Kämäräinen, Teemu; Siekkinen, Matti

Foveated Spatial Compression for Remote Rendered Virtual Reality

Published in:
MetaSys 2023 - Proceedings of the 1st Workshop on Metaverse Systems and Applications, Part of MobiSys 2023

DOI:
10.1145/3597063.3597359

Published: 18/06/2023

Document Version
Publisher's PDF, also known as Version of record

Foveated Spatial Compression for Remote Rendered Virtual Reality

Teemu Kämäräinen
University of Helsinki
Finland
teemu.kamarainen@helsinki.fi

Matti Siekkinen
Aalto University and University of Helsinki
Finland
matti.siekkinen@aalto.fi

ABSTRACT

In remote rendered virtual reality (VR), the rendering of the application is moved to the cloud enabling high quality real-time content to be consumed on low-powered standalone head mounted displays (HMDs). The rendered frames are encoded to a video stream and streamed to a thin client which relays user's input to the server and decodes and displays the incoming video. Latency and high bandwidth requirements are key challenges for remote rendered graphics. Foveation can be used to optimize the quality of the transmitted frames to be in line with the human visual system (HVS). In this paper we evaluate multiple different strategies on how to apply foveation to spatially compress video frames, i.e., reduce their resolution, before transmission. We also show how the foveation methods can be used together with super resolution to alleviate the bandwidth usage of real-time remote rendered VR and optimize the perceived image quality.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; • **Computing methodologies** → **Mixed / augmented reality**; **Perception**.

KEYWORDS

foveation, super resolution, virtual reality

ACM Reference Format:
Teemu Kämäräinen and Matti Siekkinen. 2023. Foveated Spatial Compression for Remote Rendered Virtual Reality. In *First Workshop on Metaverse Systems and Applications (MetaSys '23), June 18–22, 2023, Helsinki, Finland.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3597063.3597359

1 INTRODUCTION

Remote rendering can be used to bring high quality content to low-powered standalone head mounted displays (HMDs). Latency from user's action or movement to response is crucial in all VR applications and the problem is highlighted when the rendering is offloaded to a cloud or edge server. Remote rendering also requires a high bandwidth connection between the server and the client as high resolution and high-quality video transmission is needed for an immersive experience in VR applications. The available bandwidth can be optimally utilized by using foveation methods which consider the properties of the human visual system (HVS) when rendering and compressing frames.

The human visual system has spatially non-uniform acuity, sharp central vision with progressively lower acuity in the peripheral vision. The area of the highest visual acuity is only 2° due to cortical magnification which quickly drops with respect to eccentricity. The shape of the magnification curve has been the motivation for rendering and encoding frames with similar quality characteristics; highest quality where the user is looking at and lowest quality in the periphery.

Different rendering and post-rendering projections have been proposed in previous research for cloud-rendered VR. The idea is to project the pixels of the frame into a smaller spatial dimension, i.e., reduce the resolution of the frame, at the server side and to re-project them back to the original resolution after streaming the frames at the client side. This spatial compression is done in such a way that quality reduction due to this compression is optimized for HVS, hence we call it foveated spatial compression. Fisheye and other radial projections have the wanted property of highest sampling density in the middle of the frame which is often where VR systems benefit the most from super sampling. The highest sampling density can however be moved according to user gaze by modifying the sampling function. Foveation is possible, as gaze trackers are becoming more popular in the latest HMDs. Even without gaze tracking, users tend to focus to the center of the frame in VR and this saliency and the characteristics of the lenses could still make it useful to utilize foveated streaming. Another method, named Axis-aligned distorted transfer (AADT), divides the frame into rectangular areas with different sampling rates. In this paper we quantify the benefits of both radial distortions and AADT methods for the perceived quality in remote rendered VR.

Super resolution is another technique related to spatial compression which tries to recover the quality of a low-resolution image back to the original quality and resolution, so the input is already spatially compressed. The increase of computing power for neural network inference on mobile devices has made the use of super resolution neural networks on thin client devices viable. In this paper we study the added benefit of super resolution together with the different foveated spatial compression methods for image quality.

Our contributions are as follows. To the best of our knowledge, we are first to quantify the effects of different foveated spatial compression methods on image quality. We are also first to combine the foveated spatial compression methods with client-side mobile super resolution. Finally, we show a promising unique hybrid application of learned downsampling on the server side to learn an optimal downsampling method to match the light-weight upsampling neural network of the client. The results confirm that foveated spatial

MetaSys '23, June 18–22, 2023, Helsinki, Finland

ACM ISBN 979-8-4007-0213-6/23/06.
https://doi.org/10.1145/3597063.3597359

compression methods effectively focus the quality where the user is looking at and even outperform traditional non-foveated super resolution methods. We show that with bandwidth constraints, best image quality is achieved when combining the foveated spatial compression methods with super resolution.

2 RELATED WORK AND BACKGROUND

In this section, we introduce the related work and background on remote rendered virtual reality, different foveated spatial compression methods in addition to super resolution.

2.1 Remote Rendered Virtual Reality

Virtual Reality (VR) provides an audiovisual experience that immerses the human user into a virtual world that is entirely composed of computer-generated graphics. Current VR HMDs are split into tethered solutions which require a cable or a high-speed radio link between the PC which renders the graphics and more affordable mobile standalone HMDs that perform the graphics processing on device. The drawback of the standalone solutions is their limited computing power. Remote rendering of graphics has been proposed as a solution to bring the quality and immersiveness of mobile stand-alone VR headsets on par with tethered solutions [10, 14, 19]. In remote rendering, the graphics processing is outsourced to a cloud server equipped with a powerful GPU. In this work, we focus for the scenario of remote rendered VR.

The main challenges with remote graphics rendering are latency and network bandwidth. User's interactions need to be perceived on the headset as soon as possible. In addition, the transmission of graphics from the remote rendering server to a mobile device requires a significant amount of available bandwidth from the network, even with lossy video compression. These challenges have motivated the use of foveation techniques to reduce the workload, bandwidth and latency when streaming remote rendered graphics.

2.2 Foveated Rendering, Encoding and Transmission

Foveated graphics is a natural method to reduce the bandwidth requirements of VR. In foveated rendering, the amount of computational work is reduced as the frames are rendered with non-uniform acuity [6, 11]. Foveated rendering can be combined with foveated encoding [4, 8, 21], where already rendered frames are encoded with a lesser quality outside the user's foveal region. Both techniques and their interplay has been researched in previous studies [9, 16, 27].

Different foveated spatial compression methods have also been proposed in previous research as an optimized way of transmitting a frame. Reinert et al. [28] proposed a hemispherical fisheye projection to concentrate the density to the center of the frame with gradual decay toward the periphery. Foveated warping [9] has been proposed as a way to reduce the number of pixels within each frame in a foveated manner. This yields a lower bitrate when encoded to a video. Outside research, a similar pre-processing step before streaming, named Axis-aligned Distorted Transfer (AADT), is used in both a commercial use case for streaming VR [25] and in an open-source system [2]. These methods spatially compress the image in proportion of the distance from the gaze fixation point. In our work, we quantify the effect of the different spatial compression methods for the perceived image quality for the user before combining the methods with super resolution.

2.3 Super Resolution

Super resolution using deep learning has been extensively studied in recent years with increasing attention also to mobile super resolution. The focus of the super resolution research has been to recover a high-resolution image from low resolution images which have usually been obtained using bicubic downsampling. A multitude of different architectures with variable computational requirements have been proposed [5, 13, 18, 19]. For mobile super resolution, the focus has been in finding different ways to optimize the neural network for mobile inference. In this work, we use a light-weight super resolution model XLSR [3] to evaluate the possibility of using light-weight super resolution models together with spatial compression methods to optimize the bandwidth usage in remote rendered VR.

Foveation has not been extensively researched together with super resolution methods and VR. Zhang et al. proposed to use super resolution for volumetric video streaming [30]. Wang et al. [29] proposed a method which uses more neural network blocks in the foveal area than in the periphery to focus the quality where the user is looking at. Lee et al. [15] introduced a novel technique to fuse low-resolution context with regional high-resolution context in video super resolution. In contrast to previous foveated super resolution research, we focus specially for the use case of remote rendered virtual reality and combine mobile super resolution with different foveated spatial compression methods.

3 FOVEATED SPATIAL COMPRESSION

In this section, we present and evaluate different foveated spatial compression methods for optimizing foveal quality in real-time remoted rendered VR.

3.1 Methods

Efficient use of bandwidth is important for all remote rendered applications. In traditional cloud gaming, the view is typically transmitted with a linear perspective projection which is the same projection as the game engine natively renders with. Linear perspective projection has the unwanted property that the center of the image has the lowest sample density while the periphery has the highest. This problem is highlighted with high field-of-view displays such as in VR, which is why local VR setups often render with a higher resolution than the target display to have more samples available when projecting the image to the HMD.

As stand-alone VR headsets are capable of processing graphics, frames can be transmitted with intermediate projections before finally showing the frames to the user. The motivation for these projections is to lower the resolution of the image to reduce the resource usage of the video encoder, decoder, and the required bandwidth in the transport medium. They can be thought as fixed-foveated compression with varying functions for foveation.

3.1.1 Foveated Radial Warp. Forms of radial warp functions which mimic the properties of the HVS have been proposed in previous work [9, 28]. The radial functions can be parameterized in numerous

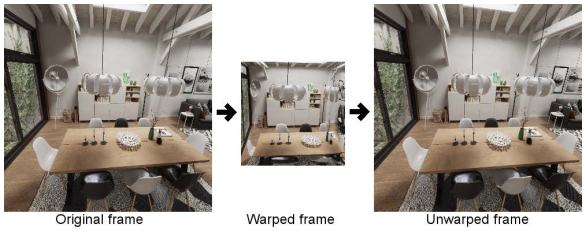

Figure 1: Example of a rendered frame (left) warped with a foveated radial warp (FRW) function (center) and inversely warped back to original shape (right). The gaze fixation point is at the center of the frame.

ways, with the common property being that the sampling rate reduces as the radial distance to the gaze fixation point increases. Some radial functions leave black areas to the corners of the frame. This can be rectified by slightly modifying the radial function. In our tests, we utilize a similar function that is used by Google in Youtube for equiangular cubemaps (EAC) taking advantage of the entire rectangular video frame. Our modified version has a magnitude variable which can be changed for different foveation rates. The equation for the foveated radial warp when gaze is in the center of the frame is:

$$f_{u,v} = 2 * \frac{\tan(2 \arctan(\frac{M}{2})(p_{x,y} - 0.5))}{M} \quad (1)$$

where $f_{u,v}$ are the warped coordinates in the range -1 to 1, M is the warping magnitude, and $p_{x,y}$ are the original coordinates in the range 0 to 1. A warping magnitude of 1 would correspond to regular bicubic sampling. Gaze point can be changed by shifting the warp function and scaling the resulting coordinates accordingly. The inverse foveated radial warp function $f_{u,v}^{-1}$ is used to sample the foveated frame back to the original form. A sample frame warped using the foveated warping function with gaze in the center using $M = 4.66$ is shown in Figure 1.

3.1.2 Axis-aligned Distorted Transfer. AADT is another foveated spatial compression method which is used by Meta in its Quest Link feature which can stream VR from a PC to a standalone headset using a Wi-Fi connection. The details of it have not been fully published and depending on the source [23, 24] it uses either a linear warping function very similar to our FWR function or divides the frame into two or more rectangular areas with each having fixed sampling rates. As the linear version of AADT is almost identical to our FRW function, we compare FRW to the fixed versions of AADT, one with 2 sampling rate areas (AADT2) and one with three (AADT3).

For AADT2, we divide the frame into 2 regions. The part where the user is looking at is sampled with the original full rate, i.e. once per each pixel, while the outer area is sampled with half rate, once per two pixels. For AADT3, we divide the frame into three parts, again using full sampling rate in the part where the user is looking at, half rate in the intermediate region and sampling once per four pixels in the periphery. A sample frame warped using the different versions of the AADT function is shown in Figure 2.

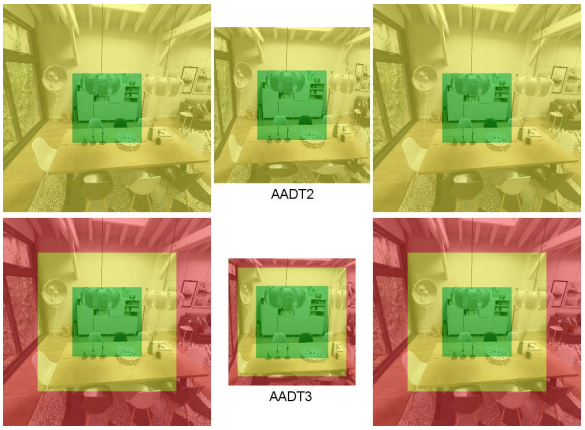

Figure 2: Example of a rendered frame (left) warped with AADT2 (top) and AADT3 (below) with both re-projected back to original shape (right). The gaze fixation point is at the center of the frame. The sampling rates are color coded for clarity: 1 (green), 1/2 (yellow), 1/4 (red).

3.2 Experiment Setup

We evaluate the different foveated spatial compression strategies using non-foveated bicubic interpolation to the same target resolution as the baseline. The target resolutions chosen are 2x and 3x smaller than the original resolution (1440x1440 for single eye) as we want to pair the foveated spatial compression methods with a super resolution network in Section 4.

We use three different metrics to evaluate the different strategies. The difference between peak signal-to-noise ratio (PSNR) and eye-tracking-weighted PSNR (EWPSNR) [17] measures how the quality shifts where the user is looking at with the expense of the periphery. EWPSNR assigns weights to pixels according to a 2D-Gaussian model of human vision. It is calculated as follows.

$$EWPSNR = 10 * \log(\frac{(2^n - 1)^2}{EWMSE}) \quad (2)$$

$$EWMSE = \frac{1}{\sum_{x=1}^{M}\sum_{y=1}^{N} w_{x,y}} \sum_{x=1}^{M}\sum_{y=1}^{N}(w_{x,y} \cdot (I'_{x,y} - I_{x,y})^2) \quad (3)$$

$$w_{x,y} = \frac{1}{2\pi\sigma_x\sigma_y} e^{-(\frac{(x-x_e)^2}{2\sigma_x^2} + \frac{(y-y_e)^2}{2\sigma_y^2}} \quad (4)$$

where I and I' are the original frame and the compared frame, M and N are the frame's height and width in pixels, n is bit depth, and $w_{x,y}$ is the weight for distortion at position (x, y). $w_{x,y}$ is calculated based on the eye fixation (x_e, y_e) and σ_x and σ_y are two parameters related to the distance and view angle, usually taken from fovea size. We use $\sigma_{x,y} = 64$ (roughly 5°) in our experiments.

FovVideoVDP [22] is a more advanced metric which models the spatial, temporal, and peripheral aspects of perception and is specially targeted for wide field-of-view video such as VR. We enabled foveated mode in FovVideoVDP and configured the display resolution to be 1440x1440 (single eye) with a diagonal field of view of 113° to be in line with our dataset still closely matching the resolutions and properties of typical standalone headsets. We

Table 1: Results for the image quality with different foveation methods for the UHDSR4K dataset of images. The best results are underlined for each metric and test case.

	UHDSR4K (images) ↓ Bicubic, ↑ Bicubic			
Scale	Foveation	PSNR	EWPSNR	FovVideoVDP
2x	None	31.79	31.53	9.59
	AADT2	30.99	35.25	9.68
	AADT3	29.43	37.95	9.60
	Warp (M=2.6)	31.20	36.43	9.60
	Warp (M=4.7)	28.13	40.74	9.26
3x	None	28.22	28.09	9.24
	AADT2	27.77	30.40	9.34
	AADT3	26.85	32.38	9.23
	Warp (M=2.6)	26.90	30.59	9.00
	Warp (M=4.7)	25.28	33.65	8.78
	Warp (M=7.9)	22.95	33.59	8.38

Table 2: Results for the image quality with different foveation methods for the Archviz dataset with video compression in between (CBR). The best results are underlined for each metric and test case.

	Archviz (video w/CBR) ↓ Bicubic, ↑ Bicubic			
Scale	Foveation	PSNR	EWPSNR	FovVideoVDP
2x	None	29.45	30.79	9.44
	AADT2	29.02	32.33	9.52
	AADT3	27.00	32.64	9.11
	Warp (M=2.6)	28.62	32.39	9.46
	Warp (M=4.7)	27.46	33.85	9.33
3x	None	27.55	28.81	9.15
	AADT2	27.21	30.07	9.24
	AADT3	26.21	30.86	9.04
	Warp (M=2.6)	26.53	30.12	9.06
	Warp (M=4.7)	25.50	31.08	8.88
	Warp (M=7.9)	24.04	31.59	8.53

utilized the single image mode of FovVideoVDP in our tests and averaged the results over all frames in the video quality tests.

The test set of the UHDSR4K [31] dataset is used to measure the perceived quality for spatially compressed single images representing single frames in a remotely rendered scenario. However, for a more realistic scenario, we also quantify the effects of video compression in a remote rendered scenario and use a continuous trace of a high fidelity architectural visualization scene, named Archviz including 5406 captured frames. An example frame of this scene is depicted in Figure 1. We apply the spatial compression methods for the trace and encode the resulting frames into compressed video using the NVENC [26] H.264 video codec with a constant bit rate (CBR) setting (10M and 5M for 2x and 3x respectively) and decode them prior to applying the inverse spatial compression.

3.3 Results

The measurement results for images and for the compressed video trace with a constant bit rate are shown in Tables 1 and 2. As expected, non-foveated downsampling has best overall PSNR as it samples the frames with a spatially uniform quality. The EWPSNR results show how the quality is shifted where the user is looking at using the foveated spatial compression methods. For 2x scaling, the radial warp function with magnitude 4.7 gives the highest score for both single images and with video compression in between. Scaling with a factor of 3 gives similar results for EWPSNR, although there, a higher magnitude gives the best results for the Archviz dataset.

FovVideoVDP gives the best score for the AADT2 spatial compression scheme for all tests. The metric seems to be configured for a low amount of preferred foveation. Overall, the results show that foveated spatial compression methods can focus the quality where the user is looking at and depending on the use case (and metric) different levels of foveation should be applied. In the next section, we measure how both a light-weight client-side and a learned downsampling-based server-side super resolution strategy can be combined with the foveated spatial compression methods.

4 SUPER RESOLUTION

In this section we combine the foveated spatial compression methods of the previous section with both a light-weight client-side super resolution network (Section 4.1) and also couple the client-side SR network with server-side learned downsampling (Section 4.2).

4.1 Client-side Super Resolution

The goal of this experiment is to evaluate the plausibility of using a light-weight super resolution network which can be run on a mobile device together with the foveated spatial compression methods. We are mostly interested in the relative gains of super resolution compared to the bicubic case without foveation. This shows how much additional perceived quality can be recovered when the input to the SR model is a spatially compressed image instead of a uniformly rendered linear perspective image.

We chose XLSR [3] as the light-weight SR model for our experiments as it has been extensively benchmarked on different mobile platforms [7] showing real-time performance on the latest mobile SoCs. We validate the inference speed of the model in Section 4.3. The network was trained with the DIV2K [1] dataset as in the original work, both for 2x and 3x scales using the same training strategy as in the original paper. We combined the super resolution network with the spatial compression methods introduced before using the same datasets introduced in Section 3.2. An example of the super resolution pipeline is presented in Figure 3. The frames are rendered and spatially compressed on the server. On the client, we however introduce a new step which applies the super resolution model to the foveated frame before re-projecting it back to the original form and displaying it.

The results of the experiment are presented in Tables 3 and 4. The non-foveated version and all the foveation strategies benefit from the SR except for AADT3, which seems to be incompatible with the kernel size of the x2 SR model. The overall PSNR values of the other foveated strategies are 0.4-3.2 higher for images and 0.6-2.0 higher with video compression. Excluding AADT3, the EW-PSNR scores

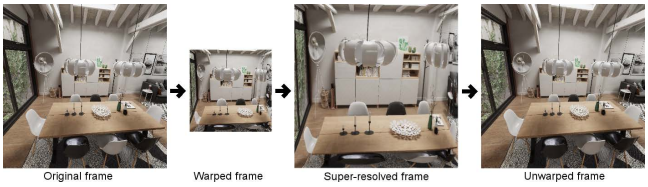

Figure 3: Example of the proposed super resolution pipeline. The original rendered frame (left) is spatially compressed on the server with a warp function before video compression. The client decodes, super-resolves and inversely warps the frame back to original shape (right).

Table 3: Results for the image quality with different foveated spatial compression methods and client-side super resolution (XLSR) for the UHDSR4K dataset of images. The best results are underlined for each metric and test case.

	UHDSR4K (images) ↓ Bicubic, ↑ XLSR			
Scale	Foveation	PSNR	EWPSNR	FovVideoVDP
2x	None	35.62	35.34	9.79
	AADT2	33.81	40.25	9.83
	AADT3	27.28	35.76	9.04
	Warp (M=2.6)	34.12	39.90	9.87
	Warp (M=4.7)	30.90	42.37	9.72
3x	None	30.77	30.62	9.48
	AADT2	29.99	33.78	9.57
	AADT3	27.28	35.05	9.15
	Warp (M=2.6)	30.14	33.74	9.63
	Warp (M=4.7)	28.33	36.18	9.46
	Warp (M=7.9)	26.02	36.78	8.98

are 1.6-5.0 higher for images and 1.3-2.2 higher for video, while the FovVideoVDP scores are 0.1-3.5 and 0.1-0.5 higher for images and video, respectively. Overall, the results show that super resolution is a viable method to increase the overall quality in remote rendered VR, also coupled with different foveation strategies.

We also did an experiment where we fine-tuned the trained super resolution network with foveated spatially compressed images matching the warping magnitude of the test case. This did not lead to image quality improvements compared to the model without fine-tuning. The result shows that regular super resolution networks generalize also for different foveated use cases which could enable the use of dynamic foveation magnitudes with a single network.

4.2 Server-assisted Mobile Super Resolution

Information is lost in the process of image downscaling both in bicubic downsampling and the foveated spatial compression methods. In super resolution, task-aware image downscaling (TAD), which uses auto-encoder based architecture, has been proposed [12] to jointly learn the downscaling and the upscaling of the image to maximize the restoration performance. Their performance has been shown to improve over traditional super resolution networks. In

Table 4: Results for the image quality with different foveated spatial compression methods and client-side super resolution (XLSR) for the Archviz dataset with video compression in between (CBR). The best results are underlined for each metric and test case.

	Archviz (video w/CBR) ↓ Bicubic, ↑ XLSR			
Scale	Foveation	PSNR	EWPSNR	FovVideoVDP
2x	None	30.65	32.15	9.52
	AADT2	30.03	33.70	9.58
	AADT3	27.11	33.47	9.12
	Warp (M=2.6)	30.21	33.99	9.62
	Warp (M=4.7)	28.96	35.64	9.60
3x	None	28.73	30.18	9.27
	AADT2	28.27	31.50	9.35
	AADT3	26.79	32.14	9.12
	Warp (M=2.6)	28.46	31.73	9.40
	Warp (M=4.7)	27.47	33.02	9.36
	Warp (M=7.9)	25.95	33.75	9.03

remote-rendered VR, the cloud server is often equipped with a GPU capable of neural network inference in addition to rendering.

In this work, we evaluate if a hybrid super resolution network could be combined using the encoder part of the model introduced by Kim et al. [12] and the light-weight super resolution model XLSR [3] introduced before as the decoder. The architecture of the hybrid autoencoder model is shown in Figure 4. We trained the network with the commonly used DIV2K dataset [1] using Charbonnier loss [20] with $\epsilon = 0.1$. We used a guidance loss for the downscaled version of the image which was compared to an image produced with bicubic downsampling. The guidance loss was equally weighted with the loss for the super-resolved image. We trained the network for 800 epochs with a triangular cyclic learning rate scheduling with a maximum learning rate of 0.001.

We evaluate the encoded images with quantization to 8 bits so that the images are useful in typical applications using super resolution. The results of the experiment for the UHDSR4K dataset of images are presented in Table 5 where TAD is task-aware downscaling. The results show that task-aware downscaling is a promising approach also for remote rendering. Overall, the results are the best in all metrics. Compared to the XLSR upsampling with bicubic downsampling case of the last section, the overall PSNR values are 1.3-4.1 higher depending on the use case. The foveated scores are also 0.1-5.4 and 0.1-0.6 higher for EWPSNR and FovVideoVDP respectively. The rank order remains the same, with lower warping magnitudes giving best results for FovVideoVDP and higher magnitudes for EWPSNR.

While the image-based results are promising, when we applied video compression to the TAD encoder output and then decompressed it prior to feeding it to the XLSR decoder, the results were slightly worse compared to bicubic downsampling combined with the XLSR super resolution. We leave the adjustment of the auto-encoder network with video compression for future work.

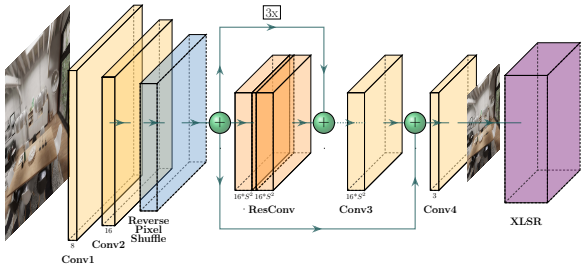

Figure 4: The auto-encoder network for non-symmetrical server-assisted mobile super resolution. XLSR [3] network is used for upscaling. S is the upscaling ratio.

Table 5: Results for image quality with different foveated spatial compression methods for the UHDSR4K dataset of images using an auto-encoder based super resolution network (TAD) for learned downsampling and XLSR for upsampling.

		UHDSR4K (images) ↓ TAD, ↑ XLSR		
Scale	Foveation	PSNR	EWPSNR	FovVideoVDP
2x	None	37.85	40.04	9.91
	AADT2	35.08	40.30	9.90
	AADT3	29.43	37.95	9.60
	Warp (M=2.6)	35.97	41.69	9.92
	Warp (M=4.7)	32.55	42.81	9.80
3x	None	34.90	35.02	9.78
	AADT2	33.19	38.78	9.83
	AADT3	30.16	36.96	9.53
	Warp (M=2.6)	33.59	39.11	9.87
	Warp (M=4.7)	30.69	40.86	9.74
	Warp (M=7.9)	27.53	40.78	9.33

4.3 Inference Time

The spatial compression methods evaluated before can all be implemented with a computationally light-weight shader which is trivial even for a mobile GPU to compute. The SR methods are however computationally heavy, and their inference time needs to be considered. We benchmarked the XLSR network with high resolution inputs with three mobile SoCs: Qualcomm Snapdragon XR2, Snapdragon 888 and Snapdragon 8 Gen 2. For the first, a standalone headset HTC Vive Focus 3 was used, for the second and third Samsung Galaxy Z Fold3 5G and Samsung Galaxy S23 Ultra mobile phones were used. We ran quantized versions of the models so we could leverage the neural network accelerators on the SoCs.

The results of the inference tests are shown in Table 6. Qualcomm Snapdragon XR2 SoC, which is used in popular standalone HMDs like the Meta Quest 2 and HTC Vive Focus 3, is by an order of magnitude slower than the newer SoCs on which we were able to utilize the Hexagon Tensor Processor. On the most recent SoC, the Snapdragon 8 Gen 2, the XLSR network could be run in 13.9 ms and 6.1 ms for 2x and 3x scales. This translates to 72 and 164 frames per second, which is promising for running real-time neural networks for SR in the next generation of standalone HMDs.

Table 6: Inference times of the XLSR SR model on mobile SoCs for 2x and 3x upscaling ratios.

Scale	Input resolution	Inference time		
		Snapdragon XR2	Snapdragon 888	Snapdragon 8 Gen 2
2x	720x1440	468.0 ms	22.1 ms	13.9 ms
3x	480x960	197.0 ms	10.7 ms	6.1 ms

5 DISCUSSION

Lowering the resolution of the transmitted frames in remote rendered VR is an effective way of lowering its bandwidth requirements. Since in practical scenarios the frames will additionally be compressed using a video codec, the spatial compression has also the added benefit of reducing the latency of the video encoder and decoder as their induced latency is proportional to the number of bits that need to be coded [9]. In our experiments, we showed that foveated spatial compression methods can effectively retain the quality in parts of the frame where the user is looking at with the expense of the periphery. This can be quantified using foveated metrics such as EWPSNR and FovVideoVDP. Spatial compression can be however also useful even without gaze tracking, as the lens distortion of current VR HMDs distort the image in the periphery. This is however, to best of our knowledge, not modelled in the foveated metrics. In future work, the results obtained here should be validated with user testing using real HMDs.

The foveated spatial compression methods together with the light-weight super resolution network could be taken into use with current hardware (with the right SoC) in remote rendering systems. The server-assisted SR version however needs additional work to also be useful in real-world scenarios with video compression present. For future work, it would be beneficial to model the effects of real-world video encoders to bring them to be part of the super resolution training pipeline. Recent advances in completely neural network-based encoders and decoders could also be studied to be combined with foveation methods.

6 CONCLUSION

Foveated spatial compression methods can be combined with gaze tracking to reduce the resolution of the transmitted frames without sacrificing the perceived image quality in remote rendered VR. This reduces used bandwidth and lowers the overall latency as fewer bits need to be encoded, transmitted and decoded. In this paper, we quantify the effects of foveated spatial compression methods and show how they can be combined with a light-weight client-side super resolution network with and without video compression. We show that with the latest mobile SoCs, the light-weight super resolution network can be run in real-time even with high resolutions required by remote rendered VR.

ACKNOWLEDGMENTS

This work has been supported by the Academy of Finland (grant number 332306).

REFERENCES

[1] Eirikur Agustsson and Radu Timofte. 2017. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 126–135.

[2] ALVR. 2023. PC VR Without the Wires. Retrieved February 27, 2023 from https://alvr-org.github.io/

[3] Mustafa Ayazoglu. 2021. Extremely lightweight quantization robust real-time single-image super resolution for mobile devices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2472–2479.

[4] Zhenzhong Chen and Christine Guillemot. 2010. Perceptually-friendly H. 264/AVC video coding based on foveated just-noticeable-distortion model. *IEEE Transactions on Circuits and Systems for Video Technology* 20, 6 (2010), 806–819.

[5] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*. Springer, 184–199.

[6] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder. 2012. Foveated 3D graphics. *ACM transactions on Graphics (tOG)* 31, 6 (2012), 1–10.

[7] Andrey Ignatov. 2023. AI-Benchmark. Retrieved Marh 1, 2023 from https://ai-benchmark.com/ranking_detailed.html

[8] Gazi Karam Illahi, Thomas Van Gemert, Matti Siekkinen, Enrico Masala, Antti Oulasvirta, and Antti Ylä-Jääski. 2020. Cloud gaming with foveated video encoding. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 16, 1 (2020), 1–24.

[9] Gazi Karam Illahi, Matti Siekkinen, Teemu Kämäräinen, and Antti Ylä-Jääski. 2021. Foveated streaming of real-time graphics. In *Proceedings of the 12th ACM Multimedia Systems Conference*. 214–226.

[10] Teemu Kämäräinen, Matti Siekkinen, Jukka Eerikäinen, and Antti Ylä-Jääski. 2018. CloudVR: Cloud accelerated interactive mobile virtual reality. In *Proceedings of the 26th ACM international conference on Multimedia*. 1181–1189.

[11] 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 Transactions on Graphics (TOG)* 38, 6 (2019), 1–13.

[12] Heewon Kim, Myungsub Choi, Bee Lim, and Kyoung Mu Lee. 2018. Task-aware image downscaling. In *Proceedings of the European conference on computer vision (ECCV)*. 399–414.

[13] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1646–1654.

[14] Zeqi Lai, Y Charlie Hu, Yong Cui, Linhui Sun, and Ningwei Dai. 2017. Furion: Engineering high-quality immersive virtual reality on today's mobile devices. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 409–421.

[15] Eugene Lee, Lien-Feng Hsu, Evan Chen, and Chen-Yi Lee. 2023. Cross-Resolution Flow Propagation for Foveated Video Super-Resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1766–1775.

[16] Sanghoon Lee, Marios S Pattichis, and Alan C Bovik. 2001. Foveated video compression with optimal rate control. *IEEE Transactions on Image Processing* 10, 7 (2001), 977–992.

[17] Zhicheng Li, Shiyin Qin, and Laurent Itti. 2011. Visual attention guided bit allocation in video compression. *Image and Vision Computing* 29, 1 (2011), 1–14.

[18] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 136–144.

[19] Luyang Liu, Ruiguang Zhong, Wuyang Zhang, Yunxin Liu, Jiansong Zhang, Lintao Zhang, and Marco Gruteser. 2018. Cutting the cord: Designing a high-quality untethered vr system with low latency remote rendering. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. 68–80.

[20] Alice Lucas, Santiago Lopez-Tapia, Rafael Molina, and Aggelos K Katsaggelos. 2019. Generative adversarial networks and perceptual losses for video super-resolution. *IEEE Transactions on Image Processing* 28, 7 (2019), 3312–3327.

[21] Pietro Lungaro, Rickard Sjöberg, Alfredo Jose Fanghella Valero, Ashutosh Mittal, and Konrad Tollmar. 2018. Gaze-aware streaming solutions for the next generation of mobile VR experiences. *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1535–1544.

[22] Rafał K Mantiuk, Gyorgy Denes, Alexandre Chapiro, Anton Kaplanyan, Gizem Rufo, Romain Bachy, Trisha Lian, and Anjul Patney. 2021. Fovvideovdp: A visible difference predictor for wide field-of-view video. *ACM Transactions on Graphics (TOG)* 40, 4 (2021), 1–19.

[23] Meta. 2019. How does Oculus Link Work? The Architecture, Pipeline and AADT Explained. Retrieved March 9, 2023 from https://developer.oculus.com/blog/how-does-oculus-link-work-the-architecture-pipeline-and-aadt-explained/

[24] Meta. 2019. Oculus Link for Quest. Retrieved March 9, 2023 from https://youtu.be/9gocUADwqo8?t=687

[25] Meta. 2023. How does Oculus Link Work? The Architecture, Pipeline and AADT Explained. Retrieved February 27, 2023 from https://developer.oculus.com/blog/how-does-oculus-link-work-the-architecture-pipeline-and-aadt-explained

[26] Nvidia. 2022. NVIDIA VIDEO CODEC SDK - ENCODER Programming Guide. Retrieved November 25, 2022 from https://docs.nvidia.com/video-technologies/video-codec-sdk/pdf/NVENC_VideoEncoder_API_ProgGuide.pdf

[27] 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 Transactions on Graphics (TOG)* 35, 6 (2016), 1–12.

[28] Bernhard Reinert, Johannes Kopf, Tobias Ritschel, Eduardo Cuervo, David Chu, and Hans-Peter Seidel. 2016. Proxy-guided image-based rendering for mobile devices. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 353–362.

[29] Lingdong Wang, Mohammad Hajiesmaili, and Ramesh K Sitaraman. 2021. Focas: Practical video super resolution using foveated rendering. In *Proceedings of the 29th ACM International Conference on Multimedia*. 5454–5462.

[30] Anlan Zhang, Chendong Wang, Bo Han, and Feng Qian. 2022. {YuZu}:{Neural-Enhanced} Volumetric Video Streaming. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*. 137–154.

[31] Kaihao Zhang, Dongxu Li, Wenhan Luo, Wenqi Ren, Björn Stenger, Wei Liu, Hongdong Li, and Ming-Hsuan Yang. 2021. Benchmarking Ultra-High-Definition Image Super-Resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 14769–14778.

Received 14 April 2023