DeepFovea: Neural Reconstruction for Foveated Rendering and Video Compression using Learned Statistics of Natural Videos

ANTON S. KAPLANYAN, ANTON SOCHENOV, THOMAS LEIMKÜHLER*, MIKHAIL OKUNEV, TODD GOODALL, and GIZEM RUFO, Facebook Reality Labs

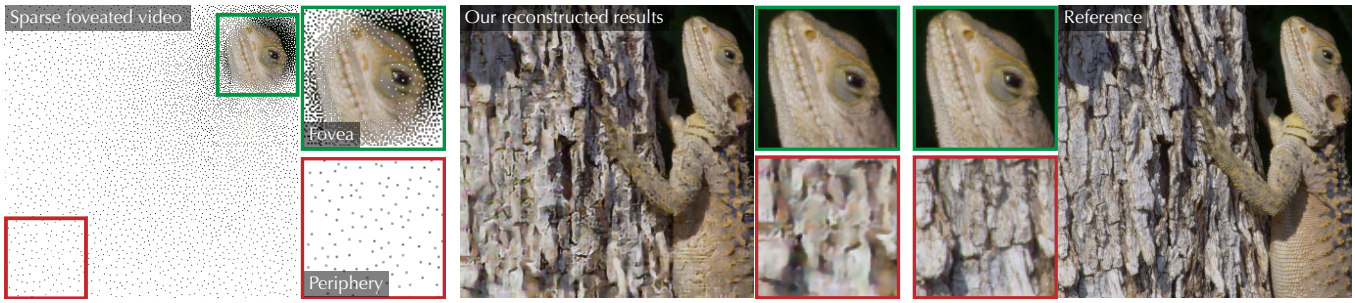


Fig. 1. Foveated reconstruction with DeepFovea. Left to right: (1) sparse foveated video frame (gaze in the upper right) with 10% of pixels; (2) a frame reconstructed from it with our reconstruction method; and (3) full resolution reference. Our method in-hallucinates missing details based on the spatial and temporal context provided by the stream of sparse pixels. It achieves 14x compression on RGB video with no significant degradation in perceived quality. Zoom-ins show the 0° foveal and 30° periphery regions with different pixel densities. Note it is impossible to assess peripheral quality with your foveal vision.

In order to provide an immersive visual experience, modern displays require head mounting, high image resolution, low latency, as well as high refresh rate. This poses a challenging computational problem. On the other hand, the human visual system can consume only a tiny fraction of this video stream due to the drastic acuity loss in the peripheral vision. Foveated rendering and compression can save computations by reducing the image quality in the peripheral vision. However, this can cause noticeable artifacts in the periphery, or, if done conservatively, would provide only modest savings. In this work, we explore a novel foveated reconstruction method that employs the recent advances in generative adversarial neural networks. We reconstruct a plausible peripheral video from a small fraction of pixels provided every frame. The reconstruction is done by finding the closest matching video to this sparse input stream of pixels on the learned manifold of natural videos. Our method is more efficient than the state-of-the-art foveated rendering, while providing the visual experience with no noticeable quality degradation. We conducted a user study to validate our reconstruction method and compare it against existing foveated rendering and video compression techniques. Our method is fast enough to drive gaze-contingent head-mounted displays in real time on modern hardware. We plan to publish the trained network to establish a new quality bar for foveated rendering and compression as well as encourage follow-up research.

CCS Concepts: • **Computing methodologies** → **Neural networks**; **Perception**; **Virtual reality**; *Image compression.*

Additional Key Words and Phrases: generative networks, perceptual rendering, foveated rendering, deep learning, virtual reality, gaze-contingent rendering, video compression, video generation

ACM Reference Format:
Anton S. Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo. 2019. DeepFovea: Neural Reconstruction for Foveated Rendering and Video Compression using Learned Statistics of Natural Videos. *ACM Trans. Graph.* 38, 4, Article 212 (July 2019), 13 pages. https://doi.org/10.1145/3355089.3356557

1 INTRODUCTION

Despite tremendous advances in consumer hardware for real-time rendering and video compression, the demand for high-fidelity visuals continues to grow. Recent advances in head-mounted displays allow us to achieve a new level of immersion by delivering the imagery straight to the eyes. However, such displays also require a significantly higher resolution and refresh rate to provide high quality immersion and good visual experience across the entire field of view. Rendering this high-quality content is challenging even on current high-end desktop systems.

On the other hand, the human eye has a very heterogeneous resolution density. It is able to resolve objects as small as 1 arcminute in the *fovea*, the center 5.2° region of the retina, and experiences a rapid acuity falloff outside the fovea toward the periphery [Curcio et al. 1990]. Fovea covers roughly 0.8% of pixels on a regular display under common viewing conditions [Guenter et al. 2012] and around 4% of pixels on consumer virtual reality (VR) headsets [Patney et al. 2016], such as HTC Vive and Oculus Rift. With the recent developments in gaze-contingent VR displays, such as the recently announced HTC Vive Pro Eye, it is also possible to estimate the user gaze in real time and perform gaze-contingent rendering and compression. This provides an important opportunity to optimize the amount of computation required to drive such displays, enabling

*Joint affiliation: Facebook Reality Labs, MPI Informatik.

Authors' address: Anton S. Kaplanyan, kaplanyan@fb.com; Anton Sochenov, anton.sochenov@oculus.com; Thomas Leimkühler, tleimkueh@mpi-inf.mpg.de; Mikhail Okunev, mokunev@fb.com; Todd Goodall, todd.goodall@oculus.com; Gizem Rufo, Facebook Reality Labs, gizem.rufo@oculus.com.

0730-0301/2019/7-ART212
https://doi.org/10.1145/3355089.3356557

higher quality visuals, larger resolution displays, and facilitating the miniaturization into mobile and wearable headsets, ultimately improving immersion and visual experience.

The foveation effect of human vision has been studied in various fields, including foveated video compression [Lee et al. 2001], and more recently foveated rendering [Guenter et al. 2012], achieving 50–70% in savings [Weier et al. 2017]. While peripheral compression has significant potential, one has to be careful about the artifacts that can be detected in the periphery, such as the loss of contrast with details (tunnel vision effect) and flicker [Patney et al. 2016], to which peripheral vision is especially sensitive [Rovamo et al. 1984]. Rendering can also employ a sparse foveated pattern [Stengel et al. 2016; Weier et al. 2016], which becomes a practical option with the recent advances in real-time ray tracing hardware, such as NVIDIA RTX, and fine resolution control for rasterization.

This motivated us to use a stochastically foveated video as an input, which is general enough to cover a wide variety of use cases. For example, this input is suitable both for foveated rendering and for foveated video compression. We use a sparse video stream with only a small fraction of pixels stochastically distributed per frame. For regular videos, we use a simple compressor that stochastically drops pixels in the periphery according to human visual acuity. This approach also makes the compressor compatible with many foveated rendering methods. Moreover, a video-producing module can be treated as a black box, compressing video streams produced from a variety of existing applications, e.g., for untethered or cloud-based VR gaming.

Our main contribution is the peripheral *reconstruction* method. DeepFovea starts with a sparse stream of color pixel values as an input. Given this sparse stream, we formulate the peripheral reconstruction problem as a projection-to-manifold problem, where the goal is to find the closest natural video that corresponds to the sparse foveated input on the manifold of all natural videos. This approach is similar to the internal model of the human visual system that infers content in the periphery using the available sparse and aliased peripheral information [Geisler 2008]. We employ this relation by using the recent advances in the adversarial training of generative video networks to train a reconstruction network to infer peripheral details based on the learned manifold of natural videos. This manifold also allows to infer the spatio-temporal semantic context based on an input video stream of sparse pixels. This allows us to achieve a significant reduction in the amount of required content without degrading the perceived quality in the peripheral vision.

At a glance, the contributions of our work are:

- A novel neural reconstruction that can inpaint details in the fovea and in-hallucinate temporally stable peripheral video content.
- A universal method that supports a video content produced by a black-box method.
- Over 14x reduction for foveated rendering without noticeable quality degradation.
- Real-time and low-latency reconstruction performance to drive gaze-contingent displays.
- Gaze-contingent user studies to analyze the quality and the detectability.
- Publishing the method as a baseline in foveated compression for the follow-up work.

In the remainder of the paper, Section 2 provides background on human perception and the potential of foveated compression, discusses existing foveated rendering and video compression methods and suitable quality metrics, and provides an overview of similar methods and recent generative methods in machine learning. Section 3 discusses our initial setting, design goals, and input assumptions. Section 4 describes the main reconstruction algorithm, network design, and training methodology we use. To evaluate the quality, we conduct a user study described in Section 5. The results, implementation details, and discussion are provided in Section 6.

2 PREVIOUS WORK

2.1 Background on Visual Perception

The ultimate receiver of the visual signal is the human eye. Its physiological structure determines how that visual signal is encoded for processing at subsequent stages of the visual system. The number of photoreceptors in the eye rapidly decreases from the fovea to the periphery [Curcio et al. 1990]. This fundamentally couples spatial sampling rate to *eccentricity*, the angular distance from the fovea. As sampling rate decreases with increasing eccentricity, our ability to perceive fine and mid-level details also decreases. Despite this loss in spatial resolution, temporal sensitivity remains roughly constant across all eccentricities [Rovamo et al. 1984].

When a video is presented, the light is captured by the 4.6 million cone photoreceptors [Curcio et al. 1990]. These photoreceptors feed subsequent retinal layers that encode this data for the midget ganglion cells which provide the pathway out of the eye. Thus, visual acuity, the ability to perceive spatial detail at or below a spatial frequency, is limited by the density of these midget cells. The Contrast Sensitivity Function (CSF) [Kelly 1984; Robson 1966] models this loss in perceptual contrast sensitivity of a stimulus as a function of its spatiotemporal frequency. Geisler and Perry [1998] provide a formulation of spatial frequency sensitivity in terms of eccentricity. Figure 2 relates cone and midget cell densities to visual acuity as a function of an angular distance from the fovea. The reduction in cell density from fovea to periphery ($0^\circ - 40^\circ$) is on the order of 30x [Dacey and Petersen 1992], providing a guide for reducing spatial details according to retinal physiology.

When designing a model based on this reduction, the spatiotemporal sensitivity of the eye must be carefully considered. Undersampling spatial details every frame without applying an appropriate pre-filter leads to aliasing-induced flicker as objects traverse points in the visual field. Neglecting spatiotemporal frequencies introduces another source of flicker as well as "tunnel vision" phenomena. Designing a model that respects these sensitivities and avoids flicker across the entire visual field is challenging.

2.2 Foveated and Perceptual Rendering

Delivering high quality content to each location in a head-mounted display (HMD) is computationally expensive. To save computation, peripheral compression becomes increasingly important for both rendered and captured video content. However, foveated rendering

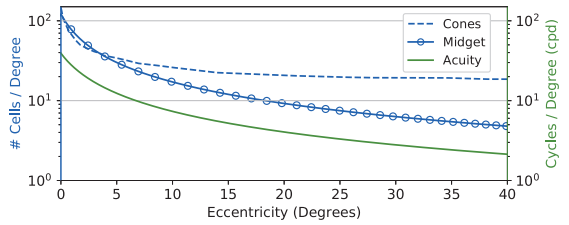


Fig. 2. Dropoff in acuity (green, cycles per degree) vs degrees eccentricity [Geisler and Perry 1998]; cone cell density [Curcio et al. 1990] and midget cell density [Bradley et al. 2014] distributions (blue).

can produce the aforementioned visual artifacts. Simply downsampling with eccentricity introduces aliasing and jitter. These phenomena encumber the design of an efficient and visually lossless foveated rendering.

The seminal initial work on foveated rendering [Guenter et al. 2012] addresses these problems by computing three gaze-centered concentric rings at progressively lower resolution. The resolution falloff and the size and the placement of the rings are chosen according to the perceptual detectability threshold [Geisler and Perry 1998]. To suppress the flicker in the periphery, jittering and temporal filtering are applied. Recent notable follow-up work performs sparser rendering in the periphery with either stochastic sampling and inpainting [Stengel et al. 2016] or using reduction of shading rate [He et al. 2014] followed by advanced image filtering [Patney et al. 2016]. Sun et. al [Sun et al. 2017] proposed a method for both foveation as well as accommodation for light field displays using a sparsely and adaptively sampled light field. In contrast, our work does not require rendered content because it takes color-only samples as input and is focused on foveation-only reconstruction. Vlachos [2015] proposed to use a checkerboard pattern to skip 2×2 pixel blocks to sparsify rendering in the periphery followed by a simple inpainting algorithm for hole filling. Foveated ray tracing [Weier et al. 2016] combines reprojection with temporal filtering to avoid artifacts. We refer an interested reader to the recent survey on perceptual rendering [Weier et al. 2017].

In contrast to most foveated rendering methods, we design a foveated reconstruction method that does not require any knowledge about how the image was generated, such as rendering-specific attributes, or a decomposition into visibility and shading. Instead, our method is inspired by the compression and inference in human visual system that is crafted to rely on natural video statistics. This allows us to design a single method for both synthetic content as well as regular videos and images. To avoid perceptual artifacts in the periphery, we rely on in-hallucinating the video content based on the learned statistics of natural videos to achieve high quality foveated compression.

2.3 Foveated Video Compression

For decades, video compression standards have emerged from incremental changes to the hybrid video encoder model. H.264, H.265 (HEVC), VP9, and AV-1 are the most popular standards used by media platforms. Unfortunately, these standards do not apply directly to foveated video compression. Attempts at applying various modifications to the input signal to exploit these encoder approaches include [Lee et al. 2001; Wang et al. 2001, 2003]. The recent work in applying convolutional neural networks (CNNs) to compression has yielded Wave One [Rippel et al. 2018], a highly efficient compression method that exploits non-constrained latent space for better compression.

360° video formats are becoming increasingly popular with the advent of fisheye cameras and HMDs. By exploiting both eye and head tracking, only the viewed portion of the entire scene needs to be decoded at full resolution. A fully functioning real-time foveated compression system has the potential to impact how 360° video formats are evolving, which is an active area of discussion within the Joint Video Exploration Team (JVET) [Ye et al. 2017].

2.4 Image and Video Error Metrics

Image and video quality metrics, such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [Wang et al. 2004], Video Quality Metric (VQM) [Pinson and Wolf 2004], and Spatio-Temporal Reduced Reference Entropic Differencing (ST-RRED) [Soundararajan and Bovik 2013] have been successfully predicting human subjective performance in the fovea [Liu et al. 2013]. Recent work on Learned Perceptual Image Patch Similarity (LPIPS) [Zhang et al. 2018] uses the calibrated perceptual metric, which passes the reconstructed image and the target image through a VGG network [Simonyan and Zisserman 2014] pretrained on ImageNet dataset. This recent metric was demonstrated to have an excellent perceptual performance on various distortions. We employ this metric both as a training loss and as one of the image metrics for ablation studies.

Only a few quality metrics were designed for foveated image quality assessment. Foveated Wavelet Image Quality Index (FWQI) [Wang et al. 2001] computes a multiscale Discrete Wavelet Transform (DWT) on images, weighs coefficients using a frequency and eccentricity dependent CSF, then pools the result using a ℓ_2 norm. Foveation-based content Adaptive Structural Similarity Index (FA-SSIM) [Rimac-Drlje et al. 2011] first weighs SSIM by a CSF that depends on frequency, eccentricity, and retinal velocity then averages these weighted coefficients. Swafford et. al [2016] extends HDR-VDP2 [Mantiuk et al. 2011] with the eccentricity-dependent CSF and a cortical magnification term.

2.5 Neural Denoising, Inpainting and Reconstruction

Machine learning models have been successfully used in a wide range of image processing tasks. The most common model design is a convolutional neural network (CNN), which is a feedforward network with a cascade of convolution layers [Lecun et al. 1998]. Such networks are able to efficiently analyze the image and build a hierarchy of multiresolutional semantic features that are trainable for specific tasks. Following recent efforts towards stabilizing training for deep networks [Krizhevsky et al. 2012], CNNs have been able to achieve impressive results in many areas of image processing, such as object localization [Girshick et al. 2014], image denoising [Schmidhuber 2015], inpainting [Pathak et al. 2016] and superresolution [Ledig et al. 2017]. We refer an interested reader to the survey [Schmidhuber 2015]. Residual networks [He et al. 2016]

reformulate the problem of learning a function to learning a delta between the input and the output. This change allows better gradient flow, often leading to better convergence and output quality.

Recurrent networks are often used for video processing tasks. They maintain temporal context by conditioning the current frame on the previous frames. This allows the model to exploit correlation across frames. Multiple types of recurrent networks are used from simple architectures [Chaitanya et al. 2017] to Long Short-term Memory networks (LSTM) [Hochreiter and Schmidhuber 1997].

2.6 Learning the Manifold of Natural Images and Videos

High quality images and video follow regular natural scene statistics [Ruderman 1994]. The human visual system has adapted to expect these statistics [Geisler 2008] and heavily relies on it when inferring the peripheral details. As a result, learning these statistics can enable more powerful perceptual compression methods.

Generative adversarial networks (GAN) [Goodfellow et al. 2014] can learn complex distributions, such as a manifold of natural images or videos, by combining a generator with a trainable adversarial loss, implemented using another network called a discriminator. This trainable loss has enough capacity to learn extremely high-dimensional distributions of data, such as the distribution of natural images or videos. The discriminator plays a minimax game with the generator network by learning to distinguish between the samples from the generator's distribution and real data samples.

Due to the inherent unstable equilibrium of the minimax game, the training process for adversarial networks is unstable and sensitive to hyperparameters. For example, if there is a significant capacity imbalance between the generator and the discriminator networks, the training can collapse with a trivial win of one network. Regularization and training improvements have improved the training robustness and stability. The Wasserstein GAN [Arjovsky et al. 2017] redefines the adversarial training problem as a simultaneous optimization of the generator and the Wasserstein-1 measure (also known as an earthmover's distance) represented by a discriminator network (also called critic). This new measure stabilizes the training by providing a smoother distance function between target and learned probability densities. It allows the generator to progress in training even when the discriminator has advanced further in training, avoiding training collapse. One recent improvement to Wasserstein GAN is called a Spectral Normalization GAN (SN-GAN) [Miyato et al. 2018] and it imposes the required Lipschitz continuity on the Wasserstein measure, while relaxing the restrictions on the underlying discriminator network and thus allowing for efficient training.

GANs have recently been used for large-scale single-image inpainting [Liu et al. 2018], high-resolution image generation [Karras et al. 2018], and generation using patch classification (PatchGAN) [Li and Wand 2016].

Recent advances in learning video manifolds with GANs demonstrate the potential of generating temporally coherent video results. Similar to human perception, generative networks can inpaint large portions of the video by learning high-level semantics and motion dynamics around the missing video fragment. For example, recent work [Wang et al. 2018] shows the feasibility of generating realistic, stable video from segmentation masks. A nested network design is used with background-foreground separation and an adversarial loss on optical flow. Another work [Pérez-Pellitero et al. 2018] introduces a recurrent network design for temporally stable video superresolution, which is trained using a variant of optical flow loss that promotes temporally consistent movements. This is achieved by comparing the current generated frame to a previous generated frame after warping according to an estimated optical flow. By contrast, a video-to-video retargeting work [Bansal et al. 2018] achieves temporally consistent results without optical flow. The method is able to generate stable frame sequences by requiring the result video to match the original after being mapped back and forth between different domains.

We employ recent advances in GANs and introduce two adversarial losses to train DeepFovea reconstruction network to reconstruct missing details in the periphery according to the learned statistics from the manifold of natural videos.

3 PROBLEM SETTING

In rendering systems, each pixel requires a high amount of computation. To reduce this workload, we draw a tiny subset of the total number of required pixels each frame and infer the rest with our model. Video captured from both the real world and realistic renders follow strong statistical regularities known as natural scene statistics [Kundu and Evans 2015; Ruderman 1994]. The human visual system is also adapted to comprehend real-world imagery that naturally possesses these statistics [Geisler 2008]. This provides a great opportunity for compression by relying on the statistics that form the manifold of all natural videos.

3.0.1 Sparse Input. To reduce the number of bits required to encode a signal, we subsample each frame using a sparse randomized mask. By reducing the number of samples in the mask, the compression rate directly increases. By shaping this mask according to the cell density layout of the retina, we can perceptually allocate bits.

For each pixel position $\mathbf{x}$ of a source video frame, we first compute the sampling rate $R(\mathbf{x}) \in [0; 1]$ based on the maximum perceptible frequency, the geometric setup of the display, and the desired compression rate. Please see supplementary material for more details.

For each video frame, our foveated sampling procedure fills an NxM binary mask, $\mathbb{M}$, according to $\mathbb{M}(\mathbf{x}) = \mathbb{1}_{R(\mathbf{x})>\mathcal{U}}$, where $\mathcal{U}$ is a random variable bounded $[0, 1]$, which can follow some uniform random distribution. In the spirit of Mitchell [1991] and to better follow the distribution of retinal cones [Cook 1986], we use a low-discrepancy blue noise sequence (see Figure 1), using the void and cluster algorithm [Ulichney 1993]. Valid pixels for a frame are then selected based on this mask, and the mask itself is provided as an input to reconstruction. We have also tested the network with other sampling patterns, including uniform random sampling. The network is largely agnostic to the sampling pattern, however, the reconstruction quality degrades.

Importantly, the mask is sampled independently at every frame, so the network can accumulate more context over time.

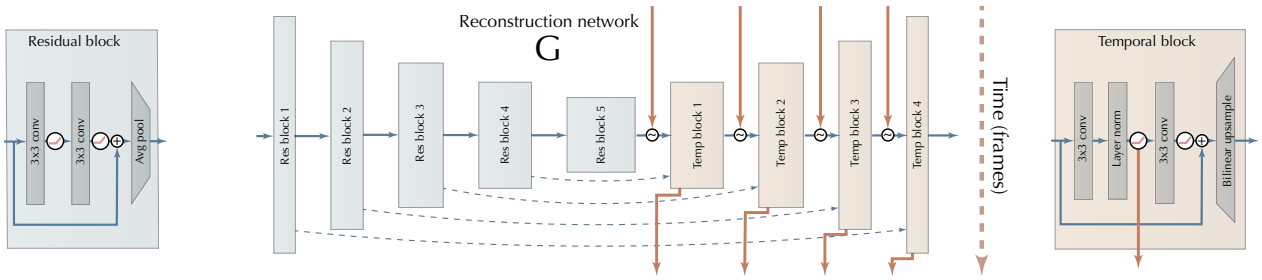


Fig. 3. The network design used for video reconstruction is a recurrent video encoder-decoder network architecture with skip connections (based on U-Net). The decoder part is modified to be stateful and hierarchically retains temporal context by concatenating (denoted with ~) recurrent connections (orange).

3.1 Reconstruction Methodology

Let $X = \{x_1, x_2, ..., x_K\}$ be a sequence of K video frames, where $X \in \mathbb{R}^{N \times M \times K}$. Let $\mathbb{M} = \{\mathrm{m}_1, \mathrm{m}_2, ..., \mathrm{m}_K\}$ be a sequence of binary masks described in the previous section. We produce a sampled video $\hat{X} = \{\hat{x}_1, \hat{x}_2, ...\hat{x}_K\}$ by applying each mask to a corresponding source video frame as $\hat{X} = X \odot \mathbb{M}$. The goal of the network G we train is to learn to approximate the mapping $\hat{X} \mapsto X$ by leveraging the large prior of the natural video manifold.

Our approach to the problem of sparse reconstruction is based on a framework of generative adversarial networks, which was recently shown to be able to learn large high-dimensional manifolds [Karras et al. 2018]. Note that in contrast to generative networks, the input to our network is not a random variable. The reconstruction network design is based on a popular U-Net encoder-decoder architecture. To allow the network to make use of inter-frame correlations, we add recurrent layers to the decoder part of the DeepFovea network. We use various techniques to stabilize network output in the temporal domain, such as optical flow and temporal regularizations. Since the ultimate goal of this network is to learn the projection from sampled sparse video to a manifold of natural videos, we train it on a large set of real-life videos. We discuss details of the DeepFovea algorithm in the subsequent section.

3.2 Design Goals

There are several goals that we would like to achieve with our method. First, the DeepFovea network should be able to operate in an online mode, i.e., it should be able to reconstruct the current frame based only on the past frames. Second, since we are targeting gaze-contingent display systems, the network should be able to operate in real time. This prohibits using complicated models or any significant number of past or future frames.

There are also strict requirements for output quality. The human visual system is not sensitive to high-frequency details in the periphery, however, motion and flicker are easily detectable. Therefore, while the peripheral reconstruction can omit fine details, it should not introduce significant noise to achieve plausible results with high compression. Given the uncertainty of the sparse video input, the network needs to balance between introducing the new content timely and suppressing flicker due to the inbound noise.

3.2.1 Causal Temporal Network with Recurrence. In order to leverage the temporal redundancy of the video and at the same time achieve higher temporal stability of the reconstruction, we employ a recurrent convolutional network architecture. This retained state is then used at the next frame, allowing the network to super-resolve the details through time (Figure 3). A common alternative approach, early fusion, feeds a network a sliding window of L last frames, however, it does not meet our performance requirements.

3.2.2 Performance Considerations. If the method is used for gaze contingent reconstruction, it has to exhibit under 50ms of latency for each frame in order to be unnoticeable for human vision [Guenter et al. 2012]. Moreover, for head-mounted displays, the method has to run at HMD's native refresh rate and high resolution to avoid motion sickness and provide a comfortable experience. For many existing VR HMDs the minimum refresh rate is 90Hz.

4 NEURAL RECONSTRUCTION

4.1 DeepFovea Network Design: Recurrent U-Net

For the reconstruction network G of our system (Figure 3), we chose the U-Net encoder-decoder design with skip connections [Ronneberger et al. 2015]. It transforms an image into a hierarchy and skip connections allow to bypass high frequencies and improve the gradient flow during training.

Each decoder block does the reverse of an encoder block, performs a spatial bilinear upsampling, while decreasing the feature count correspondingly to the symmetric encoder block. The input to a decoder block is the upscaled output of the previous decoder block concatenated with the output of the corresponding encoder block (skip connection, dashed arrows in Figure 3).

We use ELU activation function [Clevert et al. 2016] in all networks and layers (including recurrent and discriminator layers) to accelerate the training.

4.1.1 Recurrence. In order to generate temporally stable video content, the network needs to accumulate state through time. Moreover, a temporal network is able to super-resolve features through time and can work with sparser input while achieving the same quality. However, our network has to be causal (i.e., cannot see the future video stream) and should have a compact state to retain over time due to high video resolution and performance constraints. Complex recurrent layers like LSTM [Hochreiter and Schmidhuber 1997] have a large state and are computationally demanding. Therefore, in the spirit of Chaitanya et. al [2017], we employ a recurrent modification of the U-Net design with a a simple convolutional recurrent layer. A hidden state h in this layer is an output from the previous time step, i.e., for ith decoder block $o_i = h_i = f(x, h_{i-1})$ (orange arrows in Figure 3). ELU activation gives more freedom to the recurrent

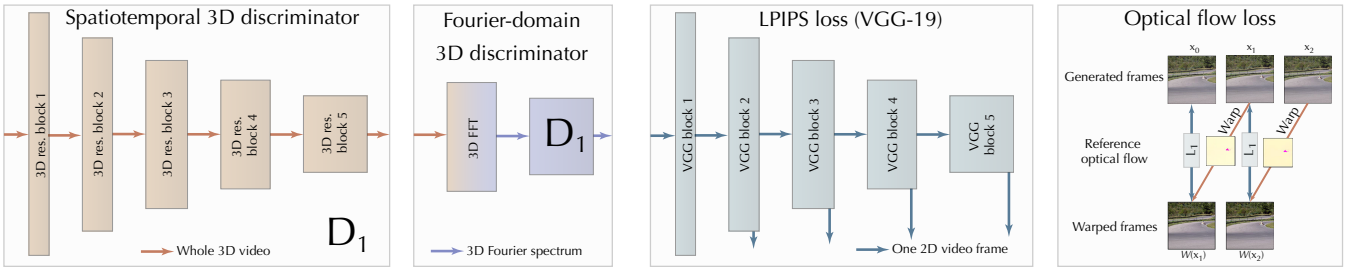


Fig. 4. Different losses we use to train the DeepFovea network to better learn video statistics and reconstruct plausible and temporally consistent videos.

layer compared to bounded activations (such as sigmoid), however, it potentially allows to have a recurrent filter with an unbounded positive feedback. Therefore, extra caution needs to be taken when training these recurrent layers to be stable on very long videos. We apply additional regularizations to recurrent connections to make the network trainable, as described in Section 4.3.

Recurrent blocks are able to handle sudden temporal changes of pixel density from gaze movements. It is sufficient to train the network with constant-density videos, while stochastically sampling the density for each video in the minibatch. In high-density regions, the temporal hidden state stores high details. When the gaze leaves the region, the recurrent blocks smoothly degrade the high-detailed information over time in this region and simultaneously update the temporal representation to be consistent with the sparse inflow of pixels. The fovea reconstruction is always high quality and not affected by the change of the density, because all valid pixels from the input frame bypass the reconstruction and are sent directly to the output frame. This bypass also forces the network to learn to eliminate any temporal lag in the reconstruction during dynamic gaze conditions, e.g., at the end of a saccade.

Convolutional recurrent blocks cannot efficiently move the content laterally to large image-space distances, because they are limited by the receptive field of their kernels. We reproject the hidden state to compensate for the large-scale lateral motion to assist the network with the head rotation in the HMD setup. We analytically calculate an optical flow for each recurrent connection using two view matrices for the last and the current frames to perform the hidden states re-projection. We treat each hidden state as a texture with multiple channels. We take each texel's 2D coordinates in hidden state and project them back to the camera rotated view space. To do so we calculate the product of inverted view (rotation only) and projection matrices used to render the current frame assuming the content is at infinity. Then we project coordinates back to the screen space using view and projection matrices used to render the last frame. Now having two 2D texel coordinates, last and current, we copy texel's data from the last to the current location. If the last location is outside of the hidden state 2D bounds then the current texel value is preserved.

Our choice of recurrent design, while being dictated by performance considerations, leads to a lightweight network that is able to efficiently accumulate and retain temporal context from a sparse video at multiple scales.

4.2 Losses

We optimize the generator network G with respect to a weighted sum of three losses (see Figure 4), namely, adversarial loss, perceptual spatial loss (LPIPS), and optical flow loss for temporal dynamics:

$$L_G = w_{\text{adv}} \cdot L_{\text{adv}} + w_{\text{LPIPS}} \cdot L_{\text{LPIPS}} + w_{\text{flow}} \cdot L_{\text{flow}}.$$

4.2.1 Adversarial loss. Adversarial loss is modeled by a discriminator network. The discriminator allows to learn the spatiotemporal manifold of natural videos by providing a boundary between a distribution of interest and the rest of possible videos. The discriminator - in contrast to the generator - processes the entire video sequence at once and can therefore reason about space-time relations and analyze the spatiotemporal dynamics. The goal of the discriminator is to classify videos into fake (constructed by the generator) and real (sampled from the dataset).

We use a Wasserstein GAN (WGAN) design [Arjovsky et al. 2017], which stabilizes the training due to its robust loss function. We use a 3D convolutional network D_1 as a Wasserstein measure (see Figure 4) with recent Spectral Normalization GAN (SN-GAN) [Miyato et al. 2018] to ensure 1-Lipschitz continuity. SN-GAN enables fast training on videos, while providing more stable adversarial training.

The network D_1 has a 3D funnel structure and consists of residual blocks [He et al. 2016] with decreasing spatial size. The network operates on the whole video as an input. In order to enable full analysis of spatiotemporal features, we employ 3D convolutional layers with $3 \times 3 \times 3$ spatiotemporal kernels. Each block contains two 3D convolutions, followed by a 3D average pooling operation that averages both spatial dimensions and the temporal one. We use ELU as activation functions to allow the discriminator to recover from sparsity, which reduces chances of training collapse. To focus the network on fine details, instead of reducing the video to a single scalar value, we follow a PatchGAN loss [Isola et al. 2017] and require the network to classify local patches of generated videos.

4.2.2 Spectral normalization. An inherent assumption of WGAN design is that the discriminator should be 1-Lipschitz continuous, i.e., $\forall x_1, x_2 : |f(x_1) - f(x_2)| \leq |x_1 - x_2|$. Standard networks generally violate this constraint. There are several approaches to ensure 1-Lipschitz continuity. We use recent Spectral Normalization in the discriminator [Miyato et al. 2018] that bounds the matrix spectrum of each layer's weights. This approach allows for fast training, which is crucial for training video networks, while leading to comparable results with other state-of-the-art methods.

4.2.3 Fourier-domain Discriminator. It is well known that the natural images have a characteristic statistics of a vanishing Fourier spectrum. Natural videos also obey a similar natural spectral statistics [Kundu and Evans 2015]. Choi and Bovik [2018] introduce flicker detection in 3D Fourier domain. In the same spirit, to help the discriminator to learn the intricate relations between spatial features and their natural motions, we introduce the second network in the adversarial loss that learns the manifold of the *spatiotemporal spectra* of natural videos. For that, we first Fourier-transform the whole input video into its 3D spectrum. Then we use another discriminator network with the same design as D_1 to learn the spectral manifold of natural videos. Since there are no image patches anymore, we append two fully connected layers with 256 and 1 unit correspondingly, with one ELU activation in between. This helps to learn the structure of spatiotemporal frequencies that occur in natural videos. Particularly, this loss helps detecting unnatural noise and flicker.

4.2.4 Perceptual Spatial Loss. To promote similarity of each reconstructed frame to its source frame, some measure of similarity is needed. Per-pixel L_1 loss is too low-level and prescriptive.

Instead, we use the calibrated perceptual loss (LPIPS) [Zhang et al. 2018]. By minimizing LPIPS, our network learns to endow each reconstructed frame of the video with natural image statistics. This also bootstraps the adversarial training, while providing enough freedom to the reconstruction. A pretrained VGG-19 consists of five blocks, each of which corresponds to a different level of abstraction of the initial image. We take outputs of the $conv_2$ layer from each block to use as feature extractors:

$$L_{\text{LPIPS}}(x_1, x_2) = \sum_{i=1}^{5} \|conv_{i,2}(x_1) - conv_{i,2}(x_2)\|_1$$

Unfortunately, this loss improves only spatial (intra-frame) features, while providing no temporal relation between frames. For peripheral video quality, it is more important to enforce temporal coherency. To make it cooperate with spatiotemporal losses and encourage the gradient flow through recurrent connections, we exponentially downweigh this loss for the first eight frames of the video. This loss corresponds well with human perception [Zhang et al. 2018] and gives enough freedom to the network.

4.2.5 Optical flow loss. We use optical flow loss to stimulate temporal consistency across frames and disentangle the spatio-temporal correlation of video frames. There are multiple ways to employ the optical flow in video generation. One is to estimate the optical flow directly in the generator and require the generator to match the target optical flow, as well as match the ground truth picture with the warped image [Wang et al. 2018]. However, this adds complexity to the network and does not meet our inference performance constraints. Our methodology here is inspired by the recent work on video super-resolution [Pérez-Pellitero et al. 2018], where the optical flow is applied only during training by requiring the network to match reconstructed frame with previous reconstructed frame, warped by the known optical flow W as $L_{\text{flow}}(\hat{x}_i, \hat{x}_{i-1}, W_{(i-1)\to i}) = \|\hat{x}_i - W_{(i-1)\to i}(\hat{x}_{i-1})\|_1$. Here, $W_{(i-1)\to i}(\cdot)$ is the warping operator that applies optical flow to reproject pixels of the frame $i-1$ to the frame i.

This indirect approach encourages the network to retain consistent content and smooth movements over time, while not prescribing any particular spatial content.

4.3 Training Details

4.3.1 Network Parameters. There are five encoder blocks in our network. Each consecutive encoder block downscales the input spatial dimensions twice and increases the feature count (Figure 3). An encoder block consists of two 3x3 convolutions with ELU activations. The second convolution layer is followed by an average pooling layer. Both convolution layers in a block have the same number of filters (32-64-128-128-128 for each block, correspondingly). The bottleneck block processes the output of the last encoder layer with a low spatial resolution and operates on high-level image semantics. It is identical to the last encoding block, except that it upsamples the input and has no skip connection.

Each decoder block consists of a 3x3 convolutional layer with a recurrence (see next paragraph), followed by the second spatial 3x3 convolution layer, and a bilinear upsampling layer. Each layer is followed by an ELU activation. The output of the recurrent layer undergoes a layer normalization before activation. Decoder blocks have the same number of convolution filters as the corresponding encoder blocks (128-128-128-64-32). Symmetric padding is used everywhere to prevent boundary artifacts on the image border.

4.3.2 Video Dataset. We train on videos sampled from a video dataset [Abu-El-Haija et al. 2016] that contains a variety of natural content such as people, animals, nature, text, etc. Each video has resolution up to 640x480 and up to 150 frames. For each video, we precompute the optical flow using the FlowNet2 network [Ilg et al. 2017]. Next, the video is downsized to 128x128 to meet GPU memory restrictions. Lastly, the videos are sliced into 32-frame-long chunks with an overlap of 8 frames. The total number of video sequences in the training set is about 350,000. During training, each 32-frames video segment is corrupted with a stochastic binary mask, which is generated with the same method as for the final reconstruction, and the location on the retina is randomly sampled.

4.3.3 Training Hyperparameters. Training follows a standard adversarial approach of interleaving updates to generator and discriminator, with one update for each network. The network G starts to train with 10% valid pixels in each frame, and this percentage is gradually decreased to 1% during the first two epochs. It allows the network to learn quicker in the beginning of the training process. We weigh the losses as $w_{\text{adv}} = 1, w_{\text{LPIPS}} = 100, w_{\text{flow}} = 20$ to roughly equalize their magnitudes. We use ADAM optimizer [Kingma and Ba 2014] with $\beta_1 = 0, \beta_2 = 0.95$ for 30 epochs and learning rate 3e-4. For training a video network, we implemented a parallel distributed training. Training for 30 epochs takes 48 hours on 7 NVIDIA DGX-1 nodes with 8 GPUs each. Batch size is chosen to be 56, corresponding to one video per GPU.

4.3.4 Stabilizing a Recurrent Video Network. We found that the network is unstable on long videos during the testing phase by collapsing into a constant color video. This collapse occurs from the unbounded positive feedback loop within the recurrent connections. We use several techniques to stabilize the recurrent connections.

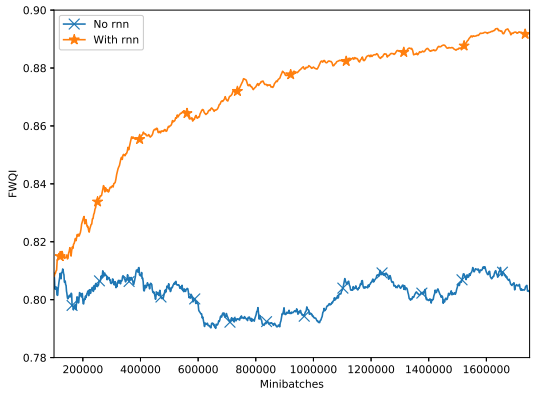


Fig. 5. Ablation results on the recurrent blocks.

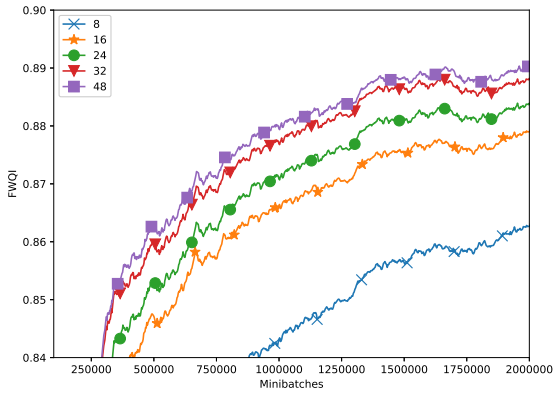


Fig. 6. Ablation results on network capacity. FWQI metric shows the difference in spatial quality after 30 epochs. Different capacity is defined by a number of filters in the first layer and a doubling every block.

First, we apply layer normalization [Ba et al. 2016] to recurrent layers, which helps keep the activations bounded. Second, we train RNNs in a stateful manner, i.e., the hidden state is retained from one minibatch to another. This allows each recurrent layer to start with a reasonable content in hidden activations that lies within the current working range during training, which helps the network to remain stable during very long videos at inference time. At the very beginning of the training, we initialize the hidden state using zero-mean Gaussian noise with $\sigma = 0.02$. These improvements help the stability of the recurrent design by preventing activation explosion during inference.

4.4 Ablation study

To validate the design choices made for the network, we conducted an ablation study. We analyze the network capacity, depth, as well as the contribution of that loss to the final result. We use FWQI metric to determine the spatial quality of reconstruction. Unfortunately, since FWQI detects only artifacts of spatial reconstruction in a single frame, it is not helpful to measure the temporal artifacts in peripheral vision, such as flicker, which is of utmost importance for peripheral reconstruction quality. To the best of our knowledge, there is no peripheral spatiotemporal video quality metric, therefore, in order to assess temporal reconstruction quality we provided sample videos in the supplemental material.

4.4.1 Network depth. Our experiments show that the network benefits from increasing the number of UNet blocks. The FWQI value first increases sharply from 1 to 3, and then plateaus from 3 to 5 blocks. All networks have similar number of parameters (around 3M). One explanation is due to the sparse nature of the input, the network benefits from the increase in receptive field. We use 5 levels in the final design.

4.4.2 Network capacity. The number of filters follows a pattern of doubling every layer with a cap of 128 filters, so we provide only the number of filters in the first layer. Figure 6 shows that FWQI increases when increasing the filters from 8 to 16. The metric keeps a steady increase for values of 24 and 32, while plateauing at 48 features. In order to constrain the network's inference performance, we choose 32 as the final setting.

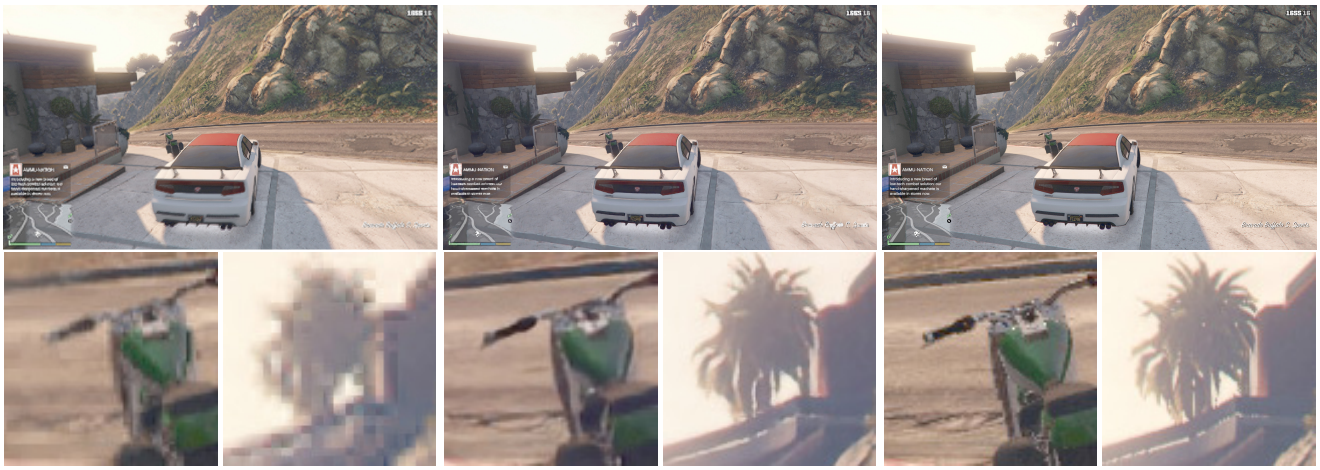
4.4.3 Recurrent blocks. We show that recurrent blocks are essential for the reconstruction with the sparse input. As shown in Figure 5, recurrent design significantly outperforms the non-recurrent one as measured by FWQI. We also demonstrate this result in the supplementary video for subjective quality assessment. The non-recurrent network works on a single sparse frame and is not able to accumulate additional details from previous frames, therefore, introducing a significant amount of temporal noise even on high levels of input density.

4.4.4 Losses. To validate that each of our losses improves the reconstruction, we compared multiple variants of the network with losses being enabled one after another as L_{LPIPS} , $L_{\text{LPIPS}} + L_{\text{adv}}$, $L_{\text{LPIPS}} + L_{\text{adv}} + L_{\text{flow}}$. Unfortunately, FWQI does not provide a meaningful comparison and can even decrease during this process. However, when observed, the video quality improves with each added loss. The reason is because FWQI does not account for temporal stability, which is the target of L_{adv} and L_{flow} losses. We provide videos in the supplementary to demonstrate the improvements.

Unsurprisingly, L_{LPIPS} allows the network to learn a only a single-frame reconstruction, leaving a substantial amount of flicker. The adversarial loss L_{adv} significantly improves the temporal stability and suppresses a large portion of flicker. Optical flow loss L_{flow} provides an additional improvement and reduces temporal noise, such as pixel crawling, especially in case of long lateral camera movements. Please refer to the accompanying video for comparison.

5 USER STUDY

Our design of the reconstruction network and its training is motivated by the reconstruction process in human visual system that is based on the natural video statistics. However, in order to validate our method, we conduct an extensive user study. We compare DeepFovea to the Multiresolution [Guenter et al. 2012] foveated rendering method, and to the baseline with Concentric H.265 compression. We use these two methods, because, unlike many foveated rendering methods, they do not require any additional attributes (such as surface normals, or semantics of the geometry). While



(a) Multiresolution (b) DeepFovea (c) Original (H.265)

Fig. 7. A frame from the middle of GTA video at the moment of extreme motion. Both Multiresolution and DeepFovea are configured for 16x compression. The gaze is in the center of the image. Our method is able to reconstruct more details under the same compression rate, while causing less flicker and ghosting than Multiresolution. Note that it is not possible to reason about the peripheral quality using your foveal vision.

our method and Multiresolution can run in gaze-contingent mode, Concentric H.265 method is too slow for this, so we precomputed various compression layers and composited them concentrically in runtime to allow gaze contingency in our experiments. First, we measure artifact detectability as a function of sampling rate using a method of constant stimuli. Second, we measure subjective quality ratings for each method. This study helps us understand whether detectable artifacts at a given sampling rate are also objectionable in terms of quality. We run our experiment on two display setups: a large projector screen and a head mounted display (HMD).

5.1 Screen Experimental Setup

We use a Digital Projection X-Vision WQXGA projector in a rear-projection mode set to 120Hz refresh rate and a screen 243x152cm with resolution 1920x1080 pixels resulting in 78° horizontal field of view. Participants are seated at 150cm distance from the projector screen, which provides 0.34 pixels per arcminute, which is sufficient for evaluating peripheral quality. The participant's head is stabilized using a chin rest. For gaze tracking, we use EyeLink 1000 eye tracker that we calibrate and validate for every subject. Its refresh rate is 500Hz, the latency is 5ms and the accuracy is 1°. We asked every participant to try to look closer to the center of the projector screen to preclude tracking imprecision at extreme gaze eccentricities. The experiment is conducted in a dark room. We include the screen captures from the user study in the supplemental video under the "Reconstruction with Dynamic Gaze" section. In contrast with the real study, the videos are looped back and forth to increase their duration for the viewer.

5.2 HMD Experimental Setup

We use an HTC Vive Pro HMD with resolution 1440x1600 pixels per eye, 110° field of view, and refresh rate 90Hz with 0.28 pixels per arcminute. We use a stock Pupil Labs eye tracker compatible with HTC Vive with 200Hz refresh rate, latency of 5ms and an accuracy of 1°. We calibrated and validated the eye tracker for every participant. We asked every participant to try to keep their head stable, keeping it roughly straight ahead, and try to avoid extreme gaze eccentricities.

5.3 Stimuli and Methods

We used ten diverse video contents for the screen experiment from [Bampis et al. 2018] including some freely available videos [Haglund 2006], as well as rendered game content. Ten 360° videos were obtained from Henry, a VR movie created by Oculus Story Studio, content creator Hugh Huo, and JVET 360 video test content for the HMD experiment. For both experiments, we rendered all of the foveated videos using one of three methods: DeepFovea, Multiresolution [Guenter et al. 2012] and Concentric H.265 compression. We tested five sampling rates for all three methods in the screen experiment. In HMD experiment, we tested three sampling rates for Concentric H.265 and five for other methods based on the results from the screen experiment. In each experiment, we also presented the full-resolution non-foveated videos. Figure 7 provides comparison across the methods.

5.3.1 DeepFovea. We run our method in real-time with both corruption and reconstruction. The corruption step is only needed for a video and subsamples it with a Sobol pattern. A rendered content can be directly provided in a sparse form. The subsampled frame is then given to the peripheral network G as an input. Upon reconstruction is done, the output of the network is presented to the user.

For HMD, we compensate for head motion by reprojecting the RNN hidden states with respect to the previous frame's rotation. This significantly helps the RNN layers to achieve temporal stability during panning motions of the video.

5.3.2 Multiresolution. The Multiresolution method [Guenter et al. 2012] identifies three radially concentric regions based on perceptually optimized parameters. Examples of these regions are provided in the supplementary materials. The exact placement and resolution of

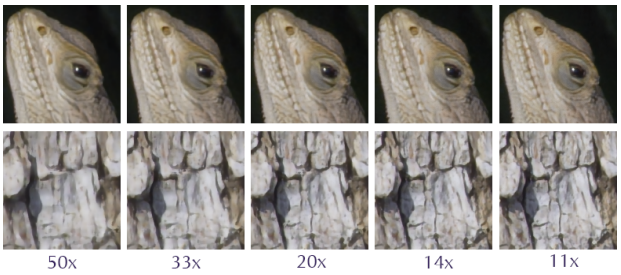


Fig. 8. Reconstruction in the same regions of Figure 1, shown with a variety of compression rates (left to right, corresponding to 2/3/5/7/9% of valid samples). Fine details degrade with higher rates.

these regions are found after optimizing against the minimum angle of resolution (MAR), which predicts acuity as a function of eccentricity. This optimization is sensitive to screen width, display distance, and horizontal screen resolution. For the display experiment, we set these parameters to 2.43 meters, 2.03 meters, and 1920px. Likewise, for the HMD experiment, we set these parameters to 0.02 meters, 0.012 meters, and 1542px. Following the work described in [Guenter et al. 2012], we find that the MAR slope $m = 0.0376$, downsampling range of region three $s_3 = 5.20$, radius of region one $e_1 = 3.25$, and radius of region two $e_2 = 9.35$ for our display. Similarly we find $m = 0.0697$, $s_3 = 6.79$, $e_1 = 3.25$, and $e_2 = 10.95$ for our HMD.

5.3.3 Concentric H.265 Compression. We use the H.265 video compression method as a reference to set a strong quality bar for our method. H.265 requires all pixels in a frame as input, and, moreover, it is allowed to look at all frames both in the past and in the future. The concentric H.265 compresses the same three concentric regions of interest identified in the Multiresolution method with the same values for e_1 and e_2. The maximum total frame bit budget, B, for a video is $B = b_{pp} * c_r * M * N$, where b_{pp} is the average number of bits per pixel, c_r is the compression ratio, and $M \times N$ is the screen resolution. The center region is encoded at 50Mbps, weighted by its area to achieve high foveal quality while still providing bits for the remaining two regions. The remaining bits in B are distributed to the middle and outer regions to minimize perceptual impact. We composite the three overlapping regions using a guardband with linear blending. See the supplementary for more details.

5.4 Procedure

For both the screen and HMD setup, participants perform the same experimental task. First, we calibrate the eye tracker and validate the tracking accuracy for each participant. In a given trial, the participants watch a 4 second video. Once the video is over, participants have to give two responses. First, they make a detection judgment about whether they detected artifacts in the video. They indicate their decision with a key press. This is followed by a subjective rating judgment where participants rate the visual quality of the video. They indicate their response on a continuous scale represented as a slider bar which is marked with three labels, "Bad" on the left end, "Excellent" on the right end and "Fair" in the middle. The participants move the marker on the slider bar by using the arrow keys. Once the participants are done with adjusting their rating, they hit space bar to record the response and continue. Participants are allowed to take as long as they wish to enter their response, however, once entered, they are not allowed to change them. In both the screen experiment participants repeat every trial three times, in HMD experiment they repeat every trial twice. The order of the trials is randomized for both experiments. The screen experiment has a total of 480 trials and takes approximately 1.5 hours to complete. The HMD experiment has 280 trials and takes approximately 1 hour to complete. Photos of the experimental setup and the task are shown in supplementary materials.

5.5 Participants

We had eight participants for screen and five participants for HMD experiment. All participants have normal or corrected to normal vision and no history of visual deficits. The participants were not aware of the experimental hypothesis. All participants provided written consent before taking part in the study. Methods were approved by an external institutional review board (IRB). Before each experiment, the participants were debriefed about the purpose of the study and the experimental task.

5.6 Analysis

5.6.1 Probability Plots. For each method we calculated average detectability probability across all participants as a function of compression rate. For each average data point we calculated 95% confidence intervals.

5.6.2 DMOS. We calculate Difference Mean Opinion Score (DMOS) from subjective rating data per video per sampling rate for both screen and HMD experiment. DMOS with normalization of subjective scores accounts for subjective variability and the perception of different content types. Seshadrinathan et. al [2010] describe in more detail the calculations for DMOS.

We also compute FWQI and FA-SSIM foveated quality metrics for each video. To analyze the correlation between DMOS derived from subjective ratings and the error metrics, we compute Spearman's Rank Correlation Coefficient (SRCC) and Linear Correlation Coefficient (LCC).

6 RESULTS

6.1 Inference Runtime Performance

The time to infer a FullHD frame on 4x NVIDIA Tesla V100 GPUs is 9ms. The DeepFovea model has 3.1 million parameters and requires 111 GFLOP with 2.2GB memory footprint per GPU for an inference pass. We implement a custom inference framework in C++ on NVIDIA CUDA 10 and cuDNN 7.6.1. All calculations are performed with 16-bit half floating point precision to leverage NVIDIA Tensor-Core computational units. Time to sparsify a frame on a single GPU is 0.7ms. We are able to achieve 90Hz in the HMD.

6.2 Numerical Quality Analysis

For each video per compression rate, we calculate two metrics, FWQI [Wang et al. 2001] and FA-SSIM [Rimac-Drlje et al. 2011], on the stimuli used in both screen and HMD experiments. Higher

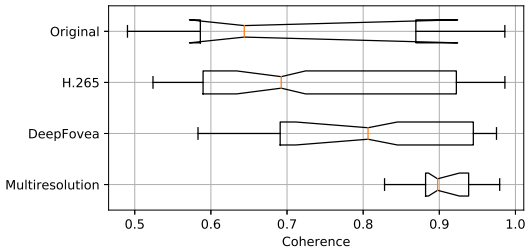


Fig. 9. Temporal coherence, computed by averaging the SSIM scores across all adjacent frames, for the reference, DeepFovea, H.265, and Multiresolution. Medians are in red and notches are 95% confidence intervals. DeepFovea yields reconstructions with similar temporal coherence to the Original, while Multiresolution yields higher coherence by temporal averaging.

values for FA-SSIM and FWQI indicate higher quality. Recall that for DMOS, lower values indicate higher quality.

We calculate Spearman's Rank Correlation Coefficient (SRCC), a measure of monotonic correlation, and the Linear Correlation Coefficient (LCC), a measure of linear correlation, to evaluate the predictive power between our subjective DMOS data and two foveated image quality metrics, FWQI and FA-SSIM.

For the screen study, we find a significant negative correlation between DMOS and FWQI with both SRCC and LCC ($r_s(150) = -0.838, p < 0.001$, $r(150) = -0.764, p < 0.001$). We find a smaller correlation between FA-SSIM and DMOS ($r_s(150) = -0.256, p < 0.01$, $r(150) = -0.261, p < 0.01$).

For the HMD study, the correlations between DMOS and FWQI are ($r_s(130) = -0.6278, p < 0.001$, $r(130) = -0.6163, p < 0.001$). We observed no significant correlation between DMOS and FA-SSIM in HMD. Therefore, DMOS has a significant negative linear and monotonic relationship with FA-SSIM for only the screen experiment and with FWQI for both the screen and HMD experiments. Scatter plots of these results are included in supplementary materials.

We averaged the SSIM computed between adjacent frames to measure temporal coherence. Figure 9 depicts the relative coherence across the three methods and original non-foveated sources for all videos. The coherence for DeepFovea is comparable to H.265, which is indistinguishable from the reference. Multiresolution has a high coherence, which is not surprising since it applies temporal averaging to enforce it.

6.3 User Study Results

Figure 10 shows a summary plot for the screen experiment. Please, refer to a Figure 7 in the supplementary materials for the HMD results. To compare the results from other methods to DeepFovea, we convert DeepFovea's sampling rate into compression rate by taking the reciprocal of the sampling rate. For Multiresolution, we count the number of pixels in each region and divide by the total to compute sampling rate, then use the same conversion process to obtain compression rate. The plot shows average detectability for all methods across the five compression rates tested. Additional plots per subject per experiment are provided in the supplementary.

Based on the results from the user study, DeepFovea achieves 50% detectability of artifacts at 37x compression rate. Overall, DeepFovea consistently out-performs Multiresolution across all sampling

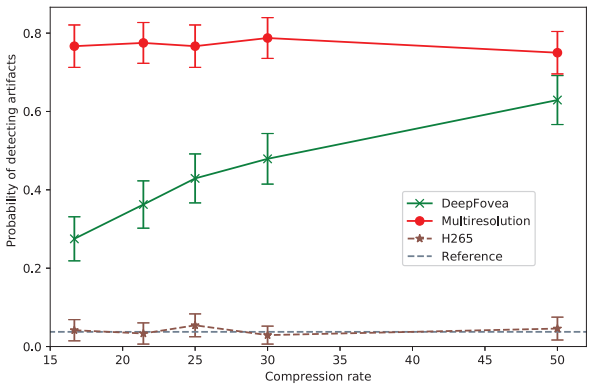


Fig. 10. A summary of detectability results from screen experiment. Green shows mean detectability for five compression rates measured for DeepFovea. Red shows Multiresolution. Dashed brown line shows H.265 and the dashed black line represents reference videos. The x-axis represents compression rate. Error bars represent bootstrapped 95% confidence intervals.

rates, suggesting that by comparison, DeepFovea performs significantly better, even in low compression rates. On the other hand, Concentric H.265 is not significantly different than reference across all sampling rates. Towards 15x compression, DeepFovea gets closer to the detectability of H.265, hence reference. Despite the fact that H.265 uses 100% of the pixels to optimally encode to each compression rate, DeepFovea starts approaching H.265 and reference performance with only 9% valid pixels. For the HMD experiment, DeepFovea is always below 50% detectability. Overall, DeepFovea consistently outperforms Multiresolution across all compression rates similar to the screen experiment results. Similar to the screen experiment, H.265 is not significantly different than the reference. With the HMD, DeepFovea has a significant improvement in artifact detectability and becomes on-par with H.265, hence reference, at around 25x compression rate, which means that at this compression rate artifacts become, on average, undetectable.

The subjective rating scores are consistent with our findings from the detectability study. Figure 11 shows FWQI vs DMOS for the "iguana" video, which is representative of the rest of the content. Lower DMOS means higher visual quality rating. In conclusion, higher compression rates reduces visual quality, which is reflected by the higher DMOS scores. A more comprehensive breakdown is provided in the supplementary materials.

The DMOS from both the screen and HMD experiments show that DeepFovea produces significantly better results than Multiresolution. When considering each content individually, the subjective ratings indicate that DeepFovea often produces better visual quality when compared to the Multiresolution method. DeepFovea uses a large learned prior for temporal reconstruction from the sparse video, suppressing flickering and ghosting artifacts. Therefore, it can achieve better results compared to the Multiresolution method.

6.4 Limitations and Future Work

While our method can be used for foveated rendering with superior compression quality, it can benefit from additional specialized knowledge about the content. For example, high-level information, such as the type of content (real/rendered/text/drawings etc.), or

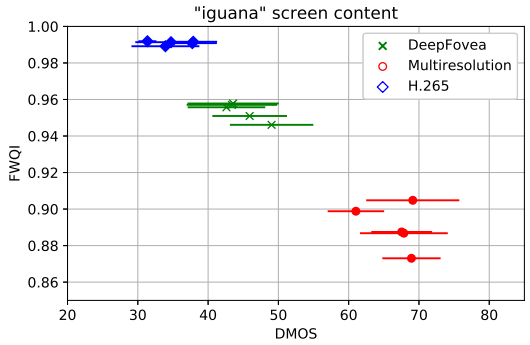


Fig. 11. Screen experiment on IGUANA video. X axis shows DMOS (lower is better) and Y shows FWQI prediction with 95% confidence intervals. Each point is a method and sampling rate averaged across all users. This shows how the quality distribution shifts across sampling rates for each given method. DeepFovea often overlaps with H.265 for high sampling rates.

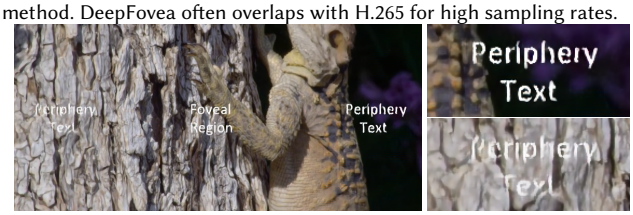


Fig. 12. Failure case showing reconstruction of a text on textured vs non-textured background. The fixation point is at the center with peripheral text located 9° from fixation. The foveated sampling pattern is set to 9%.

object classes (e.g., ImageNet classes) or scene segmentation can improve the network ability to inpaint context-dependent details.

6.4.1 Unnatural Content. Because our method is trained on the natural video content, it performs best on natural and photorealistically rendered content. While, as any regression-based method, it performs fine on unnatural imagery, such as text, illustrations, and non-photorealistic rendering, such content certainly poses hard cases for the trained network. In particular, we noticed text reconstruction quality differs depending on the background upon which the text is rendered. Textured backgrounds negatively affect the reconstruction of fine details present in the text, producing flicker and ghosting artifacts. Figure 12 demonstrates this phenomenon.

6.4.2 Specialized Networks. Training a specialized network for common content and tasks, such as text reading, non-photorealistic content, user interface, non-natural content, computer animation, and so on, would allow to both decrease its size as well as improve the quality for the specific content type it is trained for.

6.4.3 Extended Input. We decided to start with sparse images, because they can be almost universally obtained from many existing systems and it poses a challenging problem for the network (hard to super-resolve details through the noise, and hard to achieve temporal stability). However, it is possible to change the input to the network. For example, it is possible to generate a more compact trainable latent embedding of the input video. Another option is to augment the rendered input with auxiliary scene attributes, which was shown to improve the reconstruction quality in similar settings [Chaitanya et al. 2017]. There is also existing work, where a network is trained to directly consume a compressed H.265 input.

6.4.4 Adaptive Sampling. In the spirit of existing work [Stengel et al. 2016; Weier et al. 2016], it is possible to opportunistically improve the reconstruction quality by allocating samples adaptively according to the visual importance. It can be based on saliency maps, frequencies of textures, object silhouettes, context, and tasks performed by the user. Machine learning based attention and saliency estimation methods can also guide adaptive sampling of image and video content, for example, with the prediction based on image and video classification networks, such as VGG-19 features.

7 CONCLUSION

We presented a neural reconstruction method for foveated rendering and video compression. We show that it is possible to leverage the spatiotemporal statistics of natural videos to achieve an efficient video reconstruction in the periphery. Our method demonstrates temporally stable reconstruction from a noisy input and sets a new bar of 14x compression rate in savings achievable for foveated rendering with no significant degradation in perceived quality. Because the method requires only color information as an input, it is also suitable for foveated compression of video content. We open our method for follow-up research on foveated reconstruction.

ACKNOWLEDGMENTS

We acknowledge Darrel Palke for helping to capture the videos and the runtime performance, Anjul Patney for helping with editing the videos, UT Austin, JVET, Oculus Story Studio, and Hugh Hou for the test videos, Google for the YouTube 8M video dataset, user study participants, and all reviewers for their valuable feedback.

REFERENCES

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. 2016. YouTube-8M: A Large-Scale Video Classification Benchmark. *CoRR* abs/1609.08675 (2016).

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 214–223.

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR* abs/1607.06450 (2016). arXiv:1607.06450 http://arxiv.org/abs/1607.06450

Christos Bampis, Zhi Li, Ioannis Katsavounidis, Te-Yuan Huang, Chaitanya Ekanadham, and Alan C. Bovik. 2018. Towards Perceptually Optimized End-to-end Adaptive Video Streaming. *arXiv preprint arXiv:1808.03898* (2018).

Aayush Bansal, Shugao Ma, Deva Ramanan, and Yaser Sheikh. 2018. Recycle-GAN: Unsupervised Video Retargeting. In *Proc. European Conference on Computer Vision.*

Chris Bradley, Jared Abrams, and Wilson S. Geisler. 2014. Retina-V1 model of detectability across the visual field. *Journal of vision* 14, 12 (2014), 22–22.

Chakravarty R. Alla Chaitanya, Anton S. Kaplanyan, Christoph Schied, Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, and Timo Aila. 2017. Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder. *ACM Trans. Graph. (Proc. SIGGRAPH)* 36, 4, Article 98 (2017), 98:1–98:12 pages.

Lark Kwon Choi and Alan Conrad Bovik. 2018. Video quality assessment accounting for temporal visual masking of local flicker. *Signal Processing: Image Communication* 67 (2018), 182 – 198.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *Conference on Learning Representations, ICLR* abs/1511.07289 (2016).

Robert L Cook. 1986. Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG)* 5, 1 (1986), 51–72.

Christine A Curcio, Kenneth R Sloan, Robert E Kalina, and Anita E Hendrickson. 1990. Human photoreceptor topography. *Journal of comparative neurology* 292, 4 (1990), 497–523.

Dennis M Dacey and Michael R Petersen. 1992. Dendritic field size and morphology of midget and parasol ganglion cells of the human retina. *Proceedings of the National Academy of sciences* 89, 20 (1992), 9666–9670.

Wilson S. Geisler. 2008. Visual perception and the statistical properties of natural scenes. *Annu. Rev. Psychol.* 59 (2008), 167–192.

Wilson S. Geisler and Jeffrey S. Perry. 1998. Real-time foveated multiresolution system for low-bandwidth video communication. , 3299 - 3299 - 12 pages.

Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proc. Conf. Computer Vision and Pattern Recognition* (2014), 580–587.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *arXiv e-prints* (2014), arXiv:1406.2661. https://arxiv.org/abs/1406.2661

Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D Graphics. *ACM Trans. Graph. (Proc. SIGGRAPH)* 31, 6, Article 164 (2012), 164:1–164:10 pages.

Lars Haglund. 2006. The SVT High Definition Multi Format Test Set. (2006). https://media.xiph.org/video/derf/vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat/SVT_MultiFormat_v10.pdf

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition* (2016), 770–778.

Yong He, Yan Gu, and Kayvon Fatahalian. 2014. Extending the Graphics Pipeline with Adaptive, Multi-rate Shading. *ACM Trans. Graph. (Proc. SIGGRAPH)* 33, 4, Article 142 (2014), 142:1–142:12 pages.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In *Proc. Conf. Computer Vision and Pattern Recognition*. http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *Proc. Conf. Computer Vision and Pattern Recognition* (2017), 5967–5976.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations*.

D. H. Kelly. 1984. Retinal inhomogeneity. I. Spatiotemporal contrast sensitivity. *JOSA A* 1, 1 (1984), 107–113.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *CoRR* (2014).

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems* 25 (01 2012).

Debarati Kundu and Brian L. Evans. 2015. Full-reference visual quality assessment for synthetic images: A subjective study. *IEEE International Conference on Image Processing (ICIP)* (2015), 2374–2378.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. 2017. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *Proc. Conf. Computer Vision and Pattern Recognition*. 105–114.

S. Lee, M. Pattichis, and A. C. Bovik. 2001. Foveated Video Compression with Optimal Rate Control. *IEEE Transactions on Image Processing* 10, 7 (2001), 977–992.

Chuan Li and Michael Wand. 2016. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. *CoRR* abs/1604.04382 (2016).

Guilin Liu, Fitsum A. Reda, Kevin Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. 2018. Image Inpainting for Irregular Holes Using Partial Convolutions. *arXiv preprint arXiv:1804.07723* (2018).

Tsung-Jung Liu, Yu-Chieh Lin, Weisi Lin, and C-C Jay Kuo. 2013. Visual quality assessment: recent developments, coding applications and future trends. *Transactions on Signal and Information Processing* 2 (2013).

Rafat Mantiuk, Kil Joong Kim, Allan G. Rempel, and Wolfgang Heidrich. 2011. HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. In *ACM Transactions on graphics (TOG)*, Vol. 30. ACM, 40.

Don P. Mitchell. 1991. Spectrally Optimal Sampling for Distribution Ray Tracing. *Computer Graphics (Proc. SIGGRAPH)* 25, 4 (1991), 157–164.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. *CoRR* abs/1802.05957 (2018).

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. 2016. Context Encoders: Feature Learning by Inpainting. In *Proc. Conf. Computer Vision and Pattern Recognition*.

Anjul Patney, Marco Salvi, Joohwan Kim, Anton Kaplanyan, Chris Wyman, Nir Benty, David Luebke, and Aaron Lefohn. 2016. Towards Foveated Rendering for Gaze-tracked Virtual Reality. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 35, 6, Article 179 (2016), 179:1–179:12 pages.

Eduardo Pérez-Pellitero, Mehdi S. M. Sajjadi, Michael Hirsch, and Bernhard Schölkopf. 2018. Photorealistic Video Super Resolution. *CoRR* abs/1807.07930 (2018).

E. Pérez-Pellitero, M. S. M. Sajjadi, M. Hirsch, and B. Schölkopf. 2018. Photorealistic Video Super Resolution.

Margaret H. Pinson and Stephen Wolf. 2004. A new standardized method for objectively measuring video quality. *IEEE Transactions on broadcasting* 50, 3 (2004), 312–322.

S. Rimac-Drlje, G. Martinović, and B. Zovko-Cihlar. 2011. Foveation-based content Adaptive Structural Similarity index. *International Conference on Systems, Signals and Image Processing* (2011), 1–4.

Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G. Anderson, and Lubomir Bourdev. 2018. Learned Video Compression. (2018).

J. G. Robson. 1966. Spatial and Temporal Contrast-Sensitivity Functions of the Visual System. *JOSA A* 56, 8 (Aug 1966), 1141–1142.

O. Ronneberger, P. Fischer, and T. Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI) (LNCS)*, Vol. 9351. 234–241.

Jyrki Rovamo, Lea Leinonen, Pentti Laurinen, and Veijo Virsu. 1984. Temporal Integration and Contrast Sensitivity in Foveal and Peripheral Vision. *Perception* 13, 6 (1984), 665–674.

Daniel L Ruderman. 1994. The statistics of natural images. *Network: computation in neural systems* 5, 4 (1994), 517–548.

Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural Networks* 61 (2015), 85 – 117.

K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. 2010. Study of Subjective and Objective Quality Assessment of Video. *IEEE Transactions on Image Processing* 19, 6 (2010), 1427–1441.

Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

Rajiv Soundararajan and Alan C. Bovik. 2013. Video quality assessment by reduced reference spatio-temporal entropic differencing. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 4 (2013), 684–694.

Michael Stengel, Steve Grogorick, Martin Eisemann, and Marcus Magnor. 2016. Adaptive Image-Space Sampling for Gaze-Contingent Real-time Rendering. *Computer Graphics Forum (Proc. of Eurographics Symposium on Rendering)* 35, 4 (2016), 129–139.

Qi Sun, Fu-Chung Huang, Joohwan Kim, Li-Yi Wei, David Luebke, and Arie Kaufman. 2017. Perceptually-guided Foveation for Light Field Displays. *ACM Trans. Graph. (Proc. SIGGRAPH)* 36, 6, Article 192 (2017), 192:1–192:13 pages.

Nicholas T. Swafford, José A. Iglesias-Guitian, Charalampos Koniaris, Bochang Moon, Darren Cosker, and Kenny Mitchell. 2016. User, metric, and computational evaluation of foveated rendering methods. *Proc. ACM Symposium on Applied Perception* (2016), 7–14.

Robert A Ulichney. 1993. Void-and-cluster method for dither array generation. In *Human Vision, Visual Processing, and Digital Display IV*, Vol. 1913. International Society for Optics and Photonics, 332–343.

Alex Vlachos. 2015. Advanced VR Rendering. http://media.steampowered.com/apps/valve/2015/Alex_Vlachos_Advanced_VR_Rendering_GDC2015.pdf Game Developers Conference Talk.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. Video-to-Video Synthesis. In *Neural Information Processing Systems*.

Zhou Wang, Alan Conrad Bovik, Ligang Lu, and Jack L Kouloheris. 2001. Foveated wavelet image quality index. *Proc. SPIE* 4472 (2001), 42–53.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

Zhou Wang, Ligang Lu, and Alan C. Bovik. 2003. Foveation scalable video coding with automatic fixation selection. *IEEE Transactions on Image Processing* 12, 2 (2003), 243–254.

Martin Weier, Thorsten Roth, Ernst Kruijff, André Hinkenjann, Arsène Pérard-Gayot, Philipp Slusallek, and Yongmin Li. 2016. Foveated Real-Time Ray Tracing for Head-Mounted Displays. *Computer Graphics Forum* 35 (2016), 289–298.

M. Weier, M. Stengel, T. Roth, P. Didyk, E. Eisemann, M. Eisemann, S. Grogorick, A. Hinkenjann, E. Kruijff, M. Magnor, K. Myszkowski, and P. Slusallek. 2017. Perception-driven Accelerated Rendering. *Computer Graphics Forum* 36, 2 (2017), 611–643.

Y. Ye, E. Alshina, and J. Boyce. 2017. Algorithm descriptions of projection format conversion and video quality metrics in 360Lib. *Joint Video Exploration Team of ITU-T SG* 16 (2017).

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proc. Conf. Computer Vision and Pattern Recognition*.