Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

Neural Foveated Super-Resolution for Real-time VR Rendering

Jiannan Ye[1] | Xiaoxu Meng[2] | Daiyun Guo[1] | Cheng Shang[1] | Haotian Mao[1] | Xubo Yang*[1]

[1]Digital ART Lab, School of Software,, Shanghai Jiao Tong University, Shanghai, China

[2]Tencent Games Digital Content Technology Center, Tencent, CA, U.S.A

Correspondence
Corresponding author Xubo Yang, School of Electronic Information and Electrical Engineering Shanghai Jiao Tong University, Shanghai, China
Email: yangxubo@sjtu.edu.cn

Abstract

As virtual reality display technologies advance, resolutions and refresh rates continue to approach human perceptual limits, presenting a challenge for real-time rendering algorithms. Neural super-resolution is promising in reducing the computation cost and boosting the visual experience by scaling up low-resolution renderings. However, the added workload of running neural networks cannot be neglected. In this paper, we try to alleviate the burden by exploiting the foveated nature of the human visual system, where acuity decreases rapidly from the focal point to the periphery. With the help of dynamic and geometric information (i.e.,pixel-wise motion vectors, depth, and camera transformation) available inherently in the real-time rendering content, we propose a neural accumulator to effectively aggregate the amortizedly rendered low-resolution visual information from frame to frame recurrently. By leveraging a partition-assemble scheme, we use a neural super-resolution module to upsample the low-resolution image tiles to different qualities according to their perceptual importance and reconstruct the final output heterogeneously. Perceptually high-fidelity foveated high-resolution frames are generated in real-time, surpassing the quality of other foveated super-resolution methods.

KEYWORDS
foveated rendering, super-resolution, virtual reality

1 | INTRODUCTION

With the emergence of the metaverse, the demand for realistic and enjoyable immersive experiences is rising continuously. The demand drives not only the development of rendering algorithms but also virtual reality display techniques. Virtual reality headsets nowadays can achieve 8K resolution and refresh rates up to 144Hz to enable the rendering of physical reality and fancy visual effects. These together lead to a much larger computational workload than ever before, which is hard for computing devices to catch up.

With the development of artificial intelligence, there are more and more learning-based neural methods applied to the rendering field to tackle the computational burden, such as neural denoising[1,2], neural interpolation and extrapolation[3,4] and neural supersampling methods[5–7]. Supersampling, similar to super-resolution[8,9], is promising to help real-time rendering reduce the computation cost and boost the visual experience by scaling up low-resolution renderings. Our method also falls into this group. Nvidia's deep learning super sampling (DLSS)[6], and its counterparts XeSS[10], FSR[11], TAAU[12] are now widely used in video games to get a native image quality with a higher frame rate. However, they are either hardware-specific or with a limited upscale ratio (e.g., 2×2, 1080P to 4K, which the original resolution is relatively large enough). Neural supersampling for real-time rendering (NSRR) [5] is the recent research work targeting 4×4 supersampling with a fixed sliding window of 5 historical frames of extremely low-resolution rendering input. Our method also targets this difficult super-resolution task of a large upscale ratio of 4×4 to lower the cost of native rendering. It contributes to a better super-resolution quality with a well-designed temporal information accumulation mechanism with the help of neural networks that can progressively enrich information needed for super-resolution and is not limited by a sized time window.

It is worth noting that running a super-resolution neural network brings another computation workload that the overhead can not be neglected. In the context of virtual reality and wide field-of-view near-eye display, there exist tricks that display and rendering algorithms can take advantage of. For human eyes, the visual acuity rapidly drops from the eye fixation region (fovea) to the eye's periphery. Foveated Rendering algorithms [13–16] exploit this phenomenon by decreasing the rendering quality in the periphery while maintaining high fidelity in the fovea. We exploit a similar idea in our super-resolution setup by treating different regions across the view with varying schemes of super-resolution to achieve various qualities (Figure 1). This optimization reduces the computation cost of the neural networks. However, designing and training a neural network to automatically assign its workload and reconstruct an image with heterogeneous visual quality is not trivial.

Here in this paper, we design a neural network structure that can be trained concurrently with shared weights and of which the reconstruction capability is proportional to the complexity natively. Moreover, we propose an explicit apportion scheme to invoke neural networks to reduce the computation and generate foveated super-resolution results which are well-suited for arbitrary setups of foveation regions.

We summarize our contributions as follows:

- We propose a neural foveated super-resolution pipeline, which sets a new state-of-the-art baseline of foveated super-resolution.
- We introduce a recurrent scheme tailored for the foveated super-resolution pipeline, which uses a novel neural masking and weighting strategy to accumulate historical frames efficiently.
- We design a partition-assemble strategy that leverages arbitrary foveation patterns to generate heterogeneous foveated super-resolution in real-time.

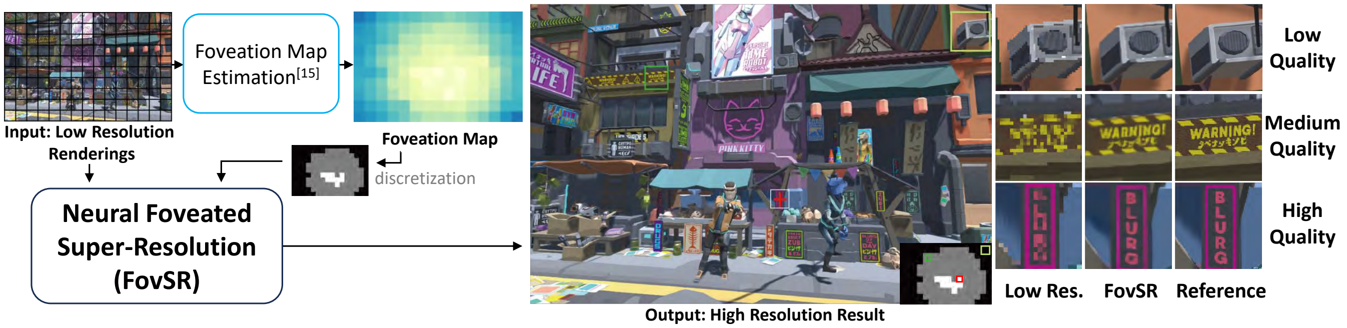

Input: Low Resolution Renderings
Foveation Map Estimation[15]
Foveation Map
discretization
Neural Foveated Super-Resolution (FovSR)
Output: High Resolution Result
Low Quality
Medium Quality
High Quality
Low Res.
FovSR
Reference

FIGURE 1 Overview of our foveated super-resolution method. We begin by generating low-resolution renderings using the native renderer. To determine the spatial visual quality required for each image patch, we estimate a luminance-contrast-aware foveation map [15], considering factors such as eccentricity and content. After discretization, the foveation map categorizes image patches into different quality levels: low, medium, and high. Our neural foveated super-resolution method, referred to as FovSR, takes both the low-resolution rendering content and the discretized foveation map as input. FovSR uses this information to generate foveated high-resolution results that exhibit varying visual quality across distinct regions. For a more detailed examination of our results, zoomed-in crops are displayed on the right, offering a closer look at the output. The fovea is marked by a red cross.

2 | RELATED WORK

2.1 | Image and Video Super Resolution

The convolutional neural network is first used in single image super-resolution (SISR) in SRCNN [17]. Afterwards, deeper structure [18], residual learning [19], dense inter-layer connections [20] are also introduced. EDSR [21] utilizes a stack of modified

ResBlocks[22] to achieve enhanced quality, making it a fundamental building block in super-resolution networks with many subsequent modifications[23,24]. And the stacking manner of blocks is a common practice in designing efficient super-resolution network[24] and is also adopted by our method.

The pioneering video super-resolution (VSR) works either rely on neural networks to implicitly integrate multi-frame information[25–27] or estimate optical flow from adjacent frames to track the pixel movement[28–33]. However, the per-pixel estimation of optical flow is time-consuming which requires an addition neural network, and the accumulated errors can lead to apparent artifacts. Multi-level mechanisms[28,32] and deformable convolution[30,31,33] are employed for more accurate optical-flow estimation, error elimination, and further modification of the sampling positions and weight. Moreover, complicated neural network structures, including the currently emerging transformer-based ones[34], are not favored for real-time performance.

SISR fails to utilize the temporal information, and VSR is getting increasingly sophisticated to integrate it better. In the context of real-time rendering, an accurate motion vector is available in the pipeline, which relieves the effort in flow estimation. NSRR[5] leverages the power of motion vectors to restore high-resolution details with simple neural networks. It directly uses rendering samples from five historical frames and stores several neural tensors requiring relatively large memory. Our method, instead, employs a recurrent scheme to break the limit of fixed window size without incurring any overhead. Our method is more efficient in filtering invalid information and enhancing the rendering quality.

2.2 | Spatial and Temporal Supersampling

Classical temporal-spatial supersampling techniques have been developed based on the observation of shading coherence and consistency across both spatial and temporal dimensions. Supersampling is initially developed to tackle the aliasing problem by generating more samples to produce better pixels while it also refers to the idea of upscale low-resolution renderings[5,6,10–12,35,36].

Classical supersampling methods involves spatial and/or temporal information enrichment. Spatially, there are multisampling antialiasing (MSAA) [37], fast approximate antialiasing (FXAA)[38], morphological antialiasing (MLAA)[39], and subpixel morphological antialiasing (SMAA)[40]. Similar to the idea of VSR, reusing the samples from previous frames inspires a series of temporal supersampling methods[41].

Traditional temporal techniques rely on a temporal accumulator as a crucial component to generate, warp, validate historical rendering samples, and accumulate current samples for every frame. The motion vector is used in a reverse reprojection caching scheme [42] for pixel shaders to look up in the screen-sized cache trying to reuse the available result before doing the standard computing in order to reduce shader execution. Amortized supersampling [43] is proposed to reduce aliasing, which reuses multiple sets of historical shading results estimated at four subpixel positions around the target pixel to retain more spatial information and incrementally update one of them with jittered sampling. Temporal antialiasing (TAA)[44] follows the basic idea of amortized supersampling and uses a low discrepancy progressive sampling sequence to avoid clustering in either space or time. A significant challenge arises because warped historical buffers do not align perfectly with the current frame, leading to ghosting artifacts and visually invalid samples due to movements and varying lighting conditions. Furthermore, successive temporal reprojection introduces blur due to sampling interpolation[43], and the information stored in the historical buffers gradually becomes outdated. TAA uses a neighborhood color clamping/clipping and an exponential moving average scheme to filter invalid samples and blend current samples but requires well-designed heuristics to run effectively[41]. A classical spatio-temporal upsampling method[36] is close to our method's neural temporal accumulator. But it needs high-resolution geometry auxiliary buffers to filter the invalid temporal samples. NSRR[5] is a state-of-the-art method which utilizes neural networks in accumulating temporal samples but only five previous frames can be considered.

In our paper, we use the idea of temporal supersampling to accumulate useful information. We jitter the camera following 16-frame sequence and update 1/16 of the 4×4 grid every frame to amortize the rendering for super-resolution. With the help of neural networks, our method is not limited in certain rendering pipeline or scenarios and is more robust in filtering the temporal information to increase the visual quality. During the reconstruction of high-resolution results, analyzing spatial proximity is also adopted in the reconstruction neural network, which is more powerful compared to traditional spatial supersampling methods.

The idea of temporal supersampling is also applied in coarse pixel shading[45] for antialiasing, in neural denoising[46] for more supporting samples, in interpolation[3] and extrapolation[4] for generating entire novel frames. Neural network is applied more and more often in rendering pipeline and our work of super-resolution relies on them to solve an important aspect of the real-time rendering.

2.3 | Foveated Rendering

Foveated rendering accelerates the rendering in virtual reality by rendering with a non-uniform resolution for the display. Classic foveated rendering methods include multi-layer rendering[13], G-buffer mapping[47,48], variable-rate shading[14,49,50]. Classical method [13] adopts a way to dividing the the field-of-view into 3 rectangular layers according to the eccentricity and assign a fixed shading rate to different layers. The shading rate can also vary according to the image content (such as luminance-contrast[15]) besides eccentricity. The regions with a fixed shading rate can be arbitrary (Figure 1). DeepFovea[51] leverages the power of generative adversarial neural networks to reconstruct a plausible peripheral video from a small fraction of pixels sampled in a foveated pattern, but it fails to adjust the computation workload while the neural network executes the same computation regardless of the region's eccentricity. FovNerf[52] accelerates the rendering of neural radiance field in VR with the idea of foveated rendering. It has to train different networks separately to deal with different foveation layers. FOCAS[53] and FovMSLapSRN (FovMLS)[54] also target on foveated super-resolution. They generate image regions with different qualities, either with the partial model or recursive neural network, respectively. However, these two methods can only handle different foveation layers in a fixed pattern and they are not specifically designed for graphical rendering. In order to reduce the infertime of super-resolution network, we design a partition-assemble scheme to generate foveated super-resolution renderings with arbitrary estimated foveation patterns in real time.

3 | METHOD

In this section, we will demonstrate how our method works. We start by describing the neural temporal accumulator, which maintains and updates a neural historical feature recurrently in Section 3.1. In Section 3.2, we show how our reconstruction neural network translates the historical neural features into foveated super-resolution results.

3.1 | Neural Temporal Accumulator

Temporal-spatial supersampling techniques rely on a temporal accumulator as a crucial component to warp, validate historical rendering samples, and accumulate current samples for every frame. To effectively address these issues such as ghosting artifacts, visually invalid samples and reprojection blur as mentioned in Section 2.2, we propose a novel method that leverages a combination of several neural networks within the temporal accumulator. This enhanced framework, referred to as the Neural Temporal Accumulator (NTA), aims to accumulation historical information as well as mask or attenuate the influence of invalid and outdated samples. The framework of NTA is showed in Figure 2(a).

In our method, the original information of the current frame includes the low-resolution motion vector v_t, depth map d_t, and color buffer r_t with a resolution of (h, w). These are generated by the native graphical renderer and incorporate quarter sub-pixel camera jitter offsets within a fixed 16-frame circular jitter sequence. Concurrently, a corresponding jitter mask J_t, with a resolution of $(4h, 4w)$, is generated, consisting of all 1 value except for the current sub-pixel sampling positions, which are filled with zeros. Furthermore, a neural feature f_t is extracted from the low-resolution rendered color and depth buffer using a feature extraction module (FEM) illustrated in Figure 3 (a) and then upscaled to F_t (4×4 bilinearly upsampling of the low-resolution f_t).

To incorporate information from previous frames into the current frame, the historical neural feature F^R_{t-1} is warped into ${}^{W}F^R_{t-1}$ with the guidance of the upsampled motion vector V_t. This process should also involve effectively validating the historical information. Classical temporal supersampling methods rely on rejection and rectification techniques for this purpose. Rejection methods typically use geometric data, such as reprojected depth, object ID, normal, and world position, to determine the validity[4,11,42]. However, these checks may not be universally applicable and often require specific thresholds, making them fragile, particularly in super-resolution tasks where imprecise upsampled low-resolution buffers are used. Rectification methods, which involve color data, are also employed to handle shading changes, such as moving shadows. In these methods, historical color is clipped to the convex hull in color space based on neighboring samples from the current frame[44]. However, in super-resolution, the neighborhood of current rendered samples may not be reliable as the color gamut is often overestimated. Additionally, the attenuation of historical information needs careful consideration to strike a balance between gradually blurred history and aliased current samples[43]. Neural networks have the potential to replace these complex heuristics, and in our approach, a neural network plays a key role in this temporal filtering, especially for our super-resolution task.

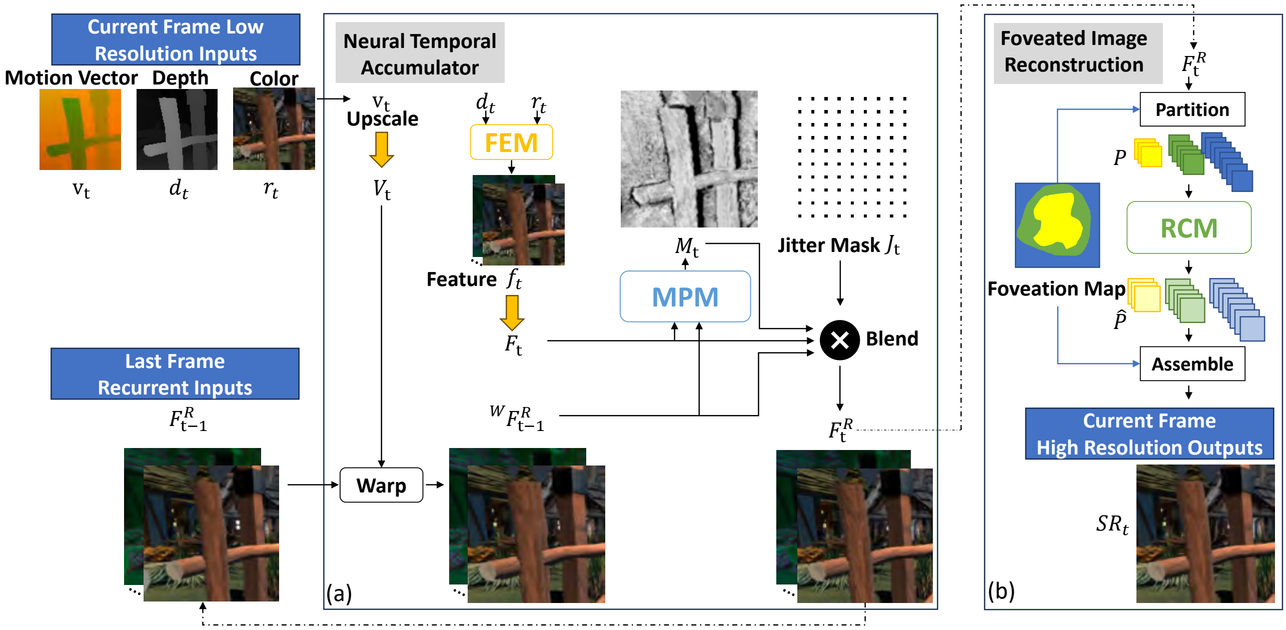


FIGURE 2 Overview of the proposed framework: (a) The process of the neural temporal accumulator, as described in Section 3.1. In each frame t, a graphical renderer generates low-resolution buffers v_t, d_t, r_t with camera jittering. A neural feature f_t is extracted from the color r_t and depth d_t buffer using Feature Extraction Module (FEM). Historical feature F^R_{t-1} is warped from the previous frame with upscaled current motion vector V_t. With warped historical feature $^WF^R_{t-1}$ and upscaled current feature F_t as inputs, a mask M_t is predicted using the mask prediction module (MPM) to blend the warped feature and current feature. In the blending step, a jitter mask J_t is also involved to indicate the shifted positions of current rendering according to the jitter sequence of the camera. Yellow arrows represent bilinearly upscaling and black arrows represent data paths. (b) An overview of the foveated image reconstruction, as described in Section 3.2. The updated neural feature F^R_t is partitioned based on a foveation map, and the reconstruction module (RCM) processes the resulting patches P to obtain corresponding super-resolved patches $\hat{P}$. The final result SR_t for display is generated by assembling these reconstructed patches, which is then passed to the screen for display.

We employ the mask prediction module (MPM) as shown in Figure 2(a) and Figure 3(b) to predict the weight of the historical frame in temporal accumulation. The MPM estimates an adaptive damping weight mask M_t, allowing for the masking out of invalid samples and the attenuation of the outdated samples. This is achieved by leveraging the combined information from the geometry and visual samples stored in the historical feature, as well as the current upsampled neural feature extracted from color and depth. Specifically, MPM takes warped historical feature ($^WF^R_{t-1}$), the current upsampled feature F_t as the inputs, and predicts a one-channel mask M_t used for temporal accumulation.

Then, we blend the warped historical feature $^WF^R_{t-1}$ and the current frame feature F_t to get the accumulated feature F^R_t employing a recursive exponential moving average (EMA) method. Equation 1 is utilized to filter the historical neural feature and integrate the current neural feature. In this equation, the symbol $*$ denotes element-wise multiplication. The introduced jitter mask J_t plays a role in refreshing the neural feature for the currently rendered positions.

$$F^R_t = M_t * J_t *^W F^R_{t-1} + (\mathbf{1} - M_t * J_t) * F_t \tag{1}$$

In our paper, with the neural temporal accumulator proposed, we get the valid accumulated feature F^R_t, which is later used to reconstruct the foveated super-resolution result.

3.2 | Foveated Image Reconstruction

In order to generate foveated super-resolution image which has heterogeneous gaze-contingent quality across the visual field, we build a super-resolution neural ReConstruction Module (RCM) as shown in Figure 2 (b). It allows the users to customize the quality of result by defining the output as

$$\hat{P} = R(P, \Phi_R, q) \tag{2}$$

where P represents a tile of the neural feature after temporal accumulation and blending, Φ_R denotes the network parameters, and q is a flag indicating the quality level.

For super-resolution with varying quality in different regions based on their perceptual importance, we adopt a partition-and-assemble scheme. The partition operation, denoted as $P(F_t^R, k, o, q)$, divides the neural feature F_t^R into small feature patches. After reconstruction, the assembling operation, denoted as $A(\hat{P}, k, o, q)$, reassembles the patches $\hat{P}$ from RCM back to the entire image SR_t. Here, k represents the patch size, o denotes the overlapping size between two patches, and q is the quality flag assigned to each patch which is indicated by a foveation map. We find that patch size k of 32×32 at original low-resolution and an overlap o of 2 are suitable for keeping the computation overhead bearable and meeting the requirement of representing arbitrary foveation patterns. Additional padding and depadding are applied to ensure that all patches have the same size, and the final image is of the correct size.

In detail, during the partitioning process, the arbitrary foveation map assigning patches with a quality flag q can be estimated and generated using any foveated visual theory[13,15]. The patches are then divided into groups according to the quality flag, and each group follows a different path within the reconstruction module to generate $\hat{P}$. The group of patches assigned a low-quality flag exits the module at the early exit, while only the patches assigned the highest-quality flag go through the entire network as shown in Figure 3(c). This approach reduces computation time compared to full-image super-resolution. As for the foveation map, we adopt the luminance-contrast-aware foveation scheme[15] as the default method to estimation foveation map. It uses low-resolution rendering as the input to predict a resolution map for each patch expressed as the acceptable level of blur. After thresholding and discretization, a foveation map of three quality levels is generated as shown in Figure 1.

The structure of the network shown in Figure 3(c), which is built upon the stacked blocks inspired by EDSR[21]. In order to reduce the weights of the neural network and accelerate the inference, we employ a simplified ResBlock, namely Shallow Res Block (SRB) with only one convolution layer[23]. Moreover, to accommodate channel magnification, reduction, and concatenation between blocks, we add an adaptive layer before the original SRB only when it is necessary, as shown in the inset of Figure 3(c) named as Adaptive Shallow Res Block (ASRB), enabling it to adapt to various skip connections and channel changes. In order to build a lighter network, in the early blocks, the channels are further reduced. PixelUnshuffle and PixelShuffle operators are placed at the start and end of this module. PixelUnshuffle transforms the original feature patches into smaller one but with more channels which accelerates the reconstruction while PixelShuffle transforms the results back to the original size.

We also draw inspiration from Unet[5] and Unet++[55]. Skip connections are inserted to establish more connections within the network to better fuse different level of features. Multiple exits for different quality levels of output are provided by adding three distinct reconstruction tails to translate the features into color images. The weights incrementally increases for each quality level while the weights of the common parts are shared. We iteratively output the three results during training and update the corresponding weights.

4 | RESULT

In this section, we evaluate the performance of our method. Necessary implementation details such as datasets, training, loss function, etc. will be first described in Section 4.1. We report full super-resolution quality in Section 4.2, its foveated super-resolution quality in Section 4.3 and runtime in Section 4.4. We demonstrate the user study results in Section 4.5 and analyze our methods with ablation study in Section 4.6.

4.1 | Implementation

We constructed three datasets using the game engine Unity, utilizing scenes obtained from its asset store. Each scene was subjected to exploration by multiple users controlling the camera, resulting in the capture of 100 sequences. Each sequence

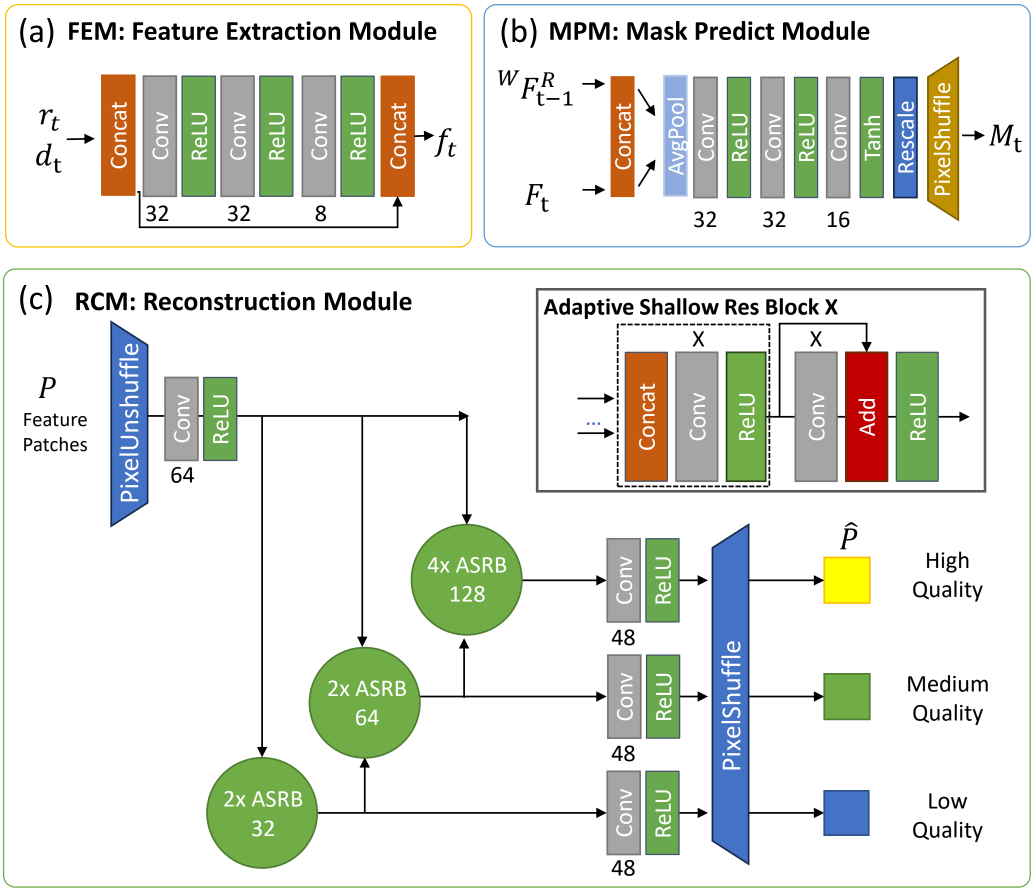


FIGURE 3 Neural Modules: (a) Feature Extraction Module. Low-resolution color buffer r_t and depth buffer d_t are first concatenated and a 8-channel feature is extracted by a three-layer convolutional neural network with ReLU activation. r_t and d_t are also concatenated at the end results in a 12-channel neural feature f_t. (b) Mask Prediction Module. Warped historical feature ${}^WF^R_{t-1}$ and current upscaled feature F_t are concatenated and then pooled with a average pooling layer with a 4×4 kernel and a stride of 4. Three convolutional layers generate a 16-channel feature and with a tanh, rescale and 4x4 pixelshuffle, one-channel mask M_t ranging from 0 to 1 at the high resolution is generated. (c) Reconstruction Module. Feature patch P is unshuffled with a ×4 PixelUnshuffle layer to get a 192-channel feature at low resolution and it is transformed to 64 channels. This feature goes through two 32-channel ASRBs, two 64-channel ASRBs and four 128-channel ASRBs to get the deep feature for high quality super-resolution. With a convolutional tail and a ×4 PixelShuffle, high quality RGB patch $\hat{P}$ is generated. Low quality and medium quality patches are generated with partial neural network in the similar way but with earlier exits and less computation. The number near a convolution layer indicates the output channel of the layer.

consisted of 60 frames of low-resolution color images, depth buffers, high-resolution target images, camera jitter information, as well as view and projection matrices, which were stored as images and text files. The 100 sequences were split into 80 training sequences, 10 validation sequences, and 10 test sequences. Following NSRR[5], the low-resolution images have a resolution of 400 × 225 with a vertical field-of-view of 50 degrees. The low-resolution images are rendered with no anti-aliasing while the high-resolution images are rendered in the resolution of 4800×2700 originally and downsampled to 1600×900 using a box-filter to get an effect of super sampling anti-aliasing. When rendering low-resolution contents, a global mip-map bias of -2 was applied to textures to maintain visual sharpness in low resolution.

We also capture a continuous sequence of head and eye motion for each scene in virtual reality (VR) using the HTC Vive Pro Eye system. The captured sequence consists of 300 frames and serves as the input for our foveated rendering demonstrations and evaluations. To ensure consistency with the recommended settings in SteamVR, we record the corresponding data with a

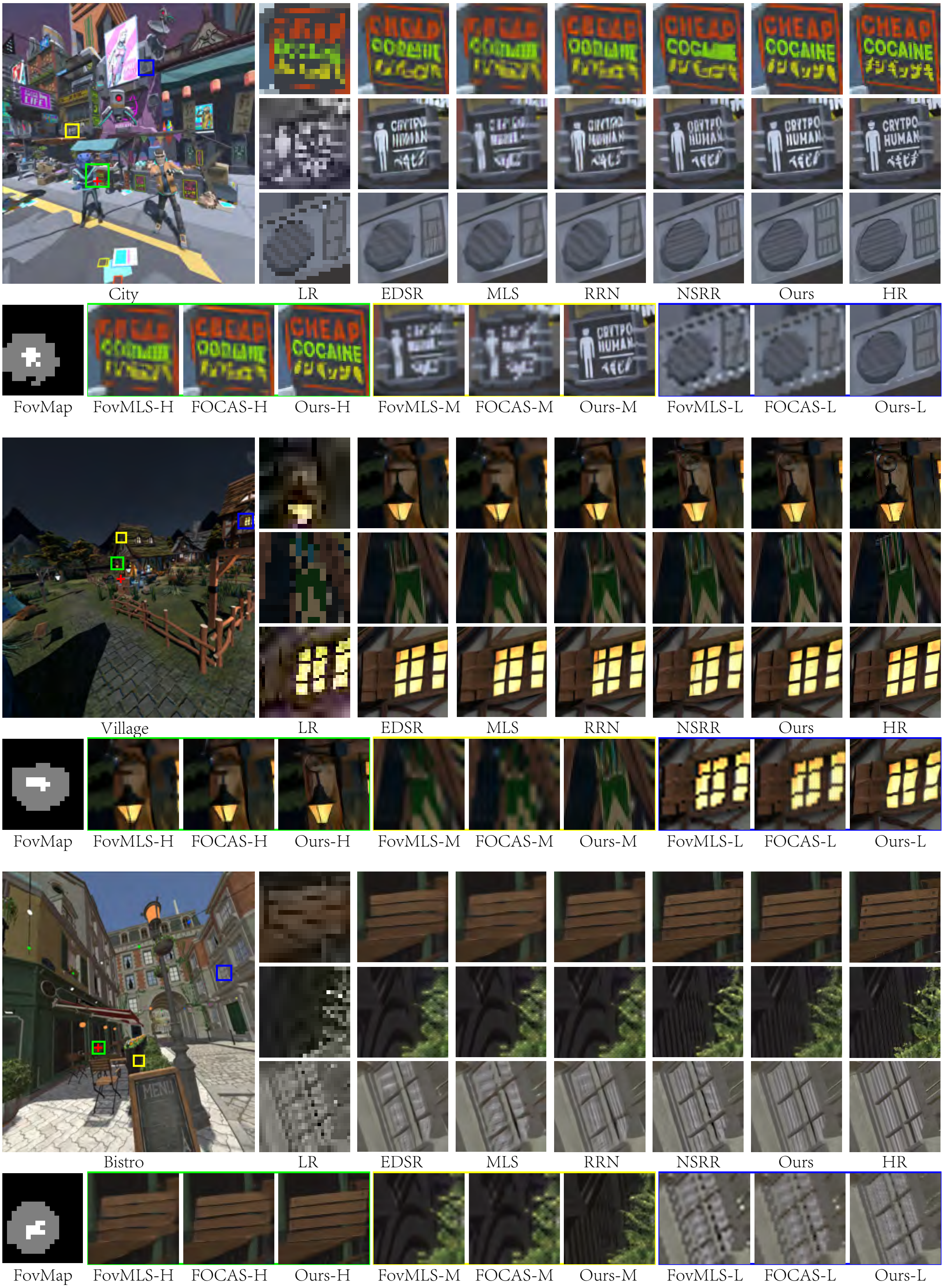


FIGURE 4 Each scene consists of three top rows showing cropped patches generated using full-image super-resolution methods. The last row showcases results obtained from foveated super-resolution methods (green outline for High quality, yellow for Medium, and blue for Low). The foveation map is presented in the bottom left corner. The large image on the left demonstrates the foveated result generated by our method. The fovea is indicated by a red cross.

target high-resolution of 2016 × 2240 and a vertical field-of-view of 104 degrees. These settings are chosen to simulate a real application environment and accurately represent the visual experience with foveated rendering.

We build our neural network using PyTorch 1.12.0, and the network is trained end-to-end with all the overlapping patches partitioned from the inputs with the corresponding high-resolution reference. Since our method aims to generate results with three different reconstruction tails with different parts of the neural network, we randomly output one of the three products and train the network in this mixed manner. The network was trained with ADAM optimizer with default hyperparameters, and the learning rate is set to $1e-4$. The network runs on a single Nvidia RTX 3090 graphic card or GTX 1080 Ti for time evaluation with neural modules optimized with TensorRT. Inspired by NSRR[5], we use a weighted combination of the perceptual loss computed from a pre-trained VGG-16 network[56] and the structural similarity index (SSIM)[57].

4.2 | Full Super-Resolution Evaluation

To show that our network works well in full image super-resolution, we compare our method with several state-of-the-art super-resolution work. We compared our method with the most related and state-of-the-art neural supersampling method, NSRR[5], considered as a benchmark in the field. Additionally, we included the EDSR[21] baseline, which is widely used and performs the second best according to the evaluation conducted by Xiao et al.[5]. To explore the effectiveness of the classical neural networks that have been applied in the context of foveated rendering[53,54], we compared our method with MSLapSRN (MLS)[58] and RRN[27]. MLS utilizes a shared network module for recursive processing and step-by-step upscaling of low-resolution images while RRN is a classical recurrent network for video super-resolution.

To ensure a fair comparison, we re-implemented all the methods and trained them using the same setup, including the dataset, learning rate, and loss function. EDSR[21] utilizes multiple simplified ResBlocks[22] as its core. We set the number of ResBlocks to 16 and the number of channels of each block to 128. NSRR[5], the state-of-the-art method, requires a 5-frame sequence input and is specifically designed for a window size of 5. We kept this configuration unchanged during our implementation. For recurrent methods like RRN[27] and our proposed method, which are not limited by a fixed window size, we set the length of training sequences to 5 and trained them in a recurrent manner. In the case of RRN[27], we set the number of ResBlocks to 10 and the number of channels to 128, consistent with FOCAS[53]. For MLS[58], we set the number of recursive blocks to 8, the depth of each block to 5, and the number of channels to 64.

Following NSRR[5], we use three quality metrics for evaluation: peak-to-noise ratio (PNSR), structural similarity index (SSIM)[57], and spatio-temporal entropic difference (STRRED)[59]. Table1 presents the average values of each metric obtained from 10 test sequences with 60 frames in each sequence. Our method and RRN were tested in a recurrent manner, as they can be executed recurrently without additional overhead, unlike NSRR[5], where the network is fixed and not scalable. To account for this, we omitted the first 4 frames since NSRR[5] does not provide corresponding outputs for them. Our method outperforms the existing methods with all metrics. And the STRRED shows that our method has better temporal stability.

Furthermore, Figure 4 presents a visual representation of the quantitative results obtained from several cropped regions of the full super-resolution images generated by these five methods.

TABLE 1 Quality comparisons of full super-resolution with existing methods on three virtual scenes. Results for each method are averaged across 10 test sequences in each scene. The number behind to name indicates how many frames the method takes as input while R indicates that it is tested in a recurrent manner. For PSNR and SSIM higher values mean higher quality, while for STRRED low values mean higher quality.

Metric	Dataset	EDSR(1)	MLS(1)	RRN(R)	NSRR(5)	Ours(R)
PSNR↑	City	27.09	26.14	26.99	29.27	**29.36**
	Village	27.49	26.92	27.52	29.55	**29.61**
	Bistro	25.55	24.81	25.55	27.63	**27.94**
SSIM↑	City	0.9172	0.8968	0.9158	0.9488	**0.9507**
	Village	0.8338	0.8139	0.8343	0.8893	**0.8907**
	Bistro	0.8064	0.7783	0.8064	0.8807	**0.8901**
STRRED↓	City	132.6	176.9	123.7	81.8	**74.3**
	Village	154.3	195.1	136.2	79.6	**75.1**
	Bistro	286.9	391.3	261.2	134.5	**114.0**

TABLE 2 Quality comparison of different super-resolution levels and the foveated super-resolution results on three scenes in VR setting. Different qualities of super-resolution are compared as the full image are super-resolved uniformly. H, M, L stand for high quality, medium quality and low quality, respectively. Foveated result are also compared with foveation-specific image quality metrics. Values of each method are average over all the frames in the test sequence for each scene in VR. For a fair comparison for foveated results, we keep a fixed gaze at the center and a fixed foveation map for each method.

Metric	Dataset	FovMLS L/M/H	FOCAS L/M/H	Ours L/M/H
PSNR↑	CityVR	25.86 / 27.84 / 28.25	26.28 / 26.50 / 28.84	**30.85 / 31.51 / 31.67**
	VillageVR	27.02 / 29.19 / 29.24	27.65 / 27.83 / 29.62	**31.38 / 31.98 / 32.19**
	BistroVR	24.74 / 26.70 / 26.71	25.47 / 25.77 / 27.09	**28.79 / 29.43 / 29.78**
SSIM↑	CityVR	0.886 / 0.929 / 0.938	0.894 / 0.894 / 0.946	**0.963 / 0.969 / 0.971**
	VillageVR	0.854 / 0.888 / 0.894	0.849 / 0.856 / 0.901	**0.929 / 0.935 / 0.937**
	BistroVR	0.758 / 0.808 / 0.821	0.769 / 0.777 / 0.834	**0.882 / 0.898 / 0.902**

Metric	Dataset	FovMLS-Fov	FOCAS-Fov	Ours-Fov
EWPSNR↑	CityVR	94.6377	95.4413	**98.5528**
	VillageVR	91.5090	91.9554	**95.2319**
	BistroVR	96.1158	96.8928	**100.4871**
FovVDP↑	CityVR	8.1720	8.1162	**8.5175**
	VillageVR	8.1184	8.0410	**8.6234**
	BistroVR	8.0312	7.9634	**8.6086**

4.3 | Foveated Super-Resolution Evaluation

In this section, we assess the quality of foveated super-resolution produced by our method and compare it with two super-resolution methods focusing on generating foveated results: (1) FovMLS[54], a method that leverages the characteristic of MLS[58] for foveated rendering; (2) FOCAS[53], a method employs the recurrent networks of RRN[27] for foveated super-resolution. To evaluate the performance, we utilize test datasets consisting of sequences recorded in VR, capturing head motion and gaze information as described in Section 4.1.

To generate foveated super-resolution results with varying image quality in different regions, all methods are required to upscale the input to target resolution with different levels of quality. We employ three quality levels (high-H, medium-M, low-L) for the inner, middle, and outer regions. FOCAS[53] achieves this by cropping the neural feature when passing it through stacked ResBlocks. Regions requiring higher visual quality are retained after cropping and undergo additional neural blocks. The resulting mixed feature is combined and transformed into the final result. Similarly, FovMLS[54] crops the middle region of the input image and upscales it to a 2×2 size with medium quality. It then recursively crops the inner region and upscales it to a 4×4 size with high quality. In our method, we employ a partition-assemble scheme and a multi-exit reconstruction module to generate reconstructed results, as described in Section 3.2.

In Figure 4, we present a qualitative demonstration of the foveated super-resolution images produced by our method. We compare the cropped regions at different quality levels with other foveated super-resolution methods. And in the upper rows of Table 2, we compare the PNSR and SSIM of these methods. Across all quality levels, our method outperforms the other two by a significant margin. Furthermore, our method exhibits a more gradual degradation than the other methods, indicating a smoother transition in image quality.

To ensure a fair and quantitative comparison of the foveated results' quality, we employ several control strategies. Firstly, we fix the gaze at the center position. While FOCAS and FovMLS support only nested concentric rectangular foveation patterns, our method can adapt to any pattern. Therefore, we also adopt a rectangular foveation pattern for our method in this evaluation. To determine the division of the three foveal layers and the corresponding sizes and borders of each rectangle, we collect the foveation maps generated by Tursun et al.[15] for each test frame with a center gaze. We compute the ratio of the corresponding areas for each quality level. On average, the inner region occupies 4% of the total area, while the middle region occupies 30% of the area. These correspond to sizes of $[(428, 428), (1240, 1240)]$ at the target resolution.

For FOCAS and FovMLS, we keep the size of the inner region and middle region the same as ours to showcase the quantitative quality each method can achieve in the same-sized visual region division. For FOCAS, it processes the inner fovea region with 10 blocks, the middle region with 8 blocks and 1 block for the entire outer region (FOCAS-20). For FovMLS, the inner region is super-resolved by $\times 2$ neural upscaling double times, the middle region is super-resolved by $\times 2$ neural upscaling and

$\times 2$ bicubic upscaling and the outer region is upscaled solely by $\times 4$ bicubic interpolation. In the bottom rows of Table 2, we use EWPSNR[60] and FovVDP[61] as the metric to evaluate the image and video quality of foveated results. Our method also outperforms the other foveated methods by a large margin.

4.4 | Runtime

We demonstrate the runtime breakdown of major steps of our method in Table 3 with typical mainstream last-generation and current-generation desktop GPUs as the testbeds for two common targeting resolutions.

The runtime consists of two major parts: neural temporal accumulator and reconstruction. The runtime of neural temporal accumulator consists of feature extraction, mask prediction, upscaling of motion vector and feature, feature warping as well as feature blending(Equation 1). The reconstruction neural network module dominates the workload. With the foveated super-resolution method we proposed, a majority of the computation can be saved. Partition and assembly operations can been seen as the necessary overhead of our methods but they can be executed in a relatively negligible amount of time. The total frame-time of our method includes all the time of neural temporal accumulator and foveated reconstruction, and our FovSR can run in real-time with a frame-time of $9.867ms$ for 1080p target resolution on a GTX 1080Ti GPU and $8.018ms$ for 4K on a RTX 3090 achieving around 54% and 43% reduction compared with full super-resolution. Compared to full-size native rendering with some ray-traced global illumination effect, which can take over $100ms$ at 1080p[5], native rendering at 480×270 with foveated super-resolution offers a significantly more efficient approach while still providing a visually satisfying experience.

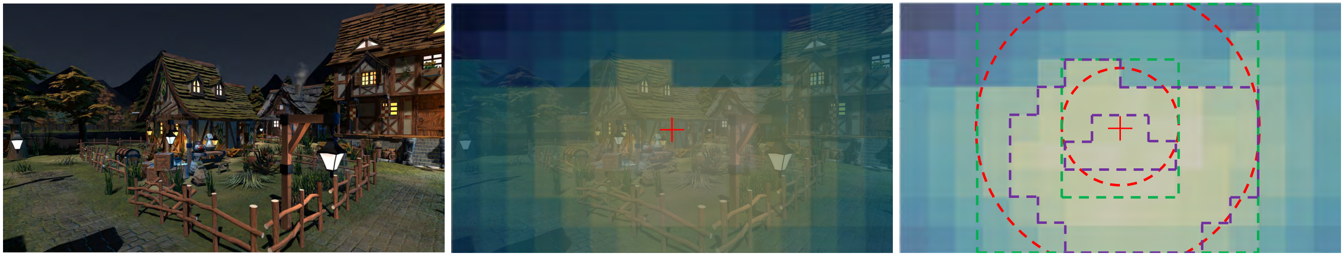
TABLE 3 Detailed Time Evaluation. The time of neural temporal accumulator and foveated reconstruction is shown for 4×4 super-resolution on two GPUs targeting two resolutions. The steps which are mainly executions of neural modules are indicated with N in the brackets and they are accelerated with TensorRT at 16-bit precision. Other steps indicated with * are texture array manipulation like upscaling, warping, blending and partition-assembly operations which are implemented with graphics API and CUDA implementation. The total time of foveted super-resolution, fullsize super-resolution and reduction ratio are listed at the last rows. The unit of time is millisecond (ms).

Devices/Resolution		GTX1080Ti 1080p	GTX1080Ti 4K	RTX3090 1080p	RTX3090 4K
Neural Temporal Accumulator	Motion Vector Upscaling(*)	0.116	0.503	0.032	0.093
	Warping(*)	0.772	3.019	0.125	0.625
	Feature Extraction Module(N)	0.583	2.378	0.134	0.425
	Feature Upscaling(*)	0.389	1.583	0.093	0.414
	Mask Prediction Module(N)	1.324	5.123	0.262	1.040
	Blending(*)	0.814	3.135	0.132	0.615
Foveated Reconstruction	Partition(*)	0.312	1.151	0.121	0.424
	Patch Reconstruction(N)	5.256	21.183	1.093	4.061
	Assembly(*)	0.301	0.871	0.107	0.321
FovSR Total		**9.867**	**38.946**	**2.099**	**8.018**
High Quality FullSR Total		21.121	85.333	3.732	14.167
FovSR Reduction Ratio		53.28%	54.36%	43.76%	43.40%

The comparison of runtime and GFLOPs (giga floating point of operations) of our methods and others in fullsize and foveated super-resolution are shown in Table 4 on the GTX 1080Ti for 1080p target resolution. Our method can achieve better fullsize and foveated image quality in a shorter runtime and with less computations. It is worth noting that the speed-up achieved in our method, relative to full image super-resolution, is not fixed and can be adjusted based on different foveation maps. In Table 4, we present a specific frame from the Village scene along with a representative gaze point and foveation map generated using the method proposed in Tursun et. al.[15]. After thresholding and discretization, the foveation map is transformed into a three-level foveation pattern as shown in the blue outlined regions containing several image patches. The regions can be in arbitrary shapes and we refer to as free form foveated super-resolution (Ours-FreeForm). In traditional foveated super-resolution methods like FovMLS and FOCAS that without our partition and assembly scheme, maintaining the quality requirements for all regions necessitates the use of either circum-rectangles (FovA - highlighted in green) or circumcircle (FovB - highlighted in red) as the foveation map. The corresponding runtime for these foveation patterns is shown in Table 4. It can be seen that our method provides more freedom and flexibility in choosing foveation maps of any shape, thereby maximizing the potential for reducing computational workload.

TABLE 4 Time and computation comparison of our method and others on GTX 1080Ti for fullsize and foveated super-resolution with a target resolution of 1080p. The figures shows a rendered image with its foveation map and three foveation strategies. FovA (Green), FovB (Red) and FreeForm (Blue) stand for foveation strategies as described in Section 4.4 with corresponding reconstruction time listed. For foveated super-resoution, the time and GFLOPs of FovMLS and FOCAS are tested with FovA pattern. For GFLOPs, we only count the operations of neural modules of each method. The unit of time is millisecond (ms).

Time of Fullsize Super-Resolution	EDSR 156.732	MLS 220.982	RRN 54.655	NSRR 74.985	Ours-FullSR-H 21.121
GFLOPs of Fullsize Super-Resolution	EDSR 1020.4	MLS 1055.4	RRN 435.6	NSRR 339.5	Ours-FullSR-H 123.8
Time of Foveated Super-Resolution	FovMLS 22.829	FOCAS 23.508	Ours-FovA 12.295	Our-FovB 11.614	Ours-FreeForm 9.867
GFLOPs of Foveated Super-Resolution	FovMLS 105.8	FOCAS 185.8	Ours-FovA 48.4	Our-FovB 44.5	Ours-FreeForm 39.1



4.5 | User Study

We conducted a psychophysical experiment to evaluate the perceptual quality of our foveated super-resolution results (Ours-FovSR) in comparison with our full super-resolution results (Ours-FullSR), the ground truth at full resolution (GT), and the results obtained using two other foveated super-resolution methods, FOCAS and FovMLS. To ensure a precise comparison and eliminate the variability in gaze behavior across different trials, we constructed each set of stimuli using static stereo foveated images, as depicted in Figure 4. For each test scene, we selected two frames along with their corresponding gaze for subjective evaluation.

During the study, participants wore HTC Vive Pro Eye headsets and were seated to view the stereo images. The target gaze position was indicated by a red cross on the stimuli images. Participants were instructed to keep their gaze around the cross throughout the experiment. To ensure consistent and reliable gaze behavior, eye tracking was employed to monitor participants' gaze direction. If participants failed to fix their gaze on the target position, the images would temporarily black out. A total of 12 volunteers (3 females, mean age = 24.3) participated in and completed the study. All participants had a normal or corrected-to-normal vision and were unaware of the specific intention of the study. During each test trial, participants were presented with image pair A for a duration of 2 seconds, followed by a black screen for 0.75 seconds, and then image pair B for another 2 seconds. Subsequently, the participants were asked to perform a two-alternative-forced-choice (2AFC) to choose the better pair with better overall image quality and then score the difference between the two pairs from 1 (significant perceptual difference), 2 (minimal perceptual difference), 3 (perceptually identical). Based on the choice of A v.s. B, the score can be interpreted as a perceptual score of A with regard to B ranging from -2 to 2 where 0 represents equal quality. Several warm-up trials were conducted initially to familiarize participants with the procedure. Totally 96 trials for each intended comparison were conducted. Between trails, participants were free to relax as much as they wanted. The sequence of trials were randomized for every participant.

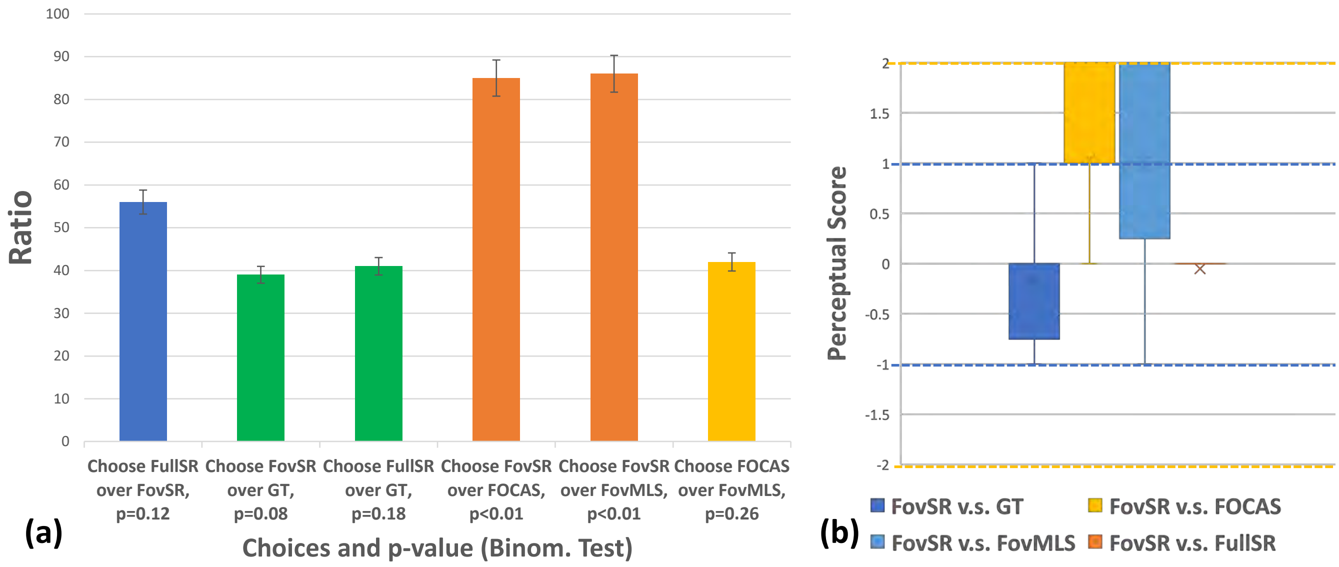


FIGURE 5 The result of user study. (a) The choice ratio of 2AFC. Each bar represents the ratio of choice explained under it. The p value of binomial test is also listed. (b) A box plot of perceptual score of our FovSR with regard to others. The perceptual score of A v.s. B means, taking B as a reference, how much the A is perceptually better than B, where 0 means the user has no certain preference.

The results in Figure 5 show that it is a random choice between FovSR v.s. FullSR ($p = 0.12$, binom. test), FovSR v.s. GT ($p = 0.08$), FullSR v.s. GT ($p = 0.18$) showing that our foveated super-resolution is perceptually identical statistically as ground truth and full image super-resolution. And our foveated super-resolution is preferred significantly over the other methods we compared ($p<0.05$) while the perceptual quality of FOCAS and FovMLS is similar ($p=0.26$). We further check the perceptual quality of our foveated super-resolution and find that it is slightly worse than ground truth but much better the other two. It also shows equal quality as full super-resolution, proving that our foveation strategy is effective.

4.6 | Analysis and ablation study

Here we analyze the necessity of several essential designs of our method that help us to get a better super-resolution quality. The reported average quality scores are tested with the test set of the City scene, all with the full high-quality super-resolution output. All analyses in this section follow the same setting.

4.6.1 | Historical neural feature

Unlike classical temporal supersampling methods that typically store and maintain raw color buffers as the key historical data, our method introduces the concept of maintaining and updating a 12-channel historical neural feature from frame to frame. This feature comprises raw color and depth information in the first 4 channels and learned deep features extracted from FEM. To ascertain the necessity of this additional deep feature in our method, we conduct an ablation experiment where we only maintain and update the color and depth buffer while adjust the input channel of MPM and RCM accordingly. As shown in Figure 6 (a), the characters are noisy, indicating that the details are not properly restored when the network is trained without this neural feature. The numerical results demonstrate the improvements gained from utilizing the historical neural feature in Table 5 comparing “Our-H” with “w/o neural feature”.

4.6.2 | Temporal Neural Masking

Effective temporal accumulation necessitates a reliable filtering method to eliminate invalid and outdated samples while avoiding accumulating potential errors. This aspect is particularly crucial and challenging for super-resolution tasks. To evaluate the effectiveness of our mask prediction module, we compare it with a simple handcrafted filtering mechanism. In this mechanism, we reconstruct the depth of the previous frame using the current depth buffer[11]. By comparing the warped reconstructed depth with the current depth, we apply a threshold to identify and mask out geometrically invalid samples. For the rectification of the historical neural feature, we calculate the Axis-Aligned Bounding Box (AABB) for each channel of the current upscaled feature F_t within a 3×3 neighborhood. Subsequently, we straightforwardly clamp the warped historical feature ${}^{W}F^{R}_{t-1}$ values within the determined bounding box. By replacing our neural mask prediction module with this manual solution, the performance of our method experiences a significant decrease, as shown in Table 5 which demonstrate that our neural masking scheme is necessary for our method to achieve better super-resolution quality. In Figure 6, a decrease of visual quality can been seen when comparing "Our-H" with "w/o neural masking".

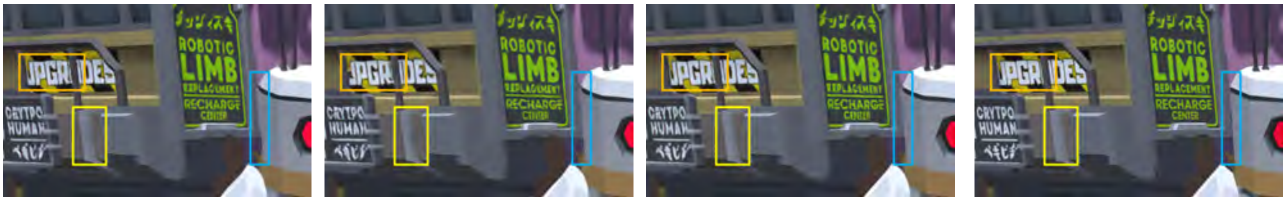

(a) w/o neural feature
(b) w/o neural masking
(c) Ours-H
(d) High Resolution Reference

FIGURE 6 Visual comparison of our methods and modified ones without neural feature and neural masking. A region containing rich details, changing shadows and moving edges are cropped from a specific frame generated by our full method (Ours-H) and the alternative methods without neural feature, neural masking and the high resolution reference.

4.6.3 | Reconstruction Module

This paper presents a reconstruct network structure with stacked ASRBs and additional skip connections. This architectural not only enables us to generate different levels of image quality but also improves the overall image reconstruction. To validate the contribution of these modifications, we delete the skip connections, and replace the every two ASRBs in the core of our reconstruction module with one ResBlocks[21,53]. From the results shown in Table 5, we can see the improvement brought by skip connections and the use of ASRBs.

In order to compare our design with traditional super-resolution networks, we also employ the design of EDSR and RFLN[62] (the winner of main track of NTIRE 2022 Efficient Super-Resolution Challenge[63]) instead . To ensure a similar runtime of different reconstruction modules, we replace our reconstruction module with 10 ResBlocks of 64 channels, 6 RFLBs (residual local feature block) of 52 channels. The quantitative results of PSNR, SSIM are shown in Table 5. Our reconstruction module outperforms other efficient neural networks in image quality with similar runtime.

4.7 | Limitations and Future Work

While our method offers an effective solution for foveated super-resolution in VR applications compared to other methods, the current implementation for higher resolutions on low-end desktop GPUs can be further optimized computationally. Potential optimizations include network structural optimization, INT8 network quantization[7], and hardware-specific engineering. While we are currently targeting on PCVR devices, implementation and test of our method on mobile and standalone devices need further efforts in the future[64].

There is still room for improvement in achieving perfect super-resolution. Future investigations can explore more complex and longer jittering sequences instead of the fixed ones used in our current work. Additionally, exploring advanced resampling

TABLE 5 Quality comparisons of ablation study. Results for each method are averaged across 10 test sequences in City scene. Ours-H represents the full model of our method. We remove the FEM and maintain only color and depth temporal the super-resolution result is shown in the column of w/o neural feature. We remove the MPM and replace the neural accumulator with a manul solution only color and show the results in the column of w/o neural masking. We replace the reconstruction module with some other design and the results are shown in the rest columns.

Settings	Ours-H	w/o neural feature	w/o neural masking	Reconstruction Module Replacement			
				w/o skip connections	SRB to ResBlock[21]	ResBlock b10c64[21]	RFLB b6c52[62]
PSNR↑	29.36	28.71	28.50	29.20	29.13	29.25	29.12
SSIM↑	0.9507	0.9420	0.9398	0.9491	0.9473	0.9476	0.9479

techniques beyond bilinear sampling could lead to better results. FuseSR[65] achieves higher scale super-resolution (8×8) with the help of additional high resolution G-buffer and demodulation. This recent work indicates a future direction to further boost our method. Since our method targets VR applications, considering stereo information from both eyes is an important aspect that remains in future work.

5 | CONCLUSIONS

In this paper, we present a novel neural foveated super-resolution method for real-time rendering in virtual reality. Our approach leverages the limitations of human vision and applies foveation to enhance the super-resolution of the rendered content. We propose a neural accumulator that can be trained end-to-end to effectively accumulate and update temporally amortized rendering information using visual and geometric cues inherent in the rendering pipeline. We introduce a neural reconstruction module that efficiently translates the maintained historical features into high-resolution quality. The output quality can be flexibly controlled, allowing for precise manipulation. Additionally, we propose a partition-and-assemble scheme that enables the generation of foveated super-resolution with customizable foveation patterns or quality distribution, maximizing computational efficiency while maintaining perceptually acceptable results. Our method surpasses existing approaches in foveated super-resolution and holds promise in delivering perceptually high-quality visual experiences for virtual reality, mitigating concerns about the rendering workload.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (2018YFB1004902). We thank all the participants for the user study and reviewer for their advice.

REFERENCES

1. Meng X, Zheng Q, Varshney A, Singh G, Zwicker M. Real-time Monte Carlo Denoising with the Neural Bilateral Grid. In: EGSR (DL); 2020. p. 13–24.
2. Fan H, Wang R, Huo Y, Bao H. Real-time Monte Carlo Denoising with Weight Sharing Kernel Prediction Network. In: Computer Graphics Forum. vol. 40. Wiley Online Library; 2021. p. 15–27.
3. Briedis KM, Djelouah A, Meyer M, McGonigal I, Gross M, Schroers C. Neural frame interpolation for rendered content. ACM Transactions on Graphics (TOG). 2021;40(6):1–13.
4. Guo J, Fu X, Lin L, Ma H, Guo Y, Liu S, et al. ExtraNet: real-time extrapolated rendering for low-latency temporal supersampling. ACM Transactions on Graphics (TOG). 2021;40(6):1–16.
5. Xiao L, Nouri S, Chapman M, Fix A, Lanman D, Kaplanyan A. Neural supersampling for real-time rendering. ACM Transactions on Graphics (TOG). 2020;39(4):142–1.
6. NVIDIA. Deep Learning Super Sampling (DLSS) Technology | NVIDIA. https://www.nvidia.com/en-us/geforce/technologies/dlss/ 2020. (Accessed on 01/23/2023).
7. Thomas MM, Vaidyanathan K, Liktor G, Forbes AG. A reduced-precision network for image reconstruction. ACM Transactions on Graphics (TOG). 2020;39(6):1–12.
8. Liu H, Ruan Z, Zhao P, Dong C, Shang F, Liu Y, et al. Video super-resolution based on deep learning: a comprehensive survey. Artificial Intelligence Review. 2022;p. 1–55.
9. Yang W, Zhang X, Tian Y, Wang W, Xue JH, Liao Q. Deep learning for single image super-resolution: A brief review. IEEE Transactions on Multimedia. 2019;21(12):3106–3121.
10. Intel. Intel Arc- Xe Super Sampling. https://www.intel.com/content/www/us/en/products/docs/arc-discrete-graphics/xess.html 2021. (Accessed on 01/23/2023).
11. AMD. AMD FidelityFX Super Resolution | AMD. https://www.amd.com/en/technologies/fidelityfx-super-resolution 2023. (Accessed on 01/23/2023).

12. UnrealEngine. Screen Percentage with Temporal Upscale in Unreal Engine | Unreal Engine 5.0 Documentation. https://docs.unrealengine.com/5.0/en-US/screen-percentage-with-temporal-upscale-in-unreal-engine/ 2022. (Accessed on 01/23/2023).
13. Guenter B, Finch M, Drucker S, Tan D, Snyder J. Foveated 3D graphics. ACM Transactions on Graphics (TOG). 2012;31(6):1–10.
14. Patney A, Salvi M, Kim J, Kaplanyan A, Wyman C, Benty N, et al. Towards foveated rendering for gaze-tracked virtual reality. ACM Transactions on Graphics (TOG). 2016;35(6):1–12.
15. Tursun OT, Arabadzhiyska-Koleva E, Wernikowski M, Mantiuk R, Seidel HP, Myszkowski K, et al. Luminance-contrast-aware foveated rendering. ACM Transactions on Graphics (TOG). 2019;38(4):1–14.
16. Krajancich B, Kellnhofer P, Wetzstein G. Towards Attention–aware Foveated Rendering. ACM Transactions on Graphics (TOG). 2023;42(4):1–10.
17. Dong C, Loy CC, He K, Tang X. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence. 2015;38(2):295–307.
18. Kim J, Lee JK, Lee KM. Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 1646–1654.
19. Ledig C, Theis L, Huszár F, Caballero J, Cunningham A, Acosta A, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 4681–4690.
20. Zhang Y, Tian Y, Kong Y, Zhong B, Fu Y. Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 2472–2481.
21. Lim B, Son S, Kim H, Nah S, Mu Lee K. Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops; 2017. p. 136–144.
22. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.
23. Liu J, Tang J, Wu G. Residual feature distillation network for lightweight image super-resolution. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. Springer; 2020. p. 41–55.
24. Li Y, Zhang Y, Timofte R, Van Gool L, Yu L, Li Y, et al. NTIRE 2023 challenge on efficient super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 1921–1959.
25. Jo Y, Oh SW, Kang J, Kim SJ. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 3224–3232.
26. Haris M, Shakhnarovich G, Ukita N. Recurrent back-projection network for video super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 3897–3906.
27. Isobe T, Zhu F, Jia X, Wang S. Revisiting temporal modeling for video super-resolution. arXiv preprint arXiv:200805765. 2020;.
28. Caballero J, Ledig C, Aitken A, Acosta A, Totz J, Wang Z, et al. Real-time video super-resolution with spatio-temporal networks and motion compensation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 4778–4787.
29. Tao X, Gao H, Liao R, Wang J, Jia J. Detail-revealing deep video super-resolution. In: Proceedings of the IEEE International Conference on Computer Vision; 2017. p. 4472–4480.
30. Wang X, Chan KC, Yu K, Dong C, Change Loy C. Edvr: Video restoration with enhanced deformable convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2019. p. 0–0.
31. Tian Y, Zhang Y, Fu Y, Xu C. Tdan: Temporally-deformable alignment network for video super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 3360–3369.
32. Xue T, Chen B, Wu J, Wei D, Freeman WT. Video enhancement with task-oriented flow. International Journal of Computer Vision. 2019;127(8):1106–1125.
33. Chan KC, Zhou S, Xu X, Loy CC. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022. p. 5972–5981.
34. Zhang Y, Zhang K, Chen Z, Li Y, Timofte R, Zhang J, et al. NTIRE 2023 challenge on image super-resolution (x4): Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023. p. 1864–1883.
35. Yang L, Sander PV, Lawrence J. Geometry-aware framebuffer level of detail. In: Computer Graphics Forum. vol. 27. Wiley Online Library; 2008. p. 1183–1188.
36. Herzog R, Eisemann E, Myszkowski K, Seidel HP. Spatio-temporal upsampling on the GPU. In: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games; 2010. p. 91–98.
37. Akeley K. Reality engine graphics. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques; 1993. p. 109–116.
38. Lottes T. FXAA. https://developer.download.nvidia.cn/assets/gamedev/files/sdk/11/FXAA_WhitePaper.pdf 2009. (Accessed on 01/23/2023).
39. Reshetov A. Morphological antialiasing. In: Proceedings of the Conference on High Performance Graphics 2009; 2009. p. 109–116.
40. Jimenez J, Echevarria JI, Sousa T, Gutierrez D. SMAA: Enhanced subpixel morphological antialiasing. In: Computer Graphics Forum. vol. 31. Wiley Online Library; 2012. p. 355–364.
41. Yang L, Liu S, Salvi M. A survey of temporal antialiasing techniques. In: Computer graphics forum. vol. 39. Wiley Online Library; 2020. p. 607–621.
42. Nehab D, Sander PV, Lawrence J, Tatarchuk N, Isidoro JR. Accelerating real-time shading with reverse reprojection caching. In: Graphics hardware. vol. 41; 2007. p. 61–62.
43. Yang L, Nehab D, Sander PV, Sitthi-Amorn P, Lawrence J, Hoppe H. Amortized supersampling. ACM Transactions on Graphics (TOG). 2009;28(5):1–12.
44. Karis B. High Quality Temporal Supersampling 2014. In ACM Trans. Graph. (Advances in Real-Time Rendering).
45. Xiao K, Liktor G, Vaidyanathan K. Coarse pixel shading with temporal supersampling. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games; 2018. p. 1–7.
46. Thomas MM, Liktor G, Peters C, Kim S, Vaidyanathan K, Forbes AG. Temporally Stable Real-Time Joint Neural Denoising and Supersampling. Proceedings of the ACM on Computer Graphics and Interactive Techniques. 2022;5(3):1–22.
47. Ye J, Xie A, Jabbireddy S, Li Y, Yang X, Meng X. Rectangular Mapping-based Foveated Rendering. In: 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE; 2022. p. 756–764.

48. Meng X, Du R, Zwicker M, Varshney A. Kernel foveated rendering. Proceedings of the ACM on Computer Graphics and Interactive Techniques. 2018;1(1):1–20.
49. Vaidyanathan K, Salvi M, Toth R, Foley T, Akenine-Möller T, Nilsson J, et al. Coarse pixel shading. In: Proceedings of High Performance Graphics; 2014. p. 9–18.
50. Weier M, Roth T, Kruijff E, Hinkenjann A, Pérard-Gayot A, Slusallek P, et al. Foveated real-time ray tracing for head-mounted displays. In: Computer Graphics Forum. vol. 35. Wiley Online Library; 2016. p. 289–298.
51. Kaplanyan AS, Sochenov A, Leimkühler T, Okunev M, Goodall T, Rufo G. DeepFovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. ACM Transactions on Graphics (TOG). 2019;38(6):1–13.
52. Deng N, He Z, Ye J, Duinkharjav B, Chakravarthula P, Yang X, et al. FoV-NeRF: Foveated Neural Radiance Fields for Virtual Reality. IEEE Transactions on Visualization and Computer Graphics. 2022;.
53. Wang L, Hajiesmaili M, Sitaraman RK. Focas: Practical video super resolution using foveated rendering. In: Proceedings of the 29th ACM International Conference on Multimedia; 2021. p. 5454–5462.
54. Nam H, Kang H. Complexity-Reduced Super Resolution for Foveation-Based Driving Head Mounted Displays. IEEE Access. 2021;9:140042–140049.
55. Zhou Z, Siddiquee MMR, Tajbakhsh N, Liang J. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. IEEE transactions on medical imaging. 2019;39(6):1856–1867.
56. Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer; 2016. p. 694–711.
57. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing. 2004;13(4):600–612.
58. Lai WS, Huang JB, Ahuja N, Yang MH. Fast and accurate image super-resolution with deep laplacian pyramid networks. IEEE transactions on pattern analysis and machine intelligence. 2018;41(11):2599–2613.
59. Soundararajan R, Bovik AC. Video quality assessment by reduced reference spatio-temporal entropic differencing. IEEE Transactions on Circuits and Systems for Video Technology. 2012;23(4):684–694.
60. Li Z, Qin S, Itti L. Visual attention guided bit allocation in video compression. Image and Vision Computing. 2011;29(1):1–14.
61. Mantiuk RK, Denes G, Chapiro A, Kaplanyan A, Rufo G, Bachy R, et al. FovVideoVDP: A Visible Difference Predictor for Wide Field-of-View Video. ACM Trans Graph. 2021 jul;40(4). Available from: https://doi.org/10.1145/3450626.3459831.
62. Kong F, Li M, Liu S, Liu D, He J, Bai Y, et al. Residual local feature network for efficient super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2022. p. 766–776.
63. Li Y, Zhang K, Timofte R, Van Gool L, Kong F, Li M, et al. NTIRE 2022 Challenge on Efficient Super-Resolution: Methods and Results. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2022. p. 1061–1101.
64. Yang S, Zhao Y, Luo Y, Wang H, Sun H, Li C, et al. MNSS: Neural Supersampling Framework for Real-Time Rendering on Mobile Devices. IEEE Transactions on Visualization and Computer Graphics. 2023;.
65. Zhong Z, Zhu J, Dai Y, Zheng C, Huo Y, Chen G, et al.. FuseSR: Super Resolution for Real-time Rendering through Efficient Multi-resolution Fusion 2023.

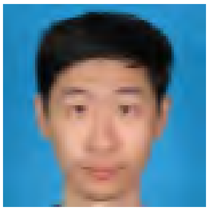
AUTHOR BIOGRAPHY

Jiannan Ye. Jiannan Ye is currently pursuing the Ph.D. degree at Digital ART Lab of School of Software at Shanghai Jiao Tong University. His research interests include foveated rendering, neural rendering and virtual reality.

Xiaoxu Meng. Xiaoxu Meng is currently employed as a Senior Researcher at Tencent America in Los Angeles. She earned her Ph.D. in Computer Science from the University of Maryland, College Park. Her research focuses on the intersection of computer graphics, computer vision, and virtual reality.

Daiyun Guo. Daiyun Guo received her B.S. degree from Shanghai Jiao Tong University in 2022. Currently, she is a Master student at Digital ART Lab of School of Software at Shanghai Jiao Tong University. Her research interests include rendering and interaction.

Cheng Shang. Cheng Shang is a Ph.D. student at the School of Software, Shanghai Jiaotong University. His research interests include computer vision, deep learning, image processing, and image understanding.



Haotian Mao. Haotian Mao received his B.S. degree from Shanghai Jiao Tong University in 2022. Currently, He is a Master student at Digital ART Lab of School of Software at Shanghai Jiao Tong University specializing in virtual reality and neural rendering.

Xubo Yang. Xubo Yang received his Ph.D. degree in computer science from the State Key Lab of CAD and CG, Zhejiang University, in 1998. He is a full professor and the director of Digital ART Lab of School of Software at Shanghai Jiao Tong University. His research interests include computer graphics, virtual reality and human computer interaction. He is the corresponding author of this paper.