International Journal on Semantic Web and Information Systems
Volume 19 • Issue 1

A Cloud-Edge Collaborative Gaming Framework Using AI-Powered Foveated Rendering and Super Resolution

Xinkun Tang, Academy of Broadcasting Science, China
https://orcid.org/0000-0003-0198-2467
Ying Xu, Academy of Broadcasting Science, China*
https://orcid.org/0000-0003-0904-1477
Feng Ouyang, Academy of Broadcasting Science, China
Ligu Zhu, Communication University of China, China
Bo Peng, Communication University of China, China

ABSTRACT

Cloud gaming (CG) has gradually gained popularity. By leveling shared computing resources on the cloud, CG technology allows those without expensive hardware to enjoy AAA games using a low-end device. However, the bandwidth requirement for streaming game video is high, which can cause backbone network congestion for large-scale deployment and expensive bandwidth bills. To address this challenge, the authors proposed an innovative edge-assisted computing architecture that collaboratively uses AI-powered foveated rendering (FR) and super-resolution (SR). Using FR, the cloud server can stream gaming video in lower resolution, significantly reducing the transmitted data volume. The edge server will then upscale the video using a game-specific SR model, recovering the quality of the video, especially for the areas players pay the most attention. The authors built a prototype system called FRSR and did thorough, objective comparative experiments to demonstrate that this architecture can reduce bandwidth usage by 39.47% compared with classic CG implementation for similar perceived quality.

KEYWORDS

Cloud Gaming, Collaborative, Edge Computing, FOCAS, Foveated Rendering, FRSR, ROI Prediction, Super Resolution

INTRODUCTION

The main benefits of cloud gaming (CG) include no need to download and install or click and play. CG technology also does not require expensive hardware configuration, and the game subscription model can reduce game costs. However, the price paid in exchange for these advantages is high bandwidth consumption because the essence of CG is a video transmission that emphasizes timely interaction.

DOI: 10.4018/IJSWIS.321751
*Corresponding Author

According to Zhang et al. (2019), a single user's recommended downstream bandwidth for an acceptable-quality gaming experience is 3 megabits per second (Mb/s). Take 1,080-pixel (P) resolution as an example: If the transmitted image is 24 bits deep, for a 30 frames per second (fps) game experience, the required bandwidth is 18.66 megabytes per second (MB/s) without compression. Limited by the 100MB bandwidth of the 4G network, it can guarantee only up to five clients for regular use simultaneously. In addition, if the gaming service provider uses a public cloud, such as Amazon Web Services (AWS), the charging standard for transferring data from the AWS EC2 server to all over the world is as large as $0.02 per GB (Amazon, 2022), so the uncompressed cloud game video stream will consume $1.26 per hour for data transfer only. Moreover, this number will quickly become unbearable when the number of users increases to hundreds of millions. On the other hand, heavy gaming traffic can cause congestion in the backbone network, seriously affecting the performance of other online services. Therefore, knowing how to compress the transmission bandwidth of cloud games is the key to improve the game experience and save infrastructure costs.

To tackle this challenge, we propose a novel architecture that uses AI-based compression and enhancement algorithms to minimize the transmission volume of game video without sacrificing the perceived experiences. The methodology behind the architecture is that by deploying a game-specific trained SR model to the edge side beforehand, the edge server essentially pre-saves game video enhancement information closer to the player, thereby allowing the edge server to send relative low-quality data and save a significant amount of bandwidth. The trade-off is that more computing resources on the player side are used for real-time enhancement, and this requirement can be perfectly satisfied by the edge computing paradigm in which computation is moved as close to the end users as possible. Here is a summary of the main contributions of this paper:

- We propose an innovative cloud-edge collaborative computing architecture for CG. This architecture fully uses the computing power advantages of the cloud and the edge. The computing power is exchanged to reduce the amount of transmitted data, achieving economic benefits.
- We describe how we implemented an end-to-end prototype gaming system FRSR. This system is the first one using state of the art FR and SR technology to evaluate the collaborative computing architecture. The SR model proposed in Wang et al. (2021) is customized to support multiple regions of interest (ROIs).
- We describe our thorough objective experiments on FRSR, which were compared to four other cloud gaming implementations for four different game genres and recorded metrics, including peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), bits per pixel (BPP), and processing time. The results demonstrate FRSR's effectiveness on bandwidth reduction while maintaining roughly the same level of perceived quality.

BACKGROUND AND RELATED RESEARCH

Cloud Gaming

Cloud gaming is a technology that runs the game on the cloud using the GPU for rendering and streaming the generated game images to the client. Under cloud gaming architecture, the cloud server handles both the storage and execution of the played game, and the rendered game images are transferred to the player through the network in the form of a real-time video stream. Thus, the client on the player side is responsible only for basic decoding and playing of the video stream. The game service providers can purchase cloud resources on demand to save cost, and game developers do not need to develop corresponding versions for different platforms.

With all these benefits mentioned, CG is facing challenges from several aspects (Dick et al., 2005). First, the game player can tolerate only 80–100ms delay time (20 ms for server-side and client-side processing time, and 80 ms for network transfer time) (OL2, Inc., 2015). The farther the distance

is between the cloud and client sides, the longer it takes to transfer the video stream. Second, video streaming has a high demand for bandwidth, usually 1–5 megabits (Mb), according to OnLive (OL2, Inc., 2015). Last but not least, the centralized service system limits expanding the service scope. Limited by the centralized architecture, the centralized game service model based on cloud computing cannot be applied to high-reliability and low-latency game applications.

Liao et al. (2016) proposed that the server-side compresses and transmits the graphic stream to the client. The client side renders it to solve the problem of high bandwidth and poor scalability caused by traditional stream-based remote rendering.

At present, many companies have released their cloud gaming platforms. GameStream was developed by NVIDIA Corporation to bring ultra-high-resolution PC games to NVIDIA SHIELD devices (NVIDIA, 2013). This technology leverages the powerful recording capabilities of the GeForce GTX graphics card, screen capture, and encoding acceleration from the hardware level to obtain the streaming data for transmission with extremely low latency, thus ensuring a real-time gaming experience. Steam Link is a cloud gaming solution developed by Valve Corporation. When Steam Link is used in a home local area network environment, any game purchased on the Steam platform can be streamed to a TV or mobile device by connecting to a streaming box sold by Steam (Steam, 2018). StreamMyGame is a pure software cloud game solution first released in 2007. The most significant feature of StreamMyGame is that it does not depend on any hardware facilities. StreamMyGame can stream Windows-based games or applications to PCs with Windows or Linux operating systems.

Edge Computing

Edge computing emerges as localized clouds (Qiu et al., 2022; Sun et al., 2008). It expands the boundaries of cloud computing through distributed computing architecture. In edge computing, data computing, and resource storage are moved from the central node of the network to the edge nodes closer to users or data sources. The transmission and processing speed are significantly improved, thus solving the two bottlenecks of cloud computing bandwidth and delay. Taken together, it has the following advantages:

- **Higher Security:** The data in edge computing is exchanged between the source and edge devices only and is no longer completely uploaded to the cloud computing platform, preventing the risk of data leakage.
- **Low Latency:** According to the ISP's estimation, if the service is processed and forwarded through the multi-access edge computing (MEC) deployed at the access point, the delay is expected to be controlled within 1 ms. If the service is processed and forwarded to the central processing unit of the access network, the wait is about 2~5 ms; even after the MEC processing in the edge data center, the delay time can be controlled within 10 ms. For scenarios with high delay requirements, such as autonomous driving, edge computing is closer to the data source, which can quickly process data and make judgments in real time to fully protect passengers' safety.
- **Reduced Bandwidth Costs:** Edge computing supports local data processing, and local offloading of large-traffic services can reduce backhaul pressure and effectively reduce costs. For example, some connected sensors (such as cameras or aggregated sensors working in the engine) generate data. In these cases, sending all this information to a cloud computing center would take a long time and be prohibitively expensive. Edge computing processing reduces bandwidth costs.

Foveated Rendering

Foveated rendering is a selective rendering technology that dramatically reduces the occupation of computing resources for rendering. In other words, based on ensuring a good experience, foveated rendering technology can reduce the requirements for the computing performance of the helmet. To a certain extent, this technology can reduce hardware costs and accelerate the popularity of cloud games.

Foveated rendering technology divides the picture people see into three areas according to the gazing process of the human eye (the gaze point is clear and nearby) and based on eye tracking technology. Foveated rendering strips and renders the three areas of the image at 100%, 60%, and 20%. The main essence of foveated rendering is to display high resolution in the area where the eyes are easy to focus and display low-key resolution in the peripheral region. Compared with the high-resolution display of the whole screen, it can effectively reduce the calculation of the GPU, thereby reducing power consumption. Foveated rendering relies on eye tracking, and eye tracking needs to be perfect. Otherwise, looking around will be distracting with details. Not all foveated rendering solutions are created equal. The better the eye is tracking, the more efficient the rendering.

At Oculus Connect 5, Facebook showed off progress on FR. Michael Abrash, the company's principal virtual reality (VR) researcher, demonstrated a new method of using machine learning to fill low-resolution regions, allowing for a tiny foveal part. The example Abrash showed was that only 5% of the display resolution is required for a 20x saving. Sony's upcoming PS VR2 will support foveated rendering (based on eye tracking), which can radically improve the picture quality of VR games (VR Gyro, 2019).

Super Resolution

Super-resolution is also known as upsampling; it can efficiently improve the density of pixels, resulting in more detailed features being recovered. These details can play essential roles in specific scenarios. First, the SR is an indeterministic problem. Because there can be multiple corresponding high-resolution candidates for one low-resolution input, there is no single source of truth. As a result, to limit the domain, reliable prior information is essential. Based on the preceding information, several classic SR methods were proposed—for example, the gradient knowledge-based method (Sun et al., 2008), statistics-based methods (Xiong et al., 2010; Wang et al., 2018), flow learning-based method (Wang et al., 2018), and sparse representations-based method (Kim & Quon, 2010). The application of deep neural network-based methods has gained popularity in the SR field, and various deep neural network models and their variants have been proposed recently.

Kappeler et al. (2016) were the first to use the convolution neural network (CNN) in an SR task, and they pointed out that making the adjacent frames align with the vital edge is essential for upsampling. To achieve this aim, their method first computes the optical flow between neighboring structures, combines these distorted frames, and passes them through a CNN model to generate the final SR result. Li et al. (2017) also adopted a two-phrase method based on optical flow motion compensation, and they proposed a new residential CNN model to recover high-frequency details. Another method is a separate optical flow network. Xue et al. (2019) offered a joint training framework to learn the most suitable optical flow feature representation for video training video tasks according to specific task arrangements. The model achieves good results in video interpolation, denoising, and super-resolution. To build a unified framework, Kim et al. (2019) proposed an efficient 3D-CNN video super-resolution method inspired by the time-series capture capability of 3D-CNNs, which does not require motion alignment as a preprocessing step. The network employs residual learning to maximize the capture of temporal nonlinearities between low- and high-resolution frames while maintaining the temporal depth of spatiotemporal feature maps. Instead of direct or slow fusion, Yi et al. (2019) proposed a new progressive fusion network to use spatiotemporal information better. Tian et al. (2020) proposed a temporal deformable alignment network named TDAN that uses deformable convolutions to adaptively align reference and support frames without calculating optical flow. Wang et al. (2019) also adopted deformable and 3D convolutions to handle the video super-segmentation task. Inspired by TDAN, Wang et al. (2019) proposed an EDVR that uses deformable convolution to complete image alignment at the intermediate feature level, effectively avoiding the problem of explicit or implicit optical flow calculation in traditional alignment methods. Li et al. (2019) used a fast spatio-temporal residual network (FSTRN) that improved 3D convolution with a convolution

decomposition strategy and proposed a more lightweight model considering the effect and running speed on video SR.

DESIGN AND IMPLEMENTATION

Architecture Overview

The classic CG pipeline consists of five main steps: game logic execution, visuals capturing, encoding and streaming, decoding and rendering, and user interaction. The first three steps are executed on a remote cloud, and the last two are performed on a thin client on the player side. There are two types of streaming implementation: one is streaming game video data, and the other is streaming rendering instructions. The main problems of streaming rendering instructions are that keeping bandwidth low is hard for the following reasons: Graphics data is complex (e.g., textures, vertex buffers, and index buffers), and streams are uncompressed at a variable rate of up to multiple gigabits per second (Gbps). In addition, command streaming cannot leverage standard compression codecs with decades of research and hardware support, such as H.265; instead, it requires research into new compression algorithms. It is safe to conclude that the market cannot anticipate the joint adoption of command streaming soon. All existing CG platforms use video streaming methods, and our work also focuses on the video streaming scenario.

Recently, Caprolu et al. (2019) and Zhang et al. (2019) proposed frameworks that offload computation-intensive 3D rendering tasks onto GPU-based infrastructures in edge cloud and stream edge-rendered visuals to end users. The center cloud is assigned only lightweight functions, such as user management, state management, and monitoring. Most heavy lifting, including game logic execution, visuals capturing, encoding, and streaming are left to the edge cloud. This design makes theoretical sense, but existing edge servers available on the market are mainly miniaturized versions of center cloud servers, which can hardly achieve consistent performance compared with the center cloud servers. Thus, this disproportionate allocation of computing could be an unbearable burden for the edge servers, causing a performance decline of the entire process.

Based on the above insights, we proposed an architecture that strikes a better balance between computation allocation and bandwidth consumption for both the cloud and the edge sides. To achieve this, we introduced another step for the CG pipeline: video stream enhancement. With the help of this enhancement step, the cloud side can stream a lower-quality video, saving transmission bandwidth and processing time without affecting the player's perception after the video is appropriately enhanced. In our design, core CG tasks, including game logic execution, screen capturing, encoding, and streaming, are still taken by the cloud side, taking full advantage of its creative and stable computation power. Still, instead of streaming directly to the client, the edge server is introduced as a man-in-the-middle component responsible for the video stream enhancement. The architecture is illustrated in Figure 1. The advantages of this architecture are twofold. First, the bandwidth consumption on the backbone network has been significantly reduced because a higher compression rate is allowed. Second, this design is noninvasive and fully compatible with mainstream CG platforms. It can be seamlessly adopted without changing current CG scheduling and security mechanisms.

Prototype Design

To validate the practicability of the architecture shown in Figure 1, we implemented a prototype CG system named FRSR that uses FR and SR together for compression and enhancement. The overall pipeline is illustrated in Figure 2.

The cloud server is mainly responsible for providing the environment for game logic execution and encoding the generated gaming video by using a classic video encoder. Specifically, the original gaming frame is fed to a game-specific ROI prediction model and a downsampler, the ROI model determines which parts of the gaming images are more important to the game player, and the downsampler will

Figure 1.
Overview of edge assistant CG architecture

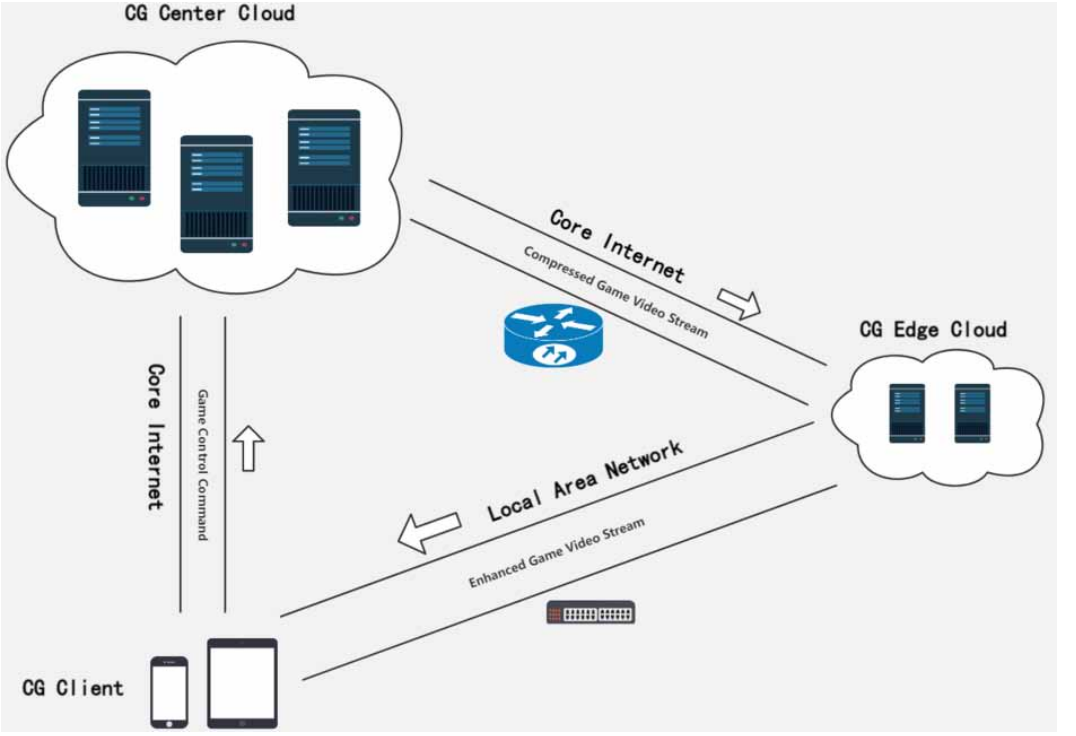
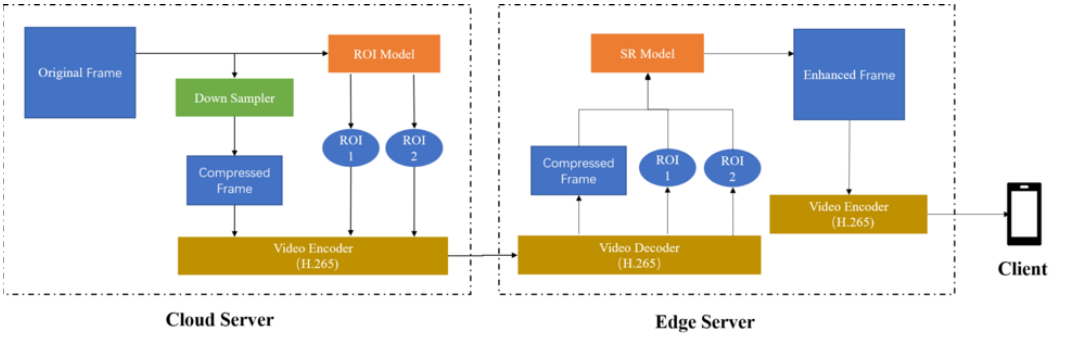


Figure 2.
The architecture of FRSR



downsample the original frame, generating a much smaller output. Then, the compressed frame and the correspondent ROI information are sent to an H.265 encoder to generate a real-time video stream to the edge server. The performance of the ROI prediction model needs to be high enough to ensure the streaming delay is low; this requirement should not be a concern given that the cloud side usually has abundant shared GPU resources that can be dynamically and efficiently scheduled. We discuss the details of the ROI model in the Edge Side Super Resolution section.

The edge server decodes the video stream delivered from the cloud server and upsampling the video content using a game-specific SR model that can efficiently restore and enhance the visuals,

especially for the ROI regions. We discuss how the SR model works in the Training and Implementation section. After the enhancement process, the video data is streamed to the thin client side.

The job for the thin client stays the same as in the classic CG system: It will decode and play the video stream from the edge server to the game player, collect players' controls and stream them back to the cloud server.

Cloud Side Foveated Rendering

Compressing the video content based on ROI is known as foveated rendering (Romero-Rondón et al., 2018; Illahi et al., 2020). Recently several ROI prediction models have been proposed in CG scenarios. Mossad et al. (2021) proposed a video coding architecture named DeepGame. They offered an ROI prediction model that learns from the game player's gaze point and then adaptively predicts the ROI area and its temporal correlation in the frame. Based on the ROI information, different regions of the video frame are encoded with corresponding qualities and control parameters according to their importance.

Our center cloud prototype uses the same model for two reasons: First, its performance can meet real-time streaming requirements according to its report. Second, it supports the prediction of multiple ROIs, which is very important for good gaming experience. We briefly introduce the model architecture, as shown in Figure 3, and refer to Mossad et al. (2021) for detailed information.

The DeepGame ROI prediction network consists of two parts. The first part uses the YOLO model for target detection and recognition in the spatial space, obtains ROI area information with annotation and location information, and then inputs the multiframe content with ROI information into a 2D network. This 2D network has two branches—one input for x-axis information and the other input for y-axis information. This information is used to predict the temporal correlation in the horizontal and vertical directions of the ROI regions, respectively. The second part of the network uses a simple long short-term memory (LSTM) structure. The structure of the LSTM network is shown in Figure 4. The network structure has 128 hidden units with an input size of M or N and the number of objects in the game. We use fully connected layers with ReLU activation functions with sizes 64 and 32. The output is the area the user is most likely to pay attention to and the corresponding confidence. Multiple ROI regions can be obtained in a single video frame through the above steps.

Edge Side Super Resolution

Super resolution was first used for CG scenarios in SRCNN (Dong et al., 2015). The major challenge of using SR in CG is that the delay introduced by the model cannot meet the low latency requirement. As mentioned in Khani et al. (2021), the inference time used by the SR model must be controlled within 20 milliseconds. Thus, only a lightweight model can be chosen. We used the lightweight model proposed in Wang et al. (2021) and offer an innovative adjustment to make it work for multiple ROIs. The experiment proved that the support for multiple ROI could achieve a better quality of experience.

Figure 3.
The architecture of deep game

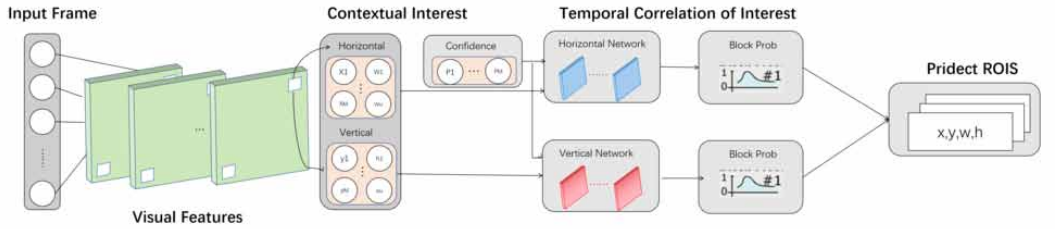
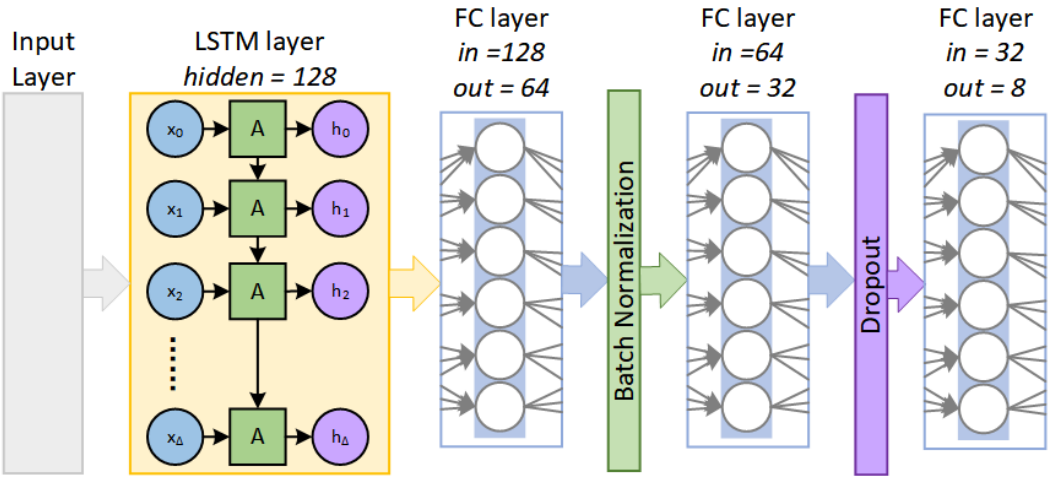


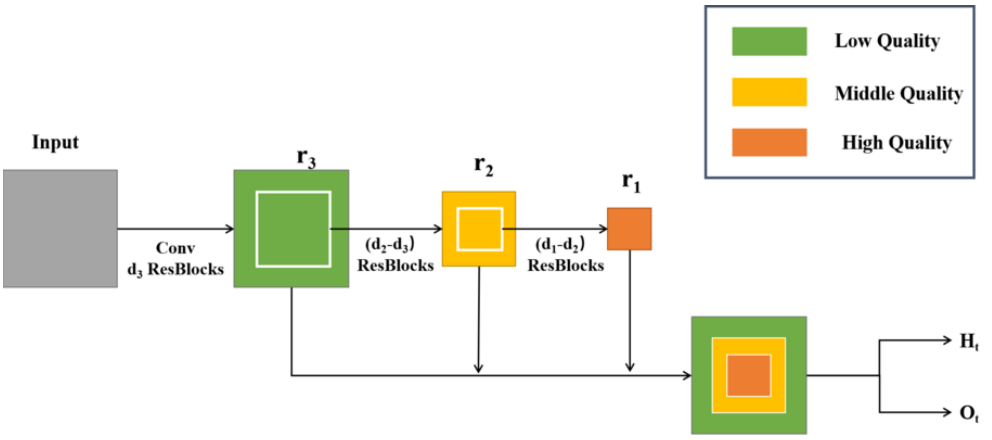
Figure 4.
The architecture of the LSTM



In our edge cloud prototype, we trained a game-specific SR model. No ROI information is used in the SR training phase because this SR model is supposed to learn the general information about upsampling the entire gaming frame. After obtaining the full-featured SR model, we could make the inferring process efficient enough for real-time enhancement. We achieved this by allowing only the ROI regions to go through a deep network to generate high-resolution upsampling and letting most non-ROI areas through a thin network to generate relatively low-resolution output. The model structure is illustrated in Figure 5, where d_1, d_2, and d_3 represent feature depth and r_1, r_2, and r_3 represent region size. However, this model has one limitation: Only one ROI can be supported, which can cause a significant adverse effect on the player's perception.

To solve this problem, we adjusted the FOCAS model to support multiple ROI regions. We used the multi-head mechanism to design the network in the inference stage, letting each head process one ROI area. First, all heads will share the same model depth (feature depth) and region size. In the infer

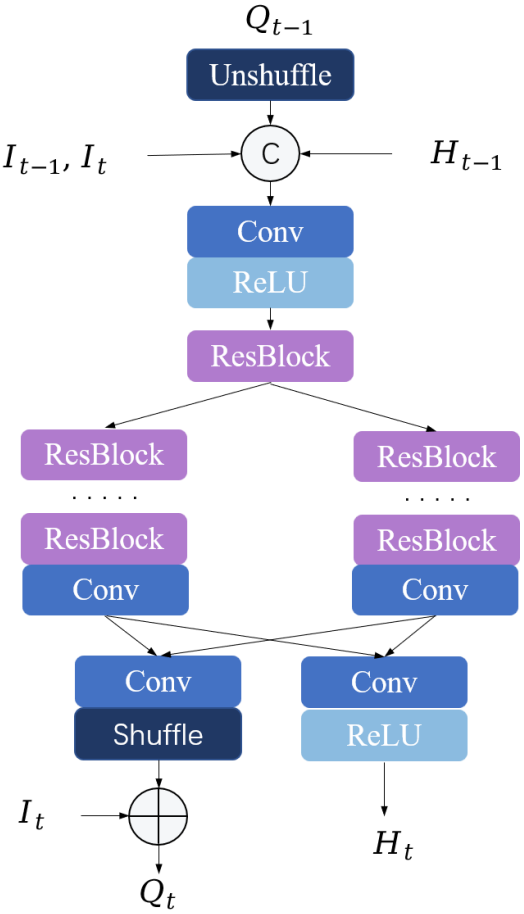
Figure 5.
Overview of FOCAS



stage, the feature input will first pass through the low-level convolution of the SR model to ensure that each head branch learns the standard underlying visual features. At the depth of the first region model, we cropped different sub-features at different spatial positions of the feature map according to the ROI data passed from the cloud side to learn features of higher quality. Each head branch will get the feature map that represents the highest quality. Finally, we averaged the characteristics of all head branches and transformed the features into predicted images through the Pixel Unshuffle layer. Because different departments share parameters and do not involve feature dimension adjustment, our prediction model can dynamically adjust the number of branches according to the number of predicted ROIs in the cloud to achieve the adjustment of SR effects in different network environments. In the experiments, our inference structure is two branches, as shown in Figure 6. Q_{t-1} represents previous SR input. H_{t-1} and H_t represent the previous feature map and the current feature map. I_{t-1}, I_t represent the previous frame and the current frame, respectively.

With the SR model we proposed, the processing pipeline on the edge side is divided into three steps. First, the video stream is decoded by H.265 to obtain a low-resolution image frame and the coordinate information of the ROI in each image frame. Second, both pieces of information pass through our SR model for enhancement, and the ROI area gets better enhancement compared with the non-ROI area. The enhanced frame stream is encoded again in the third step and sent to the client.

Figure 6.
Model architecture of FOCAS for multiple ROIs



Training and Implementation

Cloud Server

Our ROI model shares the same model structure and training process proposed in Illahi et al. (2020). This model has 65.25M parameters. The training dataset has 138,426 game frames from four popular games and the gaze location in each frame. The dataset is available at Illahi et al. (2020). The inference speed reaches more than 40 fps, which meets the needs of real-time inference. In games such as FIFA and CSGO, the ROI model can achieve a prediction accuracy of 80.23%.

After we implemented the game runtime on top of the container, the video of the game was captured and processed in the GPU directly. The ROI model was deployed on the same container to save the transformation between servers. We used H.265 and WebRTC for video encoding and streaming.

We visualized the ROI prediction results of the two games (NBA and CS: GO) during the experiment, as shown in Figure 7. Note that the ROI model can correctly identify the critical areas in the game screen.

Edge Server

Our SR model is trained in the same way as in Dick et al. (2005). This model has 13.5M parameters. The training dataset has 138,426 game frames from four popular games and the gaze location in each frame. The dataset is available at Illahi et al. (2020). In the Recurrent mode, the inference speed of the SR model is 28 fps, while in a non-recurrent way, the inference speed reaches 41 fps, which satisfies the real-time requirement.

We followed FOCAS using the same hyperparameters and training parameters. Specifically, the intermediate feature has 128 channels. The learning rate is 0.0001 and decreases to 1e-5 at epoch 60. We needed to train a total of 70 epochs by using the Adam optimizer. The parameters of the Adam optimizer are beta1 = 0.9, beta2 = 0.999, and weight decay is 5e-4.

EXPERIMENTS

To quantify the bandwidth reduction of our designed architecture while keeping a practical user experience, we built a simulation environment in the laboratory and conducted detailed objective comparative experiments.

Simulation Environments

Our CG testbed was built using two PCs and two laptops setup, as shown in Figure 8. The first PC serve as the center cloud; it has high-end specifications (Intel i7 processor, RTX 3090 GPU). The

Figure 7.
Prediction of ROI

(a)NBA (b)CS:GO

second PC serves as the edge server; it has relative low-end specifications (Intel i5 processor, RTX 3060 GPU). We used one laptop as the client; it has the lowest specification (Intel i3 processor, Integrated GPU). We used another laptop as a network bridge between the cloud server and the edge server. We used NetEm, a network emulator software that allows adding delay, packet loss, and other characteristics to packets outgoing from a selected network interface. To imitate the public Internet connection between the cloud server and edge server as in real life, the edge server and the client are connected through a local area network hub to emulate a local area network.

Evaluation Metrics

We collected and computed PSNR and SSIM and processing time to evaluate objective video quality.

PSNR

PSNR is an engineering term that expresses the ratio of the maximum possible power of a signal to the power of destructive noise that affects the accuracy of its representation. Because many signs have a wide dynamic range, the peak signal-to-noise ratio is expressed in logarithmic decibel units.

To calculate PSNR, we must know the value of the mean squared error (MSE) first. Consider two m×n monochrome images *I* and *K*. If one is a noisy approximation of the other, then their MSE is defined as shown in Equation (1):

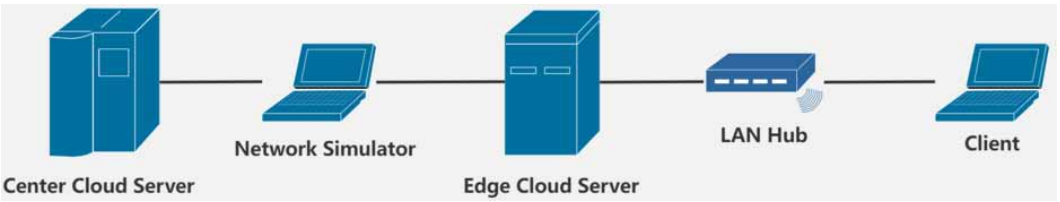
$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I\left(i,j\right)-K\left(i,j\right)\right]^2 \tag{1}$$

When the difference between the real value y and the predicted value f (x) is greater than 1, the error will be amplified. When the difference is less than 1, the error will be reduced, which is determined by the square operation. MSE will give a larger punishment for larger errors (>1), and a smaller punishment for smaller errors (<1). The concept of MSE is well known, which is also a common loss function. PSNR is obtained by MSE, and the formula is as shown in Equation (2):

$$PSNR = 10\times\log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20\times\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{2}$$

MAX_I is the maximum value that represents the color of the image point. If each sampling point is represented by 8 bits, then it is 255. The numerator in the log is the maximum value representing the color of image points. If each sampling point is represented by 8 bits, it is 255. The larger the PSNR, the better the image quality.

Figure 8.
Simulation environment step



SSIM

For the image quality assessment, the effect of local calculation of the SSIM index is better than global. First, the statistical features of the image are usually unevenly distributed in space. Second, the distortion of the image also varies in length. Third, within an average viewing distance, people can focus on only one area of the picture, so the local processing is more in line with the characteristics of the human visual system. Fourth, the local quality detection can obtain the mapping matrix of the spatial quality change of the picture, and the result can be used in other applications, as shown in Equation (3).

$$SSIM\left(x,y\right)=\frac{\left(2\mu_x\mu_y+c\right)\left(\sigma_{xy}+c_2\right)}{\left(\mu_x^2+\mu_y^2+c_1\right)\left(\sigma_x^2+\sigma_y^2+c_2\right)} \tag{3}$$

In Equation (3), μ_x and μ_y represent the average value of x and y, respectively. σ_x and σ_y represent the standard deviation of x and y, respectively. σ_{xy} represents the covariance of x and y. c_1 and c_2 are constants to avoid system errors caused by the 0 denominator.

We measured the PSNR and SSIM metrics in the ROIs because they are the main areas on which players focus.

Processing Time

We used the ROI model in the cloud to predict the ROI coordinate data. The ROI model requires continuous image frame rates as input, while according to Illahi et al. (2020), the game environment can be accurately predicted using the last second of data. Therefore, we do not need to expect ROI on every image frame. We predicted every three structures in the experiment and used the result as the ROI coordinate information of the following three frames. Considering that a single ROI prediction time is 25 million seconds and our model predicts every three structures, our solution fully meets the 30fps real-time video transmission requirements of the CG. We could use a more significant sampling period for games with higher frame rates to ensure that the solution meets real-time requirements.

The time required for cloud encoding and edge decoding is relatively fixed. Precisely encoding takes an average of 5 ms per frame, and decoding takes 2 ms. As for SR's performance, the performance gap between the non-recurrent mode and the recurrent method is small, so we choose the faster non-recurrent SR for SR inference, which takes 24.27 ms.

Objective Study

We selected four common types of games for testing: FIFA, CSGO, NBA, and NHL. We compared five CG implementations to verify the effect of our method. In the first implementation, we adopted the classic cloud game architecture. The game runs in the cloud, and the output video is directly streamed to the client after H.264 encoder. In the second implementation, we upgraded to H.265 encoder. We integrated the SR technology into the cloud game process for the last three performances and compared it with the first two classic implementations. The main difference between these three solutions was on the edge side: On the cloud side, we downsampled the game video's resolution to 480p and then encoded it through an H.265 encoder and streamed it to the edge side. In the third implementation, we directly decoded the low-resolution video stream and upsampled it to 1080P through bicubic interpolation. In the fourth implementation, we used FOCAS for SR. The fifth implementation was to use our SR model optimized for multiple ROIs.

The constant rate actor (CRF) is the default quality (and rate control) setting for the H.264 and H.265 encoders. We could set the CRF values between 0 and 51, where lower values would result in better quality at the expense of larger file sizes. Higher values mean more compression, but at some

point, some quality degradation could be observed. We used CRF to control the BPP value of the compressed output of H.264 and H.265 video streams in different experiments.

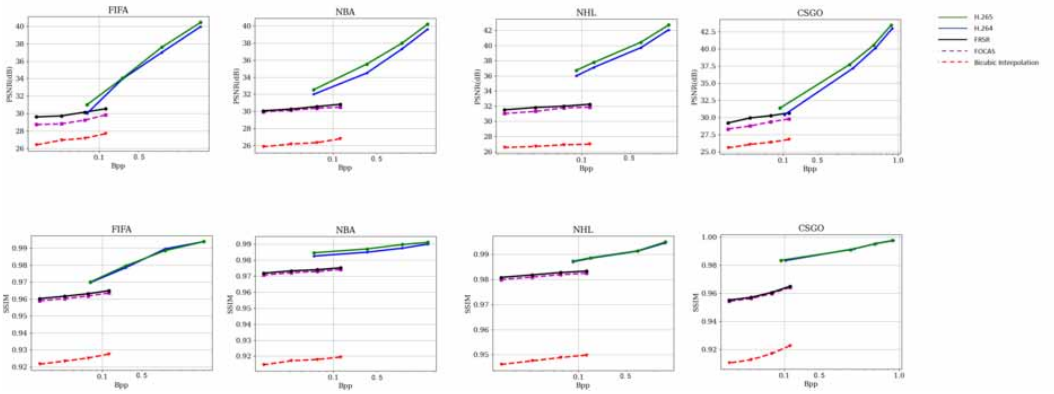
Figure 8 shows the PSNR and SSIM values achieved by different schemes at comparable BPP values. In addition, considering that different game types have additional video attributes, we offer the experimental results on four games, including CSGO, FIFA, NBA, and NHL. Table 1 shows the experimental results of CSGO. Note that under the same BPP value, FRSR and SSIM are higher than LR and FOCAS.

As shown in Figure 9, our method needs only 19.32% of the BPP value to achieve a similar performance to H.265. Specifically, to accomplish the PSNR index of 30dB, our approach needs only 0.051 BPP, while classic CG video transmission requires 0.264 and 0.289 BPP under H.265 and H.264 encoding. Furthermore, our method outperforms all other PSNR and SSIM metrics at similar BPP values. Note that combining ROI prediction and SR into the transmission process of cloud games

Table 1.
PSNR and SSIM values of CSGO game

BPP	Model	PSNR(dB)	SSIM
0.112	LR	26.80	0.923
	FOCAS	29.80	0.964
	FRSR	**30.67**	**0.965**
0.078	LR	26.39	0.917
	FOCAS	29.38	0.960
	FRSR	**30.26**	**0.961**
0.051	LR	26.07	0.913
	FOCAS	28.76	0.956
	FRSR	**29.94**	**0.957**
0.033	LR	25.57	0.910
	FOCAS	28.33	0.954
	FRSR	**29.21**	**0.955**

Figure 9.
Trade-off between video quality and bits-per-pixel for different approaches on four cloud games from datasets in Mossad et al. (2021). (Horizontal coordinate is pixel depth; coordinates are PSNR and SSIM.)



can effectively reduce the use of video streaming bandwidth, which also means that our cloud game system can maintain data transmission in a more complex network environment.

Compared with the three SR schemes in Figure 9, our method is significantly better than FOCAS in PSNR and SSIM under the same BPP value, indicating that the SR prediction of multiple ROI regions can be performed with the same bandwidth. The quality of video transmission can be improved under the circumstance, and the multibranch network structure can be adjusted according to the network situation to adapt to different network changes. The bicubic interpolation scheme has the fastest prediction speed among the three SR schemes, but the transmission quality is the worst. Under different ranges of BPP values, the PSNR index of 30 dB cannot be achieved, so it cannot be applied to the actual CG system.

Figure 9 shows that on four different types of games, our scheme achieves the best PSNR and SSIM metrics among the three side SR methods. Compared with the H.264 and H.265 standard codecs, the BPP required to complete a 30dB PSNR value is significantly reduced. This also reflects the generalization of our solution to meet the video transmission needs of various types of games.

Qualitative Results

We also qualitatively demonstrated the actual effect of real-time SR and restoration of 1080P images for two games, CSGO and FIFA. As seen in Figure 10, our method effectively increases the video data from 480P to 1080P and restores most of the details of the ROI region. Compared with the bicubic interpolation scheme, our super-resolution restoration results are more precise, and the texture, light, and shadow effects are effectively preserved.

In Figure 11, we also show the SR of different regions of the whole picture in the case of multiple ROI regions. The red boxes are two ROI regions, while the green box is a regular region. Note that the SR effect is better in red boxes, effectively improving the quality of image details from 480P to 1080P, while in the green box, the quality of SR is relatively general. There are also some textures in the corners. This also reflects the advantages of our SR at the edge. The traditional SR will improve the quality of the entire image, which will incur high computing costs, high latency, and

Figure 10.
SR using different models

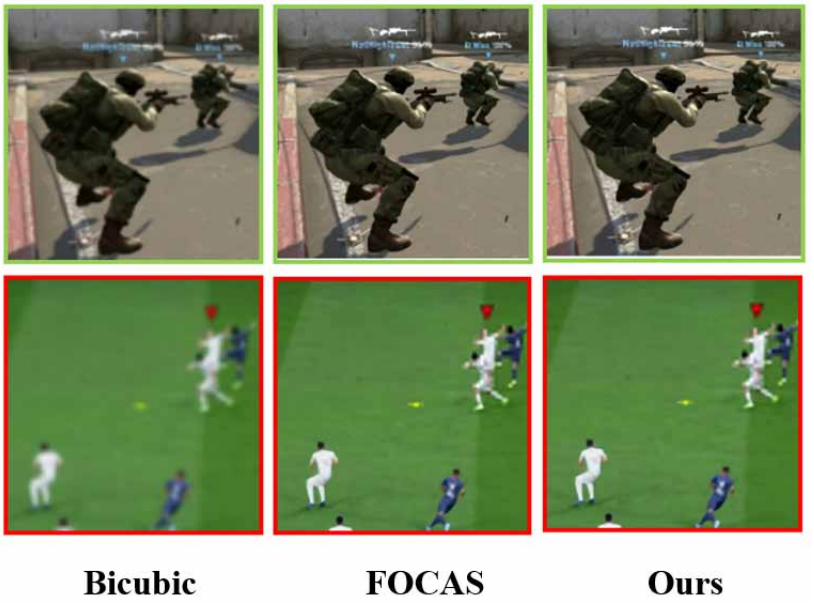
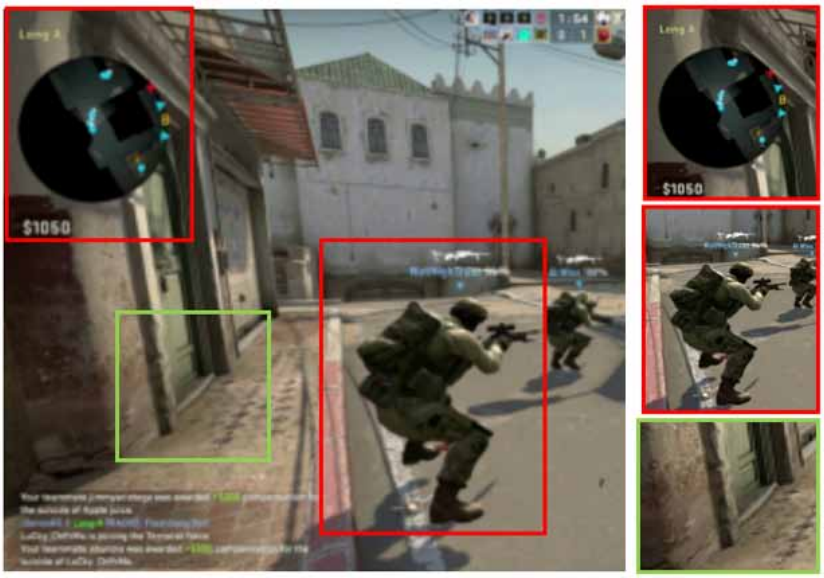


Figure 11.
SR of different regions



low frame rate, all of which are unsuitable for cloud gaming scenarios. Consider that human vision is more sensitive to video quality in attentional regions and less susceptible to peripheral areas. Our scheme better conforms to the attention mechanism. It only performs high-quality super-resolution prediction on the ROI area of each image frame, thereby reducing the computational cost, achieving lower latency and high frame rate, and achieving similar image quality to overall super resolution.

To test the actual effect of the model architecture in real life, we found 10 people to experience the original game and the processed game, and scored, as shown in Table 2. Note in this table that the architecture user experience after using the FRSR model is higher and the score is higher.

To compare the difference in user experience between one ROI region and multiple ROI regions, we invited more than 10 people to experience games with only one ROI region and multiple ROI regions. Table 3 shows that users in multiple ROI regions scored higher.

CONCLUSION

To the best of our knowledge, we are the first to propose cloud edge collaboration architecture that uses foveated rendering and super resolution together. We presented the design and implementation of prototype dubbed as FRSR. Our experiments demonstrated that FR can minimize the CG video volume on the cloud side, saving 39% of bandwidth, which is the most critical factor for the CG

Table 2.
Subjective score of game

Game	FIFA		CSGO		NBA		NHL	
Model	Base	FRSR	Base	FRSR	Base	FRSR	Base	FRSR
Score	2.5	2.5	2.6	3.4	2.2	3.3	2.9	3.7

Table 3.
Subjective evaluation on the number of ROI regions

Game	FIFA			CSGO			NBA			NHL		
Model	One ROI	Two ROIs	Three ROIs	One ROI	Two ROIs	Three ROIs	One ROI	Two ROIs	Three ROIs	One ROI	Two ROIs	Three ROIs
Score	2.0	2.5	2.7	2.1	3.2	3.3	2.5	3.2	3.7	2.7	3.2	3.6

experience. The SR on the edge side can do game-specific enhancement without sacrificing the player's perception, and the entire pipeline can be executed in real time for a qualified CG experience.

In the future, many steps in the CG process can be optimized for better performance and flexibility. To name a few, we want to extend our analysis further to derive mathematical relationships between FR and SR model parameters and quality of service parameters for cloud gaming so that the adaptive encoding scheme can be designed according to the network status. With regard to models, many transformer-based video ROI prediction algorithms have been proposed recently, and they may provide better performance compared with current two-stage RNN based methods. Furthermore, Khani et al. (2021) proposed a lightweight SR model that encodes video into two bitstreams—a content stream and a model stream. This model encodes periodic updates to an SR neural network customized for short segments of the video. We strongly believe this idea of content-adaptive video streaming scheme together with the cloud edge collaborative computing frame have the potential to be applied not only in CG scenarios but also to many other applications where high-quality content needs to be rendered on a thin end. For example, in education field it can enable immersive learning through virtual reality or augmented reality, or provide remote access to medical simulations or training, saving both bandwidth cost and expense on high end hardware. The only assumption we base our idea on is the popularization of artificial intelligence computing power, which is likely to be realized in the foreseeable future.

FUNDING

This research was funded by National Key Research and Development Program of China (No.2022YFC3302103).

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

Amazon Web Services, Inc. (n.d.). *Amazon EC2 On-Demand Pricing*. https://aws.amazon.com/ec2/pricing/on-demand/?nc1=h_ls

Caprolu, M., Di Pietro, R., Lombardi, F., & Raponi, S. (2019, July). Edge computing perspectives: Architectures, technologies, and open security issues. In *Proceedings of the 2019 IEEE International Conference on Edge Computing (EDGE)*. IEEE. doi:10.1109/EDGE.2019.00035

Chang, H., Yeung, D.-Y., & Xiong, Y. (2004, June). Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*. IEEE. doi:10.1109/CVPR.2004.1315043

Dick, M., Wellnitz, O., & Wolf, L. (2005). Analysis of factors affecting players' performance and perception in multiplayer games. In *NetGames '05: Proceedings of 4th ACM SIGCOMM Workshop on Network and System Support for Games*. Association for Computing Machinery. doi:10.1145/1103599.1103624

Dong, C., Loy, C. C., He, K., & Tang, X. (2015). Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295–307. doi:10.1109/TPAMI.2015.2439281 PMID:26761735

Gyro, V. R. (2019). *Why is foveated rendering critical to the second generation of consumer VR headsets?* [Electronic bulletin board online]. https://mp.ofweek.com/vr/a345673221816

Illahi, G. K., Gemert, T. V., Siekkinen, M., Masala, E., Oulasvirta, A., & Ylä-Jääski, A. (2020). Cloud gaming with foveated video encoding. *ACM Transactions on Multimedia Computing Communications and Applications*, *16*(1), 1–24. doi:10.1145/3369110

Kappeler, A., Yoo, S., Dai, Q., & Katsaggelos, A. K. (2016). Video super-resolution with convolutional neural networks. *IEEE Transactions on Computational Imaging*, *2*(2), 109–122. doi:10.1109/TCI.2016.2532323

Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. *Future Generation Computer Systems*, *97*, 219–235. doi:10.1016/j.future.2019.02.050

Khani, M., Sivaraman, V., & Alizadeh, M. (2021). Efficient video compression via content-adaptive super-resolution. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE. doi:10.1109/ICCV48922.2021.00448

Kim, K. I., & Kwon, Y. (2010). Single-image super-resolution using sparse regression and natural image prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(6), 1127–1133. doi:10.1109/TPAMI.2010.25 PMID:20431136

Kim, S. Y., Lim, J., Na, T., & Kim, M. (2019, September). Video super-resolution based on 3D-CNNs with consideration of scene change. In *Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. doi:10.1109/ICIP.2019.8803297

Li, D., & Wang, Z. (2017). Video super resolution via motion compensation and deep residual learning. *IEEE Transactions on Computational Imaging*, *3*(4), 749–762. doi:10.1109/TCI.2017.2671360

Li, S., He, F., Du, B., Zhang, L., Xu, Y., & Tao, D. (2019). Fast spatio-temporal residual network for video super-resolution. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. doi:10.1109/CVPR.2019.01077

Liao, X., Lin, L., Tan, G., Jin, H., Yang, X., Zhang, Y., & Li, B. (2016). LiveRender: A cloud gaming system based on compressed graphics streaming. *IEEE/ACM Transactions on Networking*, *24*(4), 2128–2139. doi:10.1109/TNET.2015.2450254

Mossad, O., Diab, K., Amer, I., & Hefeeda, M. (2021, October). DeepGame: Efficient video encoding for cloud gaming. In *MM '21: Proceedings of the 29th ACM International Conference on Multimedia*. Association for Computing Machinery. doi:10.1145/3474085.3475594

NVIDIA. (2013). *NVIDIA StreamThru* [Electronic bulletin board online]. https://www.nvidia.cn/object/steam_chs.html

OL2, Inc. (2015). *Onlive, Inc.* [Electronic bulletin board online]. https://baike.baidu.com/item/onlive/3965679 /

Qiu, H., Zhu, K., Luong, N. C., Yi, C., Niyato, D., & Kim, D. I. (2022). Applications of auction and mechanism design in edge computing: A survey. *IEEE Transactions on Cognitive Communications and Networking*, *8*(2), 1034–1058. doi:10.1109/TCCN.2022.3147196

Romero-Rondón, M. F., Sassatelli, L., Precioso, F., & Aparicio-Pardo, R. (2018, June). Foveated streaming of virtual reality videos. In *MMSys '18: Proceedings of the 9th ACM Multimedia Systems Conference.* Association for Computing Machinery. doi:10.1145/3204949.3208114

Sun, J., Xu, Z., & Shum, H.-Y. (2008, June). Image super-resolution using gradient profile prior. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. doi:10.1109/CVPR.2008.4587659

Tian, Y., Zhang, Y., Fu, Y., & Xu, C. (2020). TDAN: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE. doi:10.1109/CVPR42600.2020.00342

Valve Corporation. (2018). *Steam Link* [Electronic bulletin board online]. https://store.steampowered.com/app/353380/Steam_Link/?l=tchinese

Wang, H., Su, D., Liu, C., Jin, L., Sun, X., & Peng, X. (2019). Deformable non-local network for video super-resolution. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 177734–177744. doi:10.1109/ACCESS.2019.2958030

Wang, L., Guo, Y., Lin, Z., Deng, X., & An, W. (2018, December). Learning for video super-resolution through HR optical flow estimation. In C. V. Jawahar, H. Li, G. Mori, & K. Schindler (Eds.), Computer vision – ACCV 2018. Springer. doi:10.1007/978-3-030-20887-5_32

Wang, L., Hajiesmaili, M., & Sitaraman, R. K. (2021, October). Focas: Practical video super resolution using foveated rendering. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21).* Association for Computing Machinery. doi:10.1145/3474085.3475673

Wang, X., Chan, K. C., Yu, K., Dong, C., & Change Loy, C. (2019). EDVR: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE. doi:10.1109/CVPRW.2019.00247

Xiong, Z., Sun, X., & Wu, F. (2010). Robust web image/video super-resolution. *IEEE Transactions on Image Processing*, *19*(8), 2017–2028. doi:10.1109/TIP.2010.2045707 PMID:20236889

Xue, T., Chen, B., Wu, J., Wei, D., & Freeman, W. T. (2019). Video enhancement with task-oriented flow. *International Journal of Computer Vision*, *127*(8), 1106–1125. doi:10.1007/s11263-018-01144-2

Yi, P., Wang, Z., Jiang, K., Jiang, J., & Ma, J. (2019). Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE. doi:10.1109/ICCV.2019.00320

Zhang, X., Chen, H., Zhao, Y., Ma, Z., Xu, Y., Huang, H., Yin, H., & Wu, D. O. (2019). Improving cloud gaming experience through mobile edge computing. *IEEE Wireless Communications*, *26*(4), 178–183. doi:10.1109/MWC.2019.1800440

Xinkun Tang received a B.S. degree in electronic information engineering in 2011 from Shandong Jianzhu University, Jinan, China, and an M.S. degree in the field of electromagnetic and microblogging technology in 2014 from Communication University of China, Beijing, China. He is currently working toward a Ph.D. in computer networking and security at the Communication University of China, Beijing, China. His research interests include recommended algorithm and video super resolution.

Ying Xu received a B.S. degree in communication in 2018 at the University of China, Beijing, China, and an M.S. degree in the field of electromagnetic and microblogging technology in 2021 from Communication University of China, Beijing, China. She is currently working at the Academy of Broadcasting Science, Beijing, China. Her research interests include recommended algorithm and video super resolution.

Feng Ouyang is a professor and deputy director of Cable Network Institute. Ouyang's focus is on the research of network information technology.

Ligu Zhu is a doctor of computer science and teaches at the Communication University. His research direction is big data and artificial intelligence. He has participated in numerous national major scientific research projects, published more than 20 papers in domestic and foreign academic conferences and domestic core journals, and reached some research achievements at the leading level in China.

Bo Peng is a guest professor in the Computer Science Department of the Communication University of China. Peng's research interests focus on artificial intelligence for multimedia, human-computer interaction, and virtual reality.