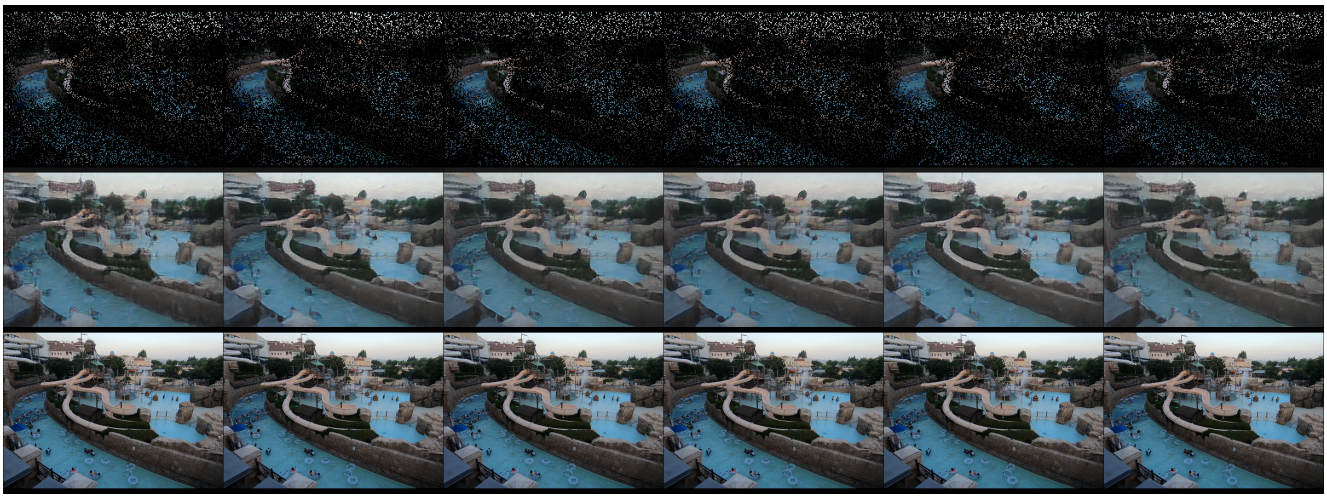

Figure 1. Results of our proposed DeepFovea++ technique. The fovea sampled input sequences of low resolution (192 X 256) image frames can be seen on the top. The reconstructed super-resolution (768 X 1024) prediction sequence is shown in the middle, and the corresponding label at the bottom.

Abstract

Image super-resolution is a well-known problem in the computer vision community. Recent papers extended the problem of super-resolution to videos and showed amazing results. On the other hand deep learning based fovea sampled image reconstruction has drawn some popularity since the DeepFovea publication of Facebook AI. DeepFovea showed outstanding results, however, the proposed reconstruction network was only able to reconstruct relatively low-resolution images of 128×128 *pixels. We revisit the proposed DeepFovea architecture to perform fovea sampled video reconstruction and super-resolution (*$192 \times 256 \rightarrow 768 \times 1024$*) at once. Our proposed architecture, DeepFovea++, first reconstructs a given video sequence by a recurrent U-Net architecture, and afterward, the desired super-resolution is learned by deformable convolutions. We tested our DeepFovea++ architecture on the challenging REDS dataset. The code is available at https://github.com/ChristophReich1996/DeepFoveaPP_for_Video_Reconstruction_and_Super_Resolution.*

1. Introduction

2. Previous Work

3. DeepFovea++ Architecture

The DeepFovea++ reconstruction model is mainly based on two parts. First, a recurrent residual U-Net [7, 4], and second, two super-resolution blocks based on deformable convolutions. We train both, the U-Net, and the super-resolution blocks, in an end-to-end setting. The whole re-

construction architecture consists of about 2.3 Mio parameters. We used a relatively small reconstruction model to be able to fit the whole DeepForvea++ framework, consisting of the reconstruction model, a 3d discriminator, and a 3D FFT-discriminator, into GPU memory.

3.1. Reconstruction model

3.2. Discriminator

3.3. FFT-Discriminator

3.4. Lossfunction

3.4.1 Supervised loss

3.4.2 Adversarial loss

3.4.3 Flow loss

4. Experiments

4.1. REDS Dataset

4.2. Implementation Details

We implemented the whole DeepFovea++ framework in PyTorch 1.4.0 [6]. Our implementation is based on multiple existing implementations. First, we build our framework on top of the deformable convolution v2 [10] implementation included in the mmdetection toolbox [2]. Second, we used the correlation package [3] of Nvidia for implementing the optical flow loss. Furthermore, to estimate the optical flow for the corresponding optical flow loss, we utilized a pre-trained PWC-Net [8] by Nvidia research. Finally, our main supervised loss function is based on the adaptive robust loss function implementation by Jonathan T. Barron [1].
For optimizing all networks we utilized the Adam optimizer [5]. For the reconstruction model, we set the learning rate to 3×10^{-4}. The first and second-order running average factors were set 0.1 and 0.95, respectively. In both the 3d discriminator and the 3d FFT-discriminator we set the learning rate to 10^{-4}. The first and second-order running average factors were set to the same value as for the reconstruction model.

TODO: Fovea mask!

TODO: Preprcessing

We were able to fit one sequence consisting of six rgb frames on one Tesla V100 (16GB). We trained our whole framework for 15 epochs on two GPUs. This results in a batch size of two. This training process took us about two days.

4.3. Results

For analyzing our framework results we calculated multiple metrics. First, we compute the common L1 metric and the Mean-Squared-Error (MSE, L2). In case for a reconstructed image $\mathrm{I}_{\mathrm{pred}} \in \mathbb{R}^{c,h,w}$ and the corresponding label $\mathrm{I}_{\mathrm{label}} \in \mathbb{R}^{c,h,w}$, the L1 and L2 loss is defined as

$$\mathrm{L1} = \frac{1}{n \times c \times w} \left\| \mathrm{I}_{\mathrm{pred}} - \mathrm{I}_{\mathrm{label}} \right\|_1 \tag{1}$$

$$\mathrm{L2} = \frac{1}{n \times c \times w} \left\| \mathrm{I}_{\mathrm{pred}} - \mathrm{I}_{\mathrm{label}} \right\|_2. \tag{2}$$

Additionally we compute the Peak-Signal-ToNoise (PSNR) and the, and Structural-Similarity-Image-Metric (SSIM) [9] to evaluate quality of prediction, which are defined as

$$\mathrm{PSNR} = 10 \log_{10}, \left(\frac{\max\{\mathrm{I}_{\mathrm{pred}}\}^2}{\mathrm{L2}\left(\mathrm{I}_{\mathrm{pred}}, \mathrm{I}_{\mathrm{label}}\right)} \right) \tag{3}$$

$$\mathrm{SSIM} = \frac{4\mathbb{E}\left[\mathrm{I}_{\mathrm{pred}}\right] \mathbb{E}\left[\mathrm{I}_{\mathrm{label}}\right] \mathrm{Cov}\left[\mathrm{I}_{\mathrm{pred}}, \mathrm{I}_{\mathrm{label}}\right]}{\left(\mathbb{E}\left[\mathrm{I}_{\mathrm{pred}}\right]^2 + \mathbb{E}\left[\mathrm{I}_{\mathrm{label}}\right]^2\right) \left(\mathrm{Var}\left[\mathrm{I}_{\mathrm{pred}}\right] + \mathrm{Var}\left[\mathrm{I}_{\mathrm{label}}\right]\right)}. \tag{4}$$

We tested our DeepFovea++ framework with two settings. In the first setting, we reset the recurrent state of the reconstruction model after each video sequence. In the second setting, we preserve the recurrent state over the whole training and validation process. Our tests lead to the following results.

TODO: Insert results

5. Conclusion

References

[1] J. T. Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019. 2

[2] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. mmdetection. https://github.com/open-mmlab/mmdetection, 2018. 2

[3] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 2

[4] A. S. Kaplanyan, A. Sochenov, T. Leimkühler, M. Okunev, T. Goodall, and G. Rufo. Deepfovea: neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. *ACM Transactions on Graphics (TOG)*, 38(6):1–13, 2019. 1

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 2

[7] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[8] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 2

[9] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[10] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 2

6. Appendix

Figure 2. Results of the first DeepFovea++ setting. The fovea sampled input sequences (192 X 256) on the top. The reconstructed super-resolution (768 X 1024) prediction sequence in the middle, and the corresponding label at the bottom.

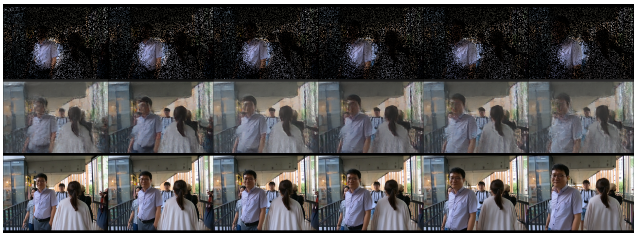

Figure 3. Results of the second DeepFovea++ setting. The fovea sampled input sequences (192 X 256) on the top. The reconstructed super-resolution (768 X 1024) prediction sequence in the middle, and the corresponding label at the bottom.

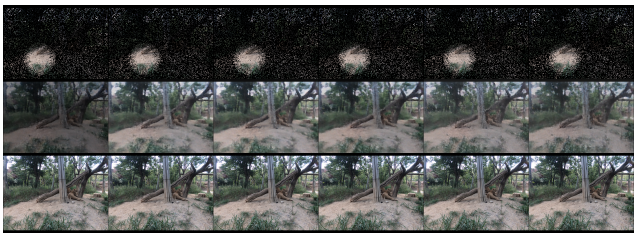

Figure 4. Results of the first DeepFovea++ setting. The fovea sampled input sequences (192 X 256) on the top. The reconstructed super-resolution (768 X 1024) prediction sequence in the middle, and the corresponding label at the bottom.

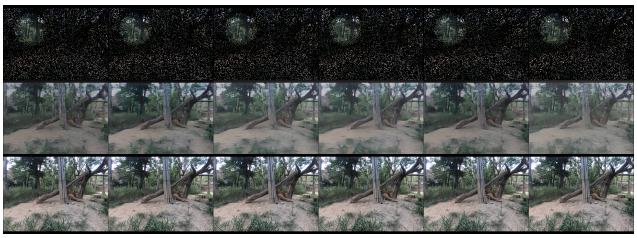

Figure 5. Results of the second DeepFovea++ setting. The fovea sampled input sequences (192 X 256) on the top. The reconstructed super-resolution (768 X 1024) prediction sequence in the middle, and the corresponding label at the bottom.

Figure 6. Results of the first DeepFovea++ setting. The fovea sampled input sequences (192 X 256) on the top. The reconstructed super-resolution (768 X 1024) prediction sequence in the middle, and the corresponding label at the bottom.

Figure 7. Results of the second DeepFovea++ setting. The fovea sampled input sequences (192 X 256) on the top. The reconstructed super-resolution (768 X 1024) prediction sequence in the middle, and the corresponding label at the bottom.