applied sciences

MDPI

Article

Cloud Gaming Video Coding Optimization Based on Camera Motion-Guided Reference Frame Enhancement

Yifan Wang [1], Hao Wang [1], Kaijie Wang [1] and Wei Zhang [1,2,*]

[1] School of Telecommunications Engineering, Xidian University, Xi'an 710071, China
[2] Artificial Intelligence Research Center, Peng Cheng Laboratory, Shenzhen 518055, China
* Correspondence: wzhang@xidian.edu.cn

Abstract: Recent years have witnessed tremendous advances in clouding gaming. To alleviate the bandwidth pressure due to transmissions of high-quality cloud gaming videos, this paper optimized existing video codecs with deep learning networks to reduce the bitrate consumption of cloud gaming videos. Specifically, a camera motion-guided network, i.e., CMGNet, was proposed for the reference frame enhancement, leveraging the camera motion information of cloud gaming videos and the reconstructed frames in the reference frame list. The obtained high-quality reference frame was then added to the reference frame list to improve the compression efficiency. The decoder side performs the same operation to generate the reconstructed frames using the updated reference frame list. In the CMGNet, camera motions were used as guidance to estimate the frame motion and weight masks to achieve more accurate frame alignment and fusion, respectively. As a result, the quality of the reference frame was significantly enhanced and thus being more suitable as a prediction candidate for the target frame. Experimental results demonstrate the effectiveness of the proposed algorithm, which achieves 4.91% BD-rate reduction on average. Moreover, a cloud gaming video dataset with camera motion data was made available to promote research on game video compression.

Keywords: video coding optimization; inter prediction; cloud gaming videos; frame enhancement; deep learning

check for updates

Citation: Wang, Y.; Wang, H.; Wang, K.; Zhang, W. Cloud Gaming Video Coding Optimization Based on Camera Motion-Guided Reference Frame Enhancement. *Appl. Sci.* **2022**, *12*, 8504. https://doi.org/10.3390/app12178504

Academic Editor: Hui Yuan

Received: 22 July 2022
Accepted: 22 August 2022
Published: 25 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

As an important information transmission medium, videos are widely used in entertainment, security monitoring, online meetings, virtual reality, and other fields. Although the rapid development of video services meets the requirements of users, the huge data volume of raw videos can hardly be directly transmitted and stored. Especially in recent years, the progress of cloud computing technology has promoted the revitalization of the cloud gaming industry [1], and the transmission of various high-definition cloud gaming videos required by cloud gaming systems has greatly increased the demand for bandwidth. Compared to natural videos, game videos contain complicated animations and visual effects, thus requiring larger bandwidth when transmitted [2]. For instance, the most popular commercial cloud gaming system, i.e., OnLive, needs at least 2 Mbps bandwidth [3]. A large amount of transmitted data and high bandwidth occupation have put great limitations on the development of cloud gaming. To this end, how to effectively compress cloud gaming videos is crucial.

Video compression technology has developed vigorously since the 1980s, and can remove the redundancy of videos while maintaining quality. As one of the core modules in video coding, inter-prediction uses the reconstructed blocks in reference frames to predict the coding blocks in the current frame. Compared to directly encoding and transmitting the raw frames, processing the predicted residuals can significantly reduce the transmitted data volume. In particular, when the reference frame is of high quality and highly related to the current frame, it can further improve the accuracy of motion estimation and motion compensation, therefore greatly reducing the bits for coding residuals. Therefore, reference frames play an important role in inter-frame prediction. In traditional video codecs, the reconstructed previous frame with the highest relation to the current frame and reconstructed frames with the highest quality in each previous Group of Pictures (GOP) are generally selected as reference frames under the Low-Delay-P (LDP) configuration. Moreover, with the improvement of computing power, deep learning neural networks have been gradually applied to video compression. Either enhancing the quality of reference frames or generating additional reference frames through deep learning networks [4–6] has shown excellent performance far beyond traditional algorithms. To put it differently, optimizing the frames in the reference frame list through a deep learning network can improve the coding efficiency to a certain extent.

However, most of the networks mentioned above are applicable to natural videos captured by cameras, while we aim at improving the coding efficiency of game videos in cloud gaming. Meanwhile, the commonly used video coding standard H.265/HEVC mainly works for simple translation movements [7] while cloud gaming videos, especially the first angle games, contain more complicated rotation movements. We can see from Figure 1 that H.265/HEVC has different compression performances under various camera motions in the same scene. Videos with camera rotation often require higher transmission bits. Considering that the core of cloud gaming is to render 3D video games on the cloud server and send game scenes as 2D videos to game players through broadband networks, the video encoder runs together with the 3D game engine so that we can obtain the camera motions in graphics rendering contexts directly. The movement between frames can be further described in detail and intuitively using the frame-by-frame camera motion information. This prior information can help obtain more accurate pixel motion vectors in various complex movements, therefore reducing the encoding residuals. In this case, we propose to leverage the camera motion information unique to cloud gaming videos to guide the optimization of reference frames with deep learning techniques. By introducing additional high-matching and high-quality reference frames, the temporal redundancy is greatly reduced, resulting in a significant reduction in bits for coding residuals, so that the compression efficiency of cloud gaming videos with a large number of rotation movements can also be improved.

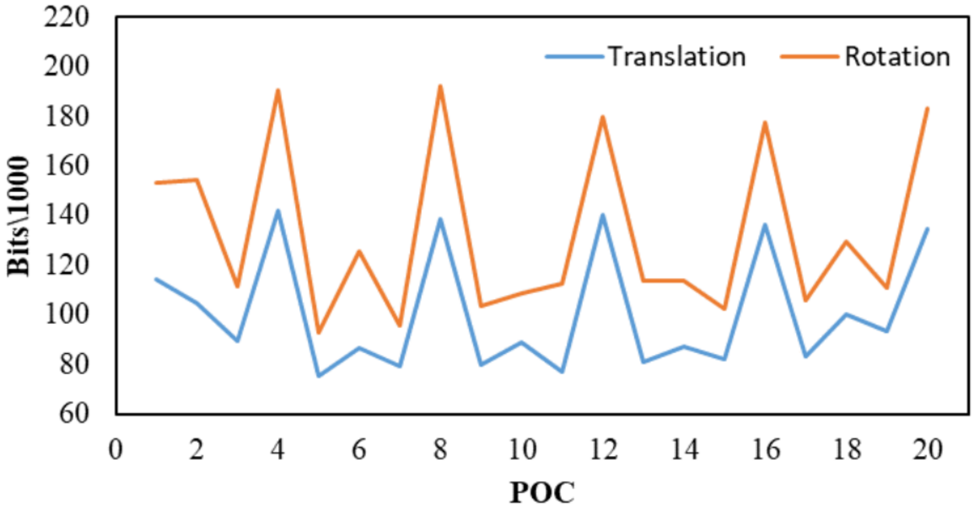


Figure 1. Relationship between camera motions and transmission bits in H.265/HEVC.

In this paper, we present how we take advantage of the camera motions to optimize video encoding for cloud gaming videos. Overall, using the camera motion information and the known reconstructed frames in the reference frame list, we generate a high-quality enhanced reference frame with high relation to the current frame through a deep learning network. Then the enhanced frame is added to the reference frame list to participate in the coding process. The operation at the decoder side is consistent with the encoder side, which recovers the updated reference frame list using the transmitted camera motion information and decodes the frames based on it (That is to say, the proposed scheme is not compatible with the H.265 standard). Specifically, to generate the enhanced reference frame, we propose a camera motion-guided network referred to as the CMGNet, which consists of three parts: the camera motion-guided prediction module, the alignment and fusion module, and the reconstruction module. We first estimate more accurate pixel-level motion information and fusion masks through the camera motion-guided prediction module, then align input frames to the target frame and merge them to acquire the fused feature. Finally, the reconstruction module is used to generate the corresponding high-quality reference frame.

The main contributions of this paper can be summarized as follows:

- We propose a coding optimization algorithm for cloud gaming videos applying deep learning techniques to optimize traditional video codecs, which generates an enhanced reference frame with high relation to the frame to be encoded and adds it to the reference frame list for better compression;
- Our proposed CMGNet for generating enhanced reference frames uses camera motions as guidance to estimate more accurate pixel-level motions for frame alignment, significantly enhancing the quality of the reference frame;
- We have established a game video dataset containing sufficient rendering frames and camera motions to promote research on game video compression;
- Experimental results demonstrated the effectiveness of the proposed coding optimization algorithm for cloud gaming videos.

The rest of the paper is organized as follows. Related work is summarized in Section 2. The proposed coding optimization algorithm for cloud gaming videos together with the proposed CMGNet for the generation of enhanced reference frame is detailed in Section 3. Experimental results are shown in Section 4 to verify the expected improvements. Finally, a brief summary of this paper is given in Section 5.

2. Related Work

2.1. Video Compression

In recent years, various video applications have emerged with the development of network and storage technologies. The diversification and high-definition trend of video applications have put forward higher requirements for video compression. The formulation of international video coding standards has promoted the video compression process. In particular, H.265/HEVC [7] standardized by ITU-T Video Coding Expert Group (VCEG) and ISO/IEC Moving Picture Expert Group (MPEG) is widely used due to its 50% bitrate reduction over its predecessor H.264/AVC [8] while maintaining the same level of video quality. Some researchers proposed algorithms to achieve computational complexity scalability for HEVC encoders [9–12] and others optimized the loop filters [12–14], as well as the intra-frame/inter-frame prediction process [15,16]. Some works use the features of the infrared videos [17], satellite videos [18] and surveillance videos [19,20] to generate additional reference frames for H.265/HEVC for further bitrate savings. Many works also applied deep learning techniques to video compression, including using deep learning networks to improve the accuracy of sub-pixel motion estimation and motion compensation [21–24], enhance the bi-prediction performance [25], and improve the quality of reference frames [4–6], etc. Considering the continuous growth of video services and the development of new industries such as cloud gaming, the demand for bandwidth is increasing exponentially and the exploration of video compression is still urgent.

2.2. Video Quality Enhancement

The purpose of video quality enhancement is to reduce the distortion of compressed videos, and the methods based on deep learning have achieved remarkable results. The early ARCNN [26] and IFCNN [27] were inspired by the NN-based image super-resolution algorithm and improved based on the SRCNN [28] structure. VRCNN [29] introduces CNN in the loop filter of HEVC intra-coding and obtains improved coding performance. With the continuous development of deep learning, more complicated structures are applied to enhancement networks. For instance, RHCNN [30] introduces residual highway units to improve the quality of reconstructed frames. ACRN [31] uses asymmetric convolution and dense structure to better extract features. RRCNN [32] adopts the recursive residual module to achieve parameter sharing, obtaining excellent performance with fewer parameters, etc. Some other works apply coding information to the network, such as the partitioning map of the Coding Unit (CU) and Transform Unit (TU) [33,34], coding residual [35], and Quantization Parameter (QP) [36] to help the network converge quickly and further improve the quality of reconstructed frames.

However, the networks mentioned above only consider the frame to be enhanced, ignoring the temporal correlation between frames in the video sequence. To further take advantage of the inter-frame relationship, methods for multi-frame quality enhancement have emerged. QENet [37], LMVE [38], and MGANet [39] use the optical flow estimated by the optical flow network [40,41] to align input frames to the current frame and feed them into the network for fusion and enhancement. MFQE [42] finds two peak quality frames to enhance the adjacent frame. MFQE 2.0 [43] further extends MFQE and significantly improves it. Moreover, STDF [44] and RFDA [45] learn a novel Spatio-Temporal Deformable Convolution (STDC) to aggregate temporal information while performing frame fusion. There are also works combining optical flow with deformable convolution for better alignment [46,47]. Either using a single frame or multiple frames for video quality enhancement has shown excellent performance.

3. The Proposed Coding Optimization Algorithm and Network Architecture

3.1. Algorithm Overview

We propose a coding optimization algorithm for cloud gaming videos with reference frame enhancement through a deep learning network. Figure 2 shows an overview of the proposed optimization compression pipeline. The blue boxes represent the conventional codec (i.e., H.265), which is applied to compress the cloud gaming videos. The cyan blocks indicate the proposed CMGNet, which is responsible for generating the enhanced reference frame. There are at most four reconstructed frames in the reference frame list under the LDP configuration. We use the four reference frames and the camera motions to generate an enhanced reference frame through the CMGNet and then add it to the reference frame list to participate in the inter-prediction process of the current frame. It is noted that there are less than four reference frames in the reference frame list of the first three predictive frames, so we copy the last reference frame with the smallest Picture Order Count (POC) in the list to make up the four reference frames. The camera motion information is signaled along with the conventional codec's bitstream. On the decoder side, the operation is consistent with the encoder side, and it is not necessary to transmit additional labels to mark the generated reference frames.

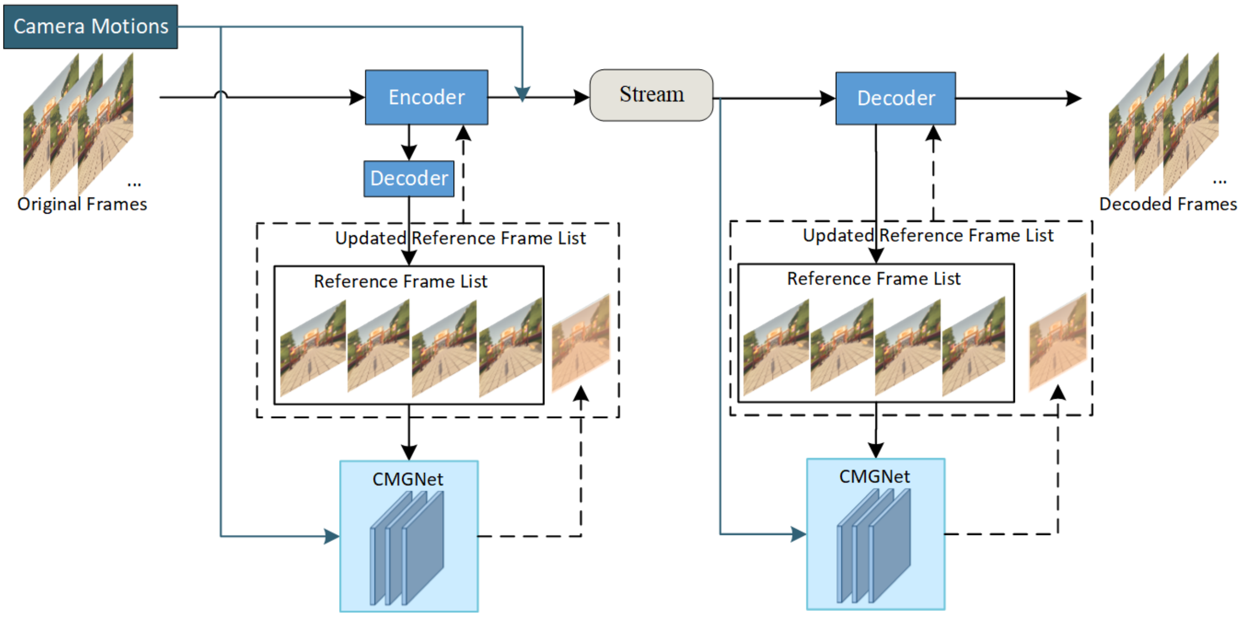


Figure 2. Overview of the proposed optimization compression pipeline.

3.2. Network Architecture

We assume that the reference frames in the reference frame list are denoted as I_{R_1}, I_{R_2}, I_{R_3}, and I_{R_4}, which are arranged in order of the POC (from the smallest to the largest). Considering that the reconstructed frame of the previous frame and those with the highest quality in each previous GOP are generally selected as reference frames under the LDP configuration, I_{R_4} has the highest temporal correlation and similarity with the frame to be encoded while I_{R_1}, I_{R_2}, and I_{R_3} have a higher objective quality. We take the high-quality reference frames I_{R_1}, I_{R_2}, and I_{R_3} to enhance the quality of I_{R_4}, using the temporal correlation between them to generate a new reference frame denoted as $I^E_{R_4}$, with both high-quality and high-temporal correlation. Thus, the proposed model can be expressed as:

$$I^E_{R_4} = f_\theta\left(\left\{I_{R_i}\right\}^4_1, \left\{V_{R_iR_4}\right\}^4_1\right) \tag{1}$$

where $V_{R_iR_4}, i \in \{1,2,3,4\}$ represents the camera motion indicating the transformation from the position of I_{R_i} to I_{R_4}. $I^E_{R_4}$ is the enhanced target frame (i.e., the output of the CMGNet). I_{R_i} and $V_{R_iR_4}$ are the inputs of the proposed CMGNet and θ represents the set of the learnable model parameters. The structure of the proposed CMGNet is shown in Figure 3, which consists of a camera motion-guided prediction module, an alignment and fusion module, and a reconstruction module.

As shown in Figure 3, I_{R_i}, I_{R_4}, and $V_{R_iR_4}$ are first fed into the camera motion-guided prediction (i.e., CMGP in Figure 2) block to predict the pixel offset p_{R_i} for the alignment of the input frame to the target frame, and the weight mask m_{R_i} for the frame fusion. Then, four pairs of offsets and masks, i.e., (p_{R_i}, m_{R_i}), together with I_{R_i}, for each $i \in \{1,2,3,4\}$, are fed into the alignment and fusion module to obtain the fused feature. We use the deformable convolution (DCNv2) [48] to perform frame alignment and fusion, which has been proved effective. Finally, the fused feature is input to the reconstruction module to fully explore the information contained in the feature map, therefore improving the quality of the reference frame. In our implementation, we select a simple yet effective eight-layer convolution with residual learning [49] as the reconstruction module due to its versatility and efficiency (eight layers for the luma component and four layers for the chroma component). Thus, the model can be re-formulated as:

$$I^E_{R_4} = f_{\text{Rect}}\left(f_{DCN}\left(f_{CMGP}\left(\left\{I_{R_i}, V_{R_iR_4}\right\}^4_1, I_{R_4}\right), \left\{I_{R_i}\right\}^4_1\right), I_{R_4}\right) \tag{2}$$

where $f_{CMGP}(\cdot)$, $f_{DCN}(\cdot)$, and $f_{\text{Rect}}(\cdot)$ represent the CMGP module, the alignment and fusion module, and the reconstruction module, respectively. The outputs of $f_{CMGP}(\cdot)$ are offset p_{R_i} and mask m_{R_i} required by DCNv2. We will describe the CMGP in detail below.

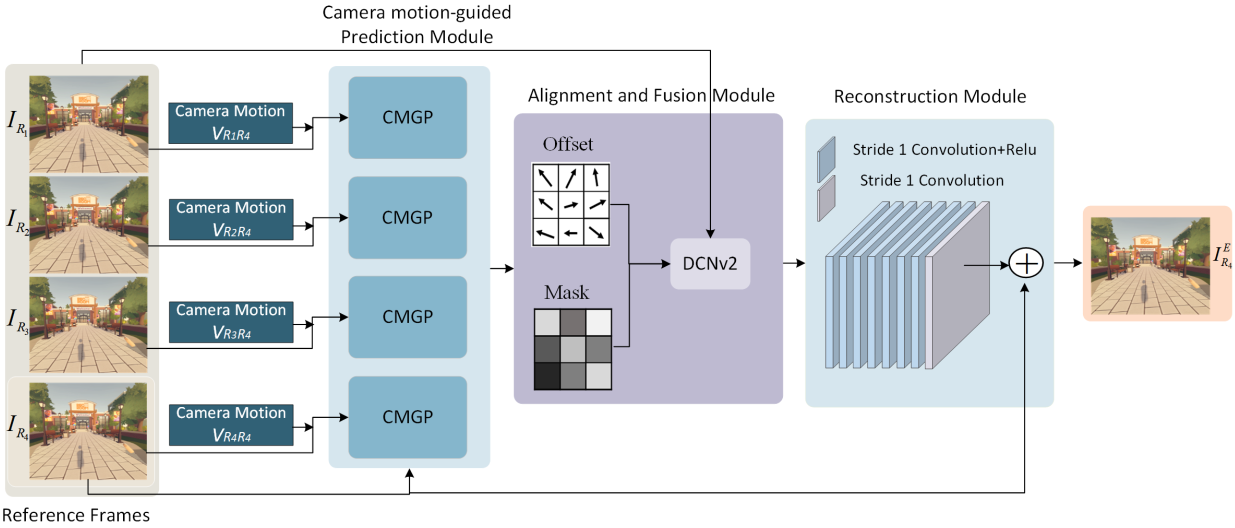


Figure 3. The proposed camera motion-guided network CMGNet.

3.2.1. Camera Motion-Guided Prediction (CMGP)

It is well-known that effectively using the temporal correlation between frames for frame alignment and fusion is a key issue. The non-consecutive inputs with varying degrees of motion offset further increase the difficulty for frame alignment and fusion. Deformable convolution has shown advanced performance in frame alignment and fusion, so we use it in our network. In deformable convolution operations, obtaining accurate alignment offsets and fusion masks is crucial. Inspired by the excellent performance of flow-guided deformable alignment [47], we adopt the flow-based method to predict the basic offset in our paper. Meanwhile, benefiting from the rendering and coding mechanism of the cloud gaming system, the camera motions that accurately record displacements between frames can be directly obtained. To some extent, the camera motion between the input frame and the target frame reflects the corresponding offset between them, and input frames with different camera motions relative to the target frame could bring different degrees of reference and correlation. Therefore, we use this information to guide the offset and mask prediction process. Figure 4 gives the illustration of the proposed CMGP. One CMGP block is selected as an example in the following discussion for simplification, and the operation is identical for all blocks.

Given a pair of reconstructed reference frames I_{R_i} and I_{R_4}, we first use a pre-trained flow prediction network to calculate the optical flow $F_{R_i \to R_4}$. The well-known pyramid structure SPyNet [50] obtains final optical flow by generating flows at different scales and gradually refining them. Its smaller model parameters with advantages in terms of accuracy and speed attract us to adopt it. Then the warped frame $I^W_{R_i \to R_4}$ is obtained by warping I_{R_i} to I_{R_4} through the optical flow $F_{R_i \to R_4}$. The above process can be expressed by the following formulas:

$$\begin{aligned} F_{R_i \to R_4} &= SPyNet(I_{R_i}, I_{R_4}) \\ I^W_{R_i \to R_4} &= warp(I_{R_i}, F_{R_i \to R_4}) \end{aligned} \tag{3}$$

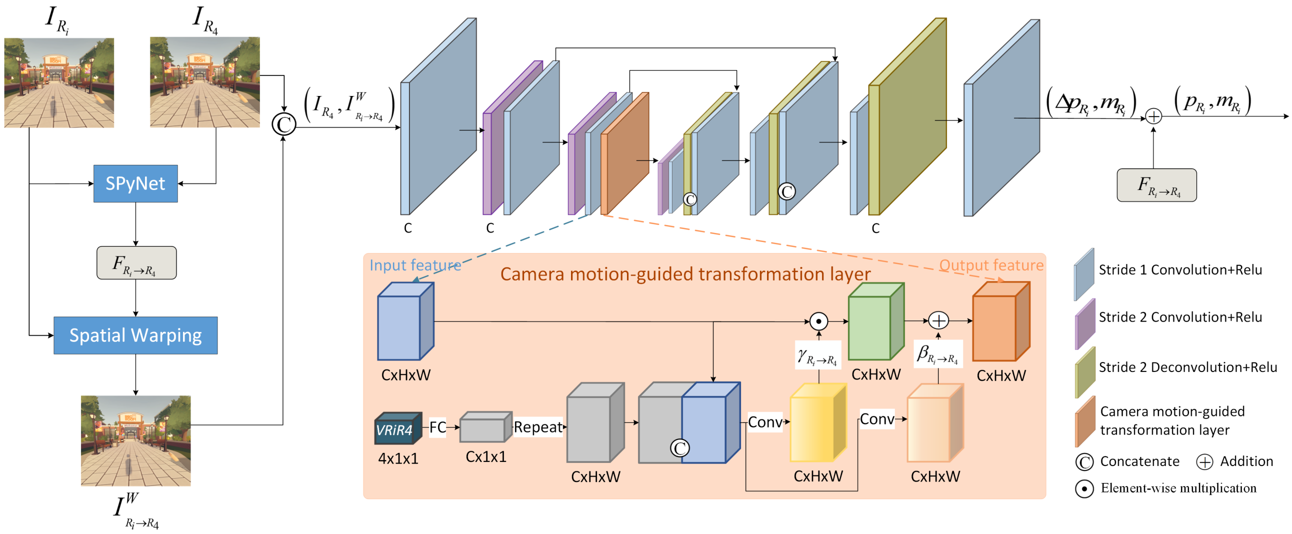


Figure 4. Illustration of the CMGP block.

We regard the optical flow as the basic offset. Then we use the warped frame $I^W_{R_i\to R_4}$ and the target frame I_{R_4} to predict the residual offset Δp_{R_i} and fusion mask m_{R_i}. As shown in Figure 4, we use an encoder-decoder architecture to predict them, where convolution and deconvolution layers with a stride of 2 are used for downsampling and upsampling, respectively. In the convolution layers with a stride of 1, we use zero padding to keep the resolution of features unchanged. There is also a camera motion-guided transformation layer inspired by [51] before the last downsampling layer. The camera motion information $V_{R_iR_4}$ participates in the prediction process by guiding the production of the affine transformation parameters for the downsampling feature in this layer. Such projection then guides the feature adjustment based on the accurate prior information, leading to more precise residual offset and mask prediction. Since we focus on the rotation movement of cloud gaming videos in this paper, the camera motion $V_{R_iR_4}$ can be represented by a quaternion in the shape of $4\times1\times1$. Then in the camera motion-guided transformation layer, we use a fully connected layer and a repeat operation to make $V_{R_iR_4}$ the same dimension as that of the input feature f_{in}. The transformation parameters $\gamma_{R_i\to R_4}$ and $\beta_{R_i\to R_4}$ are produced by a mapping function:

$$(\gamma_{R_i\to R_4}, \beta_{R_i\to R_4}) = \Psi([f_{in}, V_{R_iR_4}]) \tag{4}$$

where the camera motion and input feature are concatenated together for guidance. After that, the translation is carried out by scaling and shifting the input feature:

$$f_{out} = \gamma_{R_i\to R_4} \odot f_{in} + \beta_{R_i\to R_4} \tag{5}$$

where f_{out} represents the output feature after the camera motion-guided transformation layer, whose dimension is the same as $\gamma_{R_i\to R_4}$ and $\beta_{R_i\to R_4}$, and $\odot$ is referred to element-wise multiplication. Afterward, the residual offset and fusion mask can be obtained through several convolution layers. The final offset can thus be represented as:

$$p_{R_i} = F_{R_i\to R_4} + \Delta p_{R_i} \tag{6}$$

Considering that the general video sampling format of H.265/HEVC is YUV4:2:0, and the luma component contains more detail and high-frequency information while the chroma component is relatively flat, we process the luma and chroma separately in the paper and set the network structure of the chroma component as a simplified version of that of the luma component. Therefore, in a CMGP block, the luma component performs downsampling and upsampling three times while the chroma component performs two times. The optical flow of the chroma component is also simply set as a twice performed downsampling version of luma. It should be noted that we set $F_{R_4 \to R_4}$ to zero and do not perform the spatial warping operation when $i = 4$. That is to say, we use two I_{R_4} frames to predict the residual offset Δp_{R_4} and fusion mask m_{R_4}, and the final offset is directly represented by Δp_{R_4} in this case.

3.2.2. Loss Function

We adopt the commonly used Mean Absolute Error (MAE) to supervise the proposed network. The MAE can be represented as:

$$\begin{aligned} MAE\left(I_{R_4}, I_{R_4}^E\right) &= \frac{\left\| I_{R_4} - I_{R_4}^E \right\|_1}{W \times H + \frac{W}{2} \times \frac{H}{2} + \frac{W}{2} \times \frac{H}{2}} \\ &= \frac{\left\| Y_{R_4} - Y_{R_4}^E \right\|_1 + \left\| U_{R_4} - U_{R_4}^E \right\|_1 + \left\| V_{R_4} - V_{R_4}^E \right\|_1}{\frac{3}{2} WH} \\ &= \frac{4}{6} MAE\left(Y_{R_4}, Y_{R_4}^E\right) + \frac{1}{6} MAE\left(U_{R_4}, U_{R_4}^E\right) + \frac{1}{6} MAE\left(V_{R_4}, V_{R_4}^E\right) \end{aligned} \tag{7}$$

Furthermore, the loss function can be described accordingly as:

$$loss = 4MAE\left(Y_{R_4}, Y_{R_4}^E\right) + MAE\left(U_{R_4}, U_{R_4}^E\right) + MAE\left(V_{R_4}, V_{R_4}^E\right) \tag{8}$$

where $Y_{R_4}^E$, $U_{R_4}^E$, and $V_{R_4}^E$ are the components of the enhanced frame, Y_{R_4}, U_{R_4}, and V_{R_4} are those of the ground truth. Weighting in the ratio of 4:1:1 can better reflect the importance of each component.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets

Most current datasets for deep learning only contain camera–captured natural images while we aim at cloud gaming videos with corresponding camera motions. Herein, we use the Air Light VR (ALVR) [52], a piece of software to stream SteamVR games to the standalone VR headsets of users, to establish a cloud gaming video dataset containing the rendering images and rotating camera motions. The camera motions are represented by quaternions. We select five representative popular games [53–57] containing various scenes as the game sequence library, and the raw videos are selected across diverse types of content. Ten videos with 1000 frames are employed for training, and another three with 500 frames called Secret Shop, Robot Repair, and VRChat, are used for testing. The videos are cropped to a uniform size of 1024 × 1024. Figure 5 shows several frames of the test set as examples.

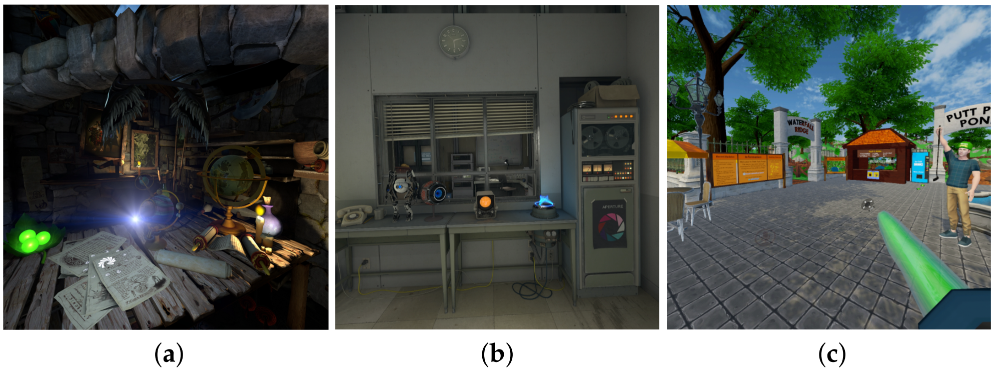


Figure 5. Examples of the test set. (**a**) Secret Shop. (**b**) Robot Repair. (**c**) VRChat.

4.1.2. Implementation Details

All videos are encoded by HM 16.22 under the LDP configuration at four different quantization parameters (QP) (i.e., 22, 27, 32, 37) on an Intel (R) Core (TM) i7-9700K CPU @ 3.60 GHz processor. We train four models from scratch for the four QPs independently. The Adam optimizer [58] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$ is adopted for the training of the proposed CMGNet. We set the batch size to 32 and randomly crop the input frame into patches in 128 × 128. The learning rate is set to 10^{-4} thorough out the training process. We train the luma component and chroma component separately. In the CMGP block, there are three downsampling and upsampling layers with 32 channels followed by ReLU activation for the luma component, while there are two for the chroma component. The number of output channels is equal to 64. We select an eight-layer convolution with residual learning in the reconstruction module for the luma component and that with four layers for the chroma component. All layers have 48 convolution filters followed by ReLU activation. The generated frame is added at the first position of the original reference frame list, and original reference frames are moved backward in sequence. For the evaluation of the performance, we adopt the Bjontegaard Deltas [59] for the rate (i.e., BD-rate,) and the Bjontegaard Deltas for the quality measure (i.e., BD-PSNR) in the YCbCr space to evaluate the proposed coding optimization algorithm, and use the Peak Signal-to-Noise Ratio (PSNR) in the YCbCr space and Structural Similarity Index (SSIM) in the RGB space to evaluate the proposed CMGNet.

4.2. Experimental Results

4.2.1. Compression Performance of the Coding Optimization Algorithm

We first evaluate the rate-distortion performance of our proposed algorithm to validate its efficiency. We take the commonly used H.265 as the anchor and compare our proposed algorithm with it in terms of BD-rate and BD-PSNR under the LDP configuration. Negative BD-rate values and positive BD-PSNR values indicate better performance. It should be noted that the proposed algorithm needs transmit the camera motion information to the decoder side, whereas the additional overhead is mostly below 0.02%. Hence, we ignore the bits for camera motion when calculating the two metrics. Table 1 lists the rate savings and quality gains of three test videos, where it can be observed that 4.0%, 8.51%, 2.20% bitrate savings and 0.12, 0.25, 0.07 quality gains for different test sequences can be achieved, respectively. On average, our proposed algorithm can achieve a 0.15 dB improvement relative to H.265 at the same bitrate or saves 4.91% of the bitrate to achieve the same PSNR. We also provide the RD curves in Figure 6 for better illustration. The blue lines in Figure 6 represent the performance of the traditional codec H.265 and the red ones indicate that of the proposed algorithm. It is obvious that the red line is above the blue line in each subplot, which demonstrates the superior RD performance of the proposed algorithm on different sequences. Moreover, our algorithm can achieve better RD performance at both low and high bit rates, proving the superiority of the proposed coding optimization algorithm.

Table 1. BD-Rate and BD-PSNR on test videos relative to H.265.

Sequences	BD-Rate	BD-PSNR
Secret Shop	−4.03%	0.1210
Robot Repair	−8.51%	0.2540
VRChat	−2.20%	0.0761
Avreage	−4.91%	0.1503

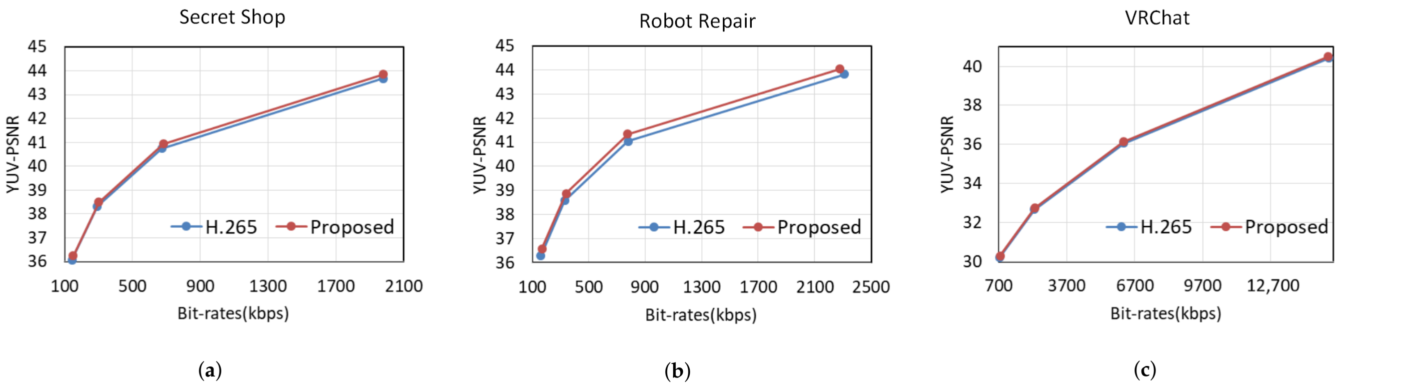


Figure 6. Rate-distortion curves of three test videos. (**a**) Secret Shop. (**b**) Robot Repair. (**c**) VRChat.

We also give the encoding and decoding time of the proposed algorithm in Table 2 to compare the complexity. All experiments are tested on an Intel (R) Core (TM) i7-9700K CPU @ 3.60 GHz processor and an NVIDIA GeForce RTX 2060 GPU. The forward operation of the proposed CMGNet is conducted with GPU acceleration, and the remaining operations are performed by the CPU. In Table 2, we can observe that compared with H.265/HEVC, our algorithm takes 137% of the encoding time, 9856% of the decoding time, and 157% of the total time. The proposed scheme has little increase in encoding complexity while greatly influencing the decoding time. The reasons are the long network inference time and the additional time-consuming CPU-GPU memory transfer operation. Choosing a high-performance GPU device can speed up inference time to a certain extent. Considering the complexity is positively related to parameters and floating-point operations per second (FLOPS) of the model, we can simplify the model to speed up the network forward operation in the future. High computational complexity is one of the disadvantages of CNN-based methods, and it is also an important factor affecting the implementation of the decoder. Further optimization to speed up the operation of the network is crucial and is currently under our investigation.

Table 2. Encoding and decoding complexity.

Sequences	ΔT_{Enc}	ΔT_{Dec}	ΔT_{Total}
Secret Shop	142%	11,023%	164%
Robot Repair	140%	11,796%	163%
VRChat	130%	6747%	144%
Avreage	137%	9856%	157%

4.2.2. Quality Enhancement of the Proposed CMGNet

In this section, we evaluate the quality enhancement performance of our proposed CMGNet in terms of PSNR and SSIM, which represent the degree of distortion and structure similarity between the enhanced frames and the original ones, respectively. Since our goal is to generate a high-matching and high-quality reference frame to update the reference frame list, the generated reference frame is supposed to be closer to the original one than that directly generated by H.265. Therefore, we compare the quality of the generated enhanced frame with that of the unenhanced frame compressed by H.265. Table 3 presents the results averaged over 500 frames for each test sequence at four different QPs. As shown in this table, our proposed CMGNet outperforms H.265 consistently in terms of PSNR and SSIM. Specifically, with QP equal to 22, the proposed network can achieve a PSNR gain of about 0.5 dB on average over the three test sequences. Even at QP 37, we can still obtain about a 0.3 dB performance gain. The same superiority is achieved in terms of SSIM.

Meanwhile, our algorithm exhibits different degrees of performance gains under different test sequences, showing a good generalization enhancement effect.

Table 3. Quantitative performance on PSNR and SSIM at four QPs.

Sequences	**Methods**	**QP = 22**		**QP = 27**		**QP = 32**		**QP = 37**	
	PSNR(dB)	**SSIM**	**PSNR(dB)**	**SSIM**	**PSNR(dB)**	**SSIM**	**PSNR(dB)**	**SSIM**	
Secret Shop	H.265	43.754	0.9417	40.845	0.9071	38.419	0.8642	36.183	0.8121
	CMGNet	**44.214**	**0.9477**	**41.320**	**0.9169**	**38.815**	**0.8743**	**36.479**	**0.8275**
Robot Repair	H.265	43.850	0.9572	41.105	0.9347	38.677	0.9051	36.405	0.8648
	CMGNet	**44.372**	**0.9613**	**41.606**	**0.9409**	**39.145**	**0.9139**	**36.806**	**0.8750**
VRChat	H.265	40.494	0.9453	36.169	0.8931	32.763	0.8217	30.267	0.7382
	CMGNet	**40.856**	**0.9493**	**36.518**	**0.9013**	**33.055**	**0.8318**	**30.508**	**0.7534**
Average	H.265	42.699	0.9481	39.373	0.9116	36.620	0.8637	34.285	0.8050
	CMGNet	**43.147**	**0.9528**	**39.815**	**0.9197**	**37.005**	**0.8733**	**34.598**	**0.8186**

Bold indicates the best performance.

We also provide qualitative evaluations in Figure 7 to demonstrate the visual performance of our model. Taking QP equal to 37 as an example, three patches from different sequences are enlarged and presented below. We recommend further zooming in on the figure for a more intuitive observation. Compared to the raw frame, the unenhanced frame compressed by H.265 inevitably loses a lot of details at such a high QP. For instance, the appearance outline of the sound in the second picture becomes extremely blurred, especially around the oval-shaped loudspeaker. The enhanced frame generated by the proposed CMGNet looks smoother with fewer artifacts due to the recovery of details. It is the same in other pictures. Although the quality of the enhanced frame is not as good as the original frame, it has been demonstrated in the previous section that the enhanced frame is sufficient to optimize the traditional codec.

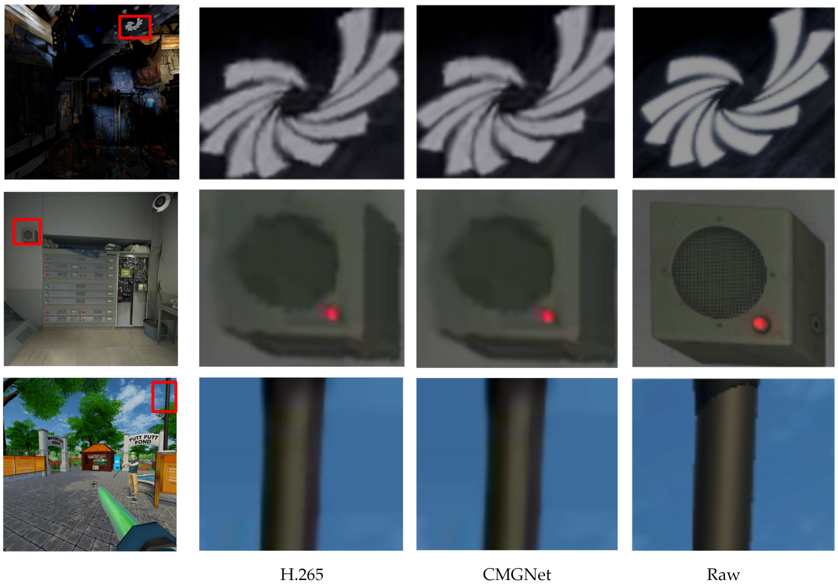
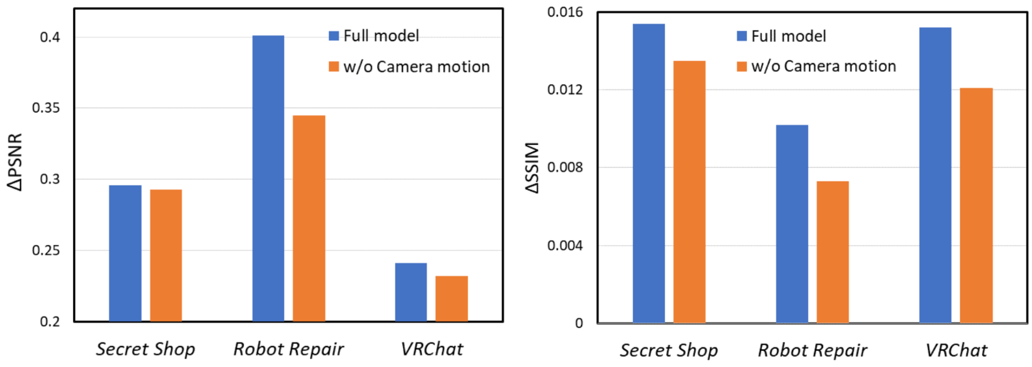


Figure 7. Qualitative comparisons of the proposed CMGNet.

To further analyze the impact of the camera motion in our network, we present the quality enhancement difference between the full model and that without camera motion in Figure 8, taking the QP equal to 37 as an example. It can be clearly seen from Figure 8 that the camera motion information plays a positive role in the quality enhancement, as the full model shows better performance than the model without motion information on both PSNR and SSIM metrics in the three test videos.



(**a**) (**b**)

Figure 8. Quality enhancement difference under QP = 37. (**a**) ΔPSNR. (**b**) ΔSSIM.

4.2.3. Comparison to State-of-the-Arts

In this section, we compare the proposed CMGNet with state-of-the-art video quality enhancement networks, i.e., MFQE2.0 [43], STDF [44], and TDAN [60]. It is noted that TDAN was originally designed for super-resolution tasks. Since it is a typical network using the deformable convolution for frame alignment, we remove the Pixel-Shuffle layer to extend it suitable for quality enhancement for comparison with the proposed CMGNet. Moreover, all of the compared methods take the preceding frame, i.e., X_{t-1}, and the succeeding frame, i.e., X_{t+1} , to help enhance the quality of the target frame, i.e., X_t, while our proposed model takes four preceding frames in the reference frame list as inputs. In addition, the compared models are all trained on camera-captured nature videos while we aim at cloud gaming videos. For a fair comparison, we modified parts of their architectures and trained them from scratch using the same game videos dataset as ours. The overall quantitative performance measured by PSNR and SSIM is shown in Table 4. We also show the parameters (sum of the luma network and the chroma network) of each model in the first row.

Table 4. Comparison to State-of-the-arts.

QP	**Sequences**	**MFQE 2.0 (408.03 K)**		**STDF (540.51 K)**		**TDAN (2.74 M)**		**CMGNet (Proposed) (1.96 M)**	
		PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
22	Secret Shop	44.210	0.9487	44.193	0.9476	44.199	0.9471	44.214	0.9477
	Robot Repair	44.258	0.9599	44.352	0.9613	44.332	0.9612	44.372	0.9613
	VRChat	40.829	0.9513	40.864	0.9500	40.793	0.9492	40.856	0.9493
	Average	43.099	**0.9533**	43.136	0.9530	43.108	0.9525	**43.147**	0.9528
27	Secret Shop	41.269	0.9170	41.245	0.9156	41.210	0.9124	41.320	0.9169
	Robot Repair	41.527	0.9383	41.540	0.9403	41.373	0.9380	41.606	0.9409
	VRChat	36.476	0.9026	36.496	0.9010	36.375	0.8981	36.518	0.9013
	Average	39.757	0.9193	39.760	0.9190	39.653	0.9162	**39.815**	**0.9197**
32	Secret Shop	38.809	0.8751	38.761	0.8747	38.740	0.8626	38.815	0.8743
	Robot Repair	39.118	0.9094	39.185	0.9141	39.173	0.9133	39.145	0.9139
	VRChat	33.030	0.8334	33.044	0.8326	32.990	0.8300	33.055	0.8318
	Average	36.986	0.8726	36.997	**0.8738**	36.967	0.8687	**37.005**	0.8733
37	Secret Shop	36.479	0.8130	36.410	0.8244	36.235	0.7825	36.479	0.8275
	Robot Repair	36.670	0.8615	36.767	0.8723	36.767	0.8730	36.806	0.8750
	VRChat	30.502	0.7450	30.507	0.7528	30.505	0.7532	30.508	0.7534
	Average	34.550	0.8065	34.561	0.8165	34.502	0.8029	**34.598**	**0.8186**
Average		38.598	0.8879	38.614	0.8906	38.558	0.8851	**38.641**	**0.8911**

Bold indicates the best performance.

It can be seen in Table 4 that the proposed model performs the best compared to the other three models in general, where PSNR has an increase of 0.02 dB∼0.08 dB and SSIM has an increase of 0.005∼0.06 on average. Taking an average of PSNR under each QP, our model achieves the highest performance. Although in terms of SSIM, our model is sub-optimal at QP 22 and 32, the decrease is negligible (i.e., only about 0.0005) compared to the best performance. In a word, the proposed CMGNet achieves satisfactory quality enhancement compared to these four state-of-the-art models. On the other hand, in terms of the number of model parameters, the TDAN has the most parameters, but it achieves the worst performance. Although the proposed model has more parameters than those of the MFQE and STDF, the proposed model achieves better performance. To put it differently, the proposed model achieves advanced tradeoffs between coding performance and computational complexity.

5. Conclusions

In this paper, we propose a coding optimization algorithm for cloud gaming videos. Specifically, an enhanced reference frame is generated through the proposed camera motion-guided network (i.e., CMGNet) and added to the reference frame list for the participant in the coding process, thus improving the coding performance. Moreover, the proposed CMGNet takes the known reconstructed reference frames in the reference frame list together with the camera information as inputs and generates a reference frame with high-quality and high-relation to the current frame through the camera motion-guided prediction module, the alignment and fusion module, and the reconstruction module in turn. Extensive experimental results demonstrated the superior performance of the proposed algorithm.

Author Contributions: Author Contributions: Y.W.: methodology, conceptualization, formal analysis, investigation, software, validation, and writing; H.W.: software, investigation, visualization, formal analysis; K.W.: resources, software, investigation; W.Z.: methodology, supervision, funding acquisition, writing—review, and validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 61801364).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cai, W.; Shea, R.; Huang, C.Y.; Chen, K.T.; Liu, J.; Leung, V.C.; Hsu, C.H. A survey on cloud gaming: Future of computer games. *IEEE Access* **2016**, *4*, 7605–7620. [CrossRef]
2. Mossad, O.; Diab, K.; Amer, I.; Hefeeda, M. DeepGame: Efficient Video Encoding for Cloud Gaming. In Proceedings of the Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 1387–1395.
3. Xu, L.; Guo, X.; Lu, Y.; Li, S.; Au, O.C.; Fang, L. A low latency cloud gaming system using edge preserved image homography. In Proceedings of the 2014 IEEE International Conference on Multimedia and Expo (ICME), Chengdu, China, 14–18 July 2014; pp. 1–6.
4. Lin, J.; Liu, D.; Li, H.; Wu, F. Generative adversarial network-based frame extrapolation for video coding. In Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, 9–12 December 2018; pp. 1–4.
5. Zhao, L.; Wang, S.; Zhang, X.; Wang, S.; Ma, S.; Gao, W. Enhanced ctu-level inter prediction with deep frame rate up-conversion for high efficiency video coding. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 206–210.
6. Zhao, L.; Wang, S.; Zhang, X.; Wang, S.; Ma, S.; Gao, W. Enhanced motion-compensated video coding with deep virtual reference frame generation. *IEEE Trans. Image Process.* **2019**, *28*, 4832–4844. [CrossRef]
7. Sullivan, G.J.; Ohm, J.R.; Han, W.J.; Wiegand, T. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]

8. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H. 264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [CrossRef]
9. Zhao, T.; Wang, Z.; Kwong, S. Flexible mode selection and complexity allocation in high efficiency video coding. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 1135–1144. [CrossRef]
10. Zhang, J.; Kwong, S.; Zhao, T.; Pan, Z. CTU-level complexity control for high efficiency video coding. *IEEE Trans. Multimed.* **2017**, *20*, 29–44. [CrossRef]
11. Zhang, J.; Kwong, S.; Zhao, T.; Wang, X.; Wang, S. Complexity Control for HEVC Inter Coding Based on Two-Level Complexity Allocation and Mode Sorting. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 3628–3632.
12. Correa, G.; Assuncao, P.; Agostini, L.; da Silva Cruz, L.A. Complexity scalability for real-time HEVC encoders. *J. Real-Time Image Process.* **2016**, *12*, 107–122. [CrossRef]
13. Norkin, A.; Bjontegaard, G.; Fuldseth, A.; Narroschke, M.; Ikeda, M.; Andersson, K.; Zhou, M.; Van der Auwera, G. HEVC deblocking filter. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1746–1754. [CrossRef]
14. Fu, C.M.; Alshina, E.; Alshin, A.; Huang, Y.W.; Chen, C.Y.; Tsai, C.Y.; Hsu, C.W.; Lei, S.M.; Park, J.H.; Han, W.J. Sample adaptive offset in the HEVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1755–1764. [CrossRef]
15. Helle, P.; Oudin, S.; Bross, B.; Marpe, D.; Bici, M.O.; Ugur, K.; Jung, J.; Clare, G.; Wiegand, T. Block merging for quadtree-based partitioning in HEVC. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1720–1731. [CrossRef]
16. Lainema, J.; Bossen, F.; Han, W.J.; Min, J.; Ugur, K. Intra coding of the HEVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1792–1801. [CrossRef]
17. Belyaev, E.; Forchhammer, S. An efficient storage of infrared video of drone inspections via iterative aerial map construction. *IEEE Signal Process. Lett.* **2019**, *26*, 1157–1161. [CrossRef]
18. Wang, X.; Hu, R.; Wang, Z.; Xiao, J. Virtual background reference frame based satellite video coding. *IEEE Signal Process. Lett.* **2018**, *25*, 1445–1449. [CrossRef]
19. Li, H.; Ding, W.; Shi, Y.; Yin, W. A double background based coding scheme for surveillance videos. In Proceedings of the 2018 Data Compression Conference, Snowbird, UT, USA, 27–30 March 2018; pp. 420–420.
20. Wang, G.; Li, B.; Zhang, Y.; Yang, J. Background modeling and referencing for moving cameras-captured surveillance video coding in HEVC. *IEEE Trans. Multimed.* **2018**, *20*, 2921–2934. [CrossRef]
21. Yan, N.; Liu, D.; Li, H.; Wu, F. A convolutional neural network approach for half-pel interpolation in video coding. In Proceedings of the 2017 IEEE international symposium on circuits and systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017; pp. 1–4.
22. Zhang, H.; Song, L.; Luo, Z.; Yang, X. Learning a convolutional neural network for fractional interpolation in HEVC inter coding. In Proceedings of the 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, 10–13 December 2017; pp. 1–4.
23. Yan, N.; Liu, D.; Li, H.; Li, B.; Li, L.; Wu, F. Convolutional neural network-based fractional-pixel motion compensation. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 840–853. [CrossRef]
24. Yan, N.; Liu, D.; Li, H.; Xu, T.; Wu, F.; Li, B. Convolutional neural network-based invertible half-pixel interpolation filter for video coding. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 201–205.
25. Zhao, Z.; Wang, S.; Wang, S.; Zhang, X.; Ma, S.; Yang, J. Enhanced bi-prediction with convolutional neural network for high-efficiency video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *29*, 3291–3301. [CrossRef]
26. Dong, C.; Deng, Y.; Loy, C.C.; Tang, X. Compression artifacts reduction by a deep convolutional network. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 11–18 December 2015; pp. 576–584.
27. Park, W.S.; Kim, M. CNN-based in-loop filtering for coding efficiency improvement. In Proceedings of the 2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), Bordeaux, France, 11–12 July 2016; pp. 1–5.
28. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef]
29. Dai, Y.; Liu, D.; Wu, F. A convolutional neural network approach for post-processing in HEVC intra coding. In Proceedings of the International Conference on Multimedia Modeling, Reykjavik, Iceland, 4–6 January 2017; pp. 28–39.
30. Zhang, Y.; Shen, T.; Ji, X.; Zhang, Y.; Xiong, R.; Dai, Q. Residual highway convolutional neural networks for in-loop filtering in HEVC. *IEEE Trans. Image Process.* **2018**, *27*, 3827–3841. [CrossRef]
31. Xia, J.; Wen, J. Asymmetric convolutional residual network for av1 intra in-loop filtering. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 1291–1295.
32. Zhang, S.; Fan, Z.; Ling, N.; Jiang, M. Recursive residual convolutional neural network-based in-loop filtering for intra frames. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 1888–1900. [CrossRef]
33. Kang, J.; Kim, S.; Lee, K.M. Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 26–30.
34. Wang, D.; Xia, S.; Yang, W.; Hu, Y.; Liu, J. Partition tree guided progressive rethinking network for in-loop filtering of HEVC. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 2671–2675.

35. Li, D.; Yu, L. An in-loop filter based on low-complexity CNN using residuals in intra video coding. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019; pp. 1–5.
36. Zhu, H.; Xu, X.; Liu, S. Residual convolutional neural network based in-loop filter with intra and inter frames processed, respectively, for Avs3. In Proceedings of the 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), London, UK, 6–10 July 2020; pp. 1–6.
37. Lu, M.; Cheng, M.; Xu, Y.; Pu, S.; Shen, Q.; Ma, Z. Learned quality enhancement via multi-frame priors for HEVC compliant low-delay applications. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 934–938.
38. Tong, J.; Wu, X.; Ding, D.; Zhu, Z.; Liu, Z. Learning-based multi-frame video quality enhancement. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 929–933.
39. Meng, X.; Deng, X.; Zhu, S.; Zhang, X.; Zeng, B. A robust quality enhancement method based on joint spatial-temporal priors for video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2020**, *31*, 2401–2414. [CrossRef]
40. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; Brox, T. Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 2758–2766.
41. Sun, D.; Yang, X.; Liu, M.Y.; Kautz, J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8934–8943.
42. Yang, R.; Xu, M.; Wang, Z.; Li, T. Multi-frame quality enhancement for compressed video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6664–6673.
43. Guan, Z.; Xing, Q.; Xu, M.; Yang, R.; Liu, T.; Wang, Z. MFQE 2.0: A new approach for multi-frame quality enhancement on compressed video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 949–963. [CrossRef] [PubMed]
44. Deng, J.; Wang, L.; Pu, S.; Zhuo, C. Spatio-temporal deformable convolution for compressed video quality enhancement. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10696–10703.
45. Zhao, M.; Xu, Y.; Zhou, S. Recursive fusion and deformable spatiotemporal attention for video compression artifact reduction. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 5646–5654.
46. Lin, J.; Huang, Y.; Wang, L. FDAN: Flow-guided deformable alignment network for video super-resolution. *arXiv* **2021**, arXiv:2105.05640.
47. Chan, K.C.; Zhou, S.; Xu, X.; Loy, C.C. BasicVSR++: Improving video super-resolution with enhanced propagation and alignment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 5972–5981.
48. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
49. Kim, J.; Lee, J.K.; Lee, K.M. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE Conference on cOmputer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
50. Ranjan, A.; Black, M.J. Optical flow estimation using a spatial pyramid network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4161–4170.
51. Chen, P.; Yang, W.; Wang, M.; Sun, L.; Hu, K.; Wang, S. Compressed Domain Deep Video Super-Resolution. *IEEE Trans. Image Process.* **2021**, *30*, 7156–7169. [CrossRef] [PubMed]
52. ALVR. Available online: https://github.com/alvr-org (accessed on 8 May 2022).
53. The Lab. Available online: https://store.steampowered.com/app/450390/The_Lab/ (accessed on 23 May 2022).
54. Half-Life: Alyx. Available online: https://store.steampowered.com/app/546560/HalfLife_Alyx/ (accessed on 23 May 2022).
55. Hover The Edge. Available online: https://store.steampowered.com/app/1822130/Hover_The_Edge/ (accessed on 23 May 2022).
56. Rec Room. Available online: https://store.steampowered.com/app/471710/Rec_Room/ (accessed on 23 May 2022).
57. VRChat. Available online: https://store.steampowered.com/app/438100/VRChat/ (accessed on 23 May 2022).
58. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
59. Bjøntegaard, G. Calculation of average PSNR differences between RD-curves. *VCEG-M33* **2001** , *16090*, 1520–9210.
60. 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, Seattle, WA, USA, 14–19 June 2020; pp. 3360–3369.