VIDEO COMPRESSION VIA PREDICTABLE CODING OVER VIRTUAL FRAMES

VIDEO COMPRESSION VIA PREDICTABLE CODING OVER VIRTUAL FRAMES

BY

YING CHEN, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2020) McMaster University

(Electrical and Computer Engineering) Hamilton, Ontario, Canada

TITLE:	Video Compression via Predictable Coding over Virtual Frames
AUTHOR:	Ying Chen B.Eng. (Electronic Information Science and Technology), University of Electronic Science and Technology of China, Chengdu, China
SUPERVISOR:	Dr. Jun Chen
NUMBER OF PAGES:	xii, 49

Abstract

Video applications have become more and more common in the past few decades, in the meantime, optimizing video coding has received more attention. Existing video codecs usually focus on the encoder itself, and try to do everything possible to compress video with spatial (intraframe) compression and temporal (interframe) compression with the premise of reasonable distortion rate and video performance. In this work, we proposed a practical approach to improve video coding efficiency at a lower bitrate, which is to combine traditional Video Codec with interpolation neural network. A new concept called "virtual frames" was proposed and applied to video coding process. We use raw frames as Ground Truth and virtual frames to train the interpolation neural network GDCN (Generalized Deformable Convolution Network), then encode the video synthesized with virtual frames via traditional AV1 video codec. With the pre-trained network, we could simply reconstruct the frames. This method can significantly improve the video compression effect compared with traditional video codec technology.

To my dear parents and lovely sister, to the happiness with my friends.

Acknowledgements

This thesis would not be achievable without a very dedicated group of people.

I want to thank Dr. Jun Chen for helping me to re-evaluate many of my choices throughout the development process. Thank him for teaching me how to run proper user studies and step up the idea and carry out a series of experiments.

Next, I would like to sincerely thank my committee members, Dr. Sorina Dumitrescuand and Dr. Jiankang Zhang for their valuable suggestions and kind comments.

It is always a significant form to thank your sponsors. Thanks Mom and Dad for being happy with my progress and always making sure I am well fuelled. Also, many thanks to Mrs. Cheryl Gies for kind support.

Finally, thank you, Wei Hao, for pretending to be clueless to make me feel like I know what I am doing. Thank you, Kangdi, for keeping Wei Hao in line and making sure I am always up to speed with all the new missions that keep coming up with. And a special thank you to Zhihao and Xiaohong for being Super Genius and blinking lights fast, for all their support and patience throughout the neural network training process. It took a while, but thanks for helping me make it.

Contents

List of Figures

List of Tables

Abbreviations

Abbreviations

ABR	Average Bitrate
AOMedia	Alliance for Open Media
AR	Augmented Reality
AV1	AOMedia Video 1
FCN	Fully Convolutional Network
GDCN	Generalized Deformable Convolution Network
GT	Ground Truth
FRUC	Frame Rate Up-Conversion
HEVC	High Efficiency Video Coding
ITU	International Telecommunication Union
MC	Motion Compensation

ME	Motion Estimation
MEPG	Motion Expert
MV	Motion Vector
VR	Virtual reality
QVI	Quadratic Video Interpolation
QP	Quantization Parameter
RDO	Rate-Distortion Optimization
PSNR	Peak signal-to-noise ratio Picture Group

Chapter 1

Introduction and Problem Statement

1.1 Introduction

Today, digital video is rapidly being used for a range of purposes, such as high-definition video entertainment, video advertisements, video conference and user-generated virtual business meetings. As technology evolves, video quality expectations of users are increasing, and users expect high-resolution video even if it is transmitted over a communication channel with limited bandwidth.

A primary fundamental technology for digital media applications is video coding technology. The first Industrial Digital Video Encoding Standard H.120 was released in 1984 by the International Telecommunication Union(ITU). The new video-coding technology is supporting digital television and IPTV (Weijia *et al.* (2014)), video

surveillance, internet video services (Zheng *et al.* (2014)), Augmented Reality or Virtual Reality(AR/VR) and other video applications after almost 30 years of development.

The main purpose of video encoding technology is to effectively store and distribute vast volumes of video information and to ensure that the visual experience of the user with limited resources is optimised to the highest extent possible. Video quality has been enhanced over the past 30 years through the production of video capture and display tools. The industry of video is also booming. Commercially available now are ultra HD videos with a 4 K (4096 x 2160 pixel) spatial resolution and a temporal resolution of over 50 frames per second.

The basic framework of the existing video coding technology has not changed significantly in 30 years. It is based on a hybrid coding framework. This framework always follows the image block structure, with the prediction, transformation, quantization, entropy coding process.

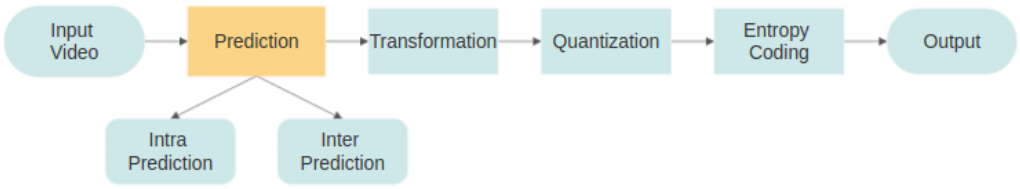


Figure 1.1: Hybrid Coding Framework

Every procedure in hybrid coding is continually evolving. Improve the compression efficiency of video through the flexible combination mode of hundreds of processing algorithms, and then solve the problem of increased complexity with the help of Moore's Law(Zhou *et al.* (2017)). The theoretical idea is mainly from the signal processing theory, derivation and design of different prediction methods, transformation

methods, and so on, then using the rate-distortion optimization (RDO) method to make coding mode selection, from a set of coding methods or parameters to choose the optimal one according to the least rate-distortion cost criteria. RDO's target is to minimize distortion without exceeding the given maximum code rate, namely

$$\min\{D\}, s.t. R \leq R_{max} \tag{1.1.1}$$

converted Equation 1.1.1 by Lagrange multiplier method:

$$\min\{J = D + \lambda * R\} \tag{1.1.2}$$

where J indicates the Lagrangian cost function and λ is Lagrange multiplier.

As mentioned earlier, the resolution requirements are getting higher and higher. When people watch videos at a frame rate of 30 frames per second, visual perception presents a major lagging problem. Converting the low frame rate of video through the Frame Rate Up-Conversion(FRUC) into high frame rate video will make the video more continuous and smoother and enhance the video playback effect effectively. Moreover, this technique can also be used in other scenes, such as playback with slow-lens. The conversion technology on the frame rate of video image mainly refers to the motion estimation and motion compensation of the video image, which is a kind of video post-processing technology. Estimating the motion vector by means of the moving object of the original video, and then completes the conversion of frame rate from low to high by interpolation. This technology is also used in video communication. In order to reduce the transmission bandwidth during video transmission, it is necessary to reduce the transmission frame rate. At the receiving end of the video,

the frame rate needs to be restored, which also reduces visual stagnation and jumping. Besides, the current flat-screen device has some shortcomings, such as image jitter, motion blur, motion tailings, etc. When the frame rate conversion algorithm applied, these issues have been greatly improved. That makes FRUC become a hot topic.

As deep learning continues to be successful in the computer vision field, researchers are trying to combine deep learning with video frame interpolation technology. Video interpolation refers to the use of the relevant information between the adjacent front and rear frames to obtain the intermediate frame. This technique can be used effectively in video compression.

1.2 Motivation

To effectively reduce the amount of data transmitted in the existing video coding process, redundant information is removed, and the frame rate in the video transmission process is reduced. And through video frame interpolation technology, we can efficiently reconstruct the frame. These kinds of research inspire us. What if we compress video with traditional video codec at a lower bitrate and reconstruct with Neural network? We could get the intermediate frames through the frame interpolation neural network, which we named them "virtual frames". Then, encode the video synthesized of virtual frames with traditional video codec, finally, using a revised interpolation network retrained with a different "virtual frames" dataset to reconstruct the decoded frames. Before we implemented, we compared several latest video interpolation approaches and standard video codecs respectively, and selected the most appropriate ones. In this thesis, we will carry out the validation.

1.3 Thesis Structure

The rest of this paper is arranged as follows: In Chapter 2, several state-of-the-art frame interpolation approaches are introduced. Next, in Chapter 3, We listed the standard video codecs of recent years. In Chapter 4, the proposed video compression via predictable coding over virtual frames has been explained in detail. Two kinds of encoding mode of AV1 are illustrated, and two different videos are selected in each encoding mode for presentation. Experiments results in numeric comparisons have presented in graphics. Finally, in Chapter 5, we conclude and take a forecast in future works.

Chapter 2

Frame Interpolation Related Work

2.1 Frame rate conversion

Frame rate conversion can be divided into two kinds: low frame rate to high frame rate and high frame rate to low frame rate. Both of them construct non-existent frames on the source video frame, and there is only a difference in the number of frames built.

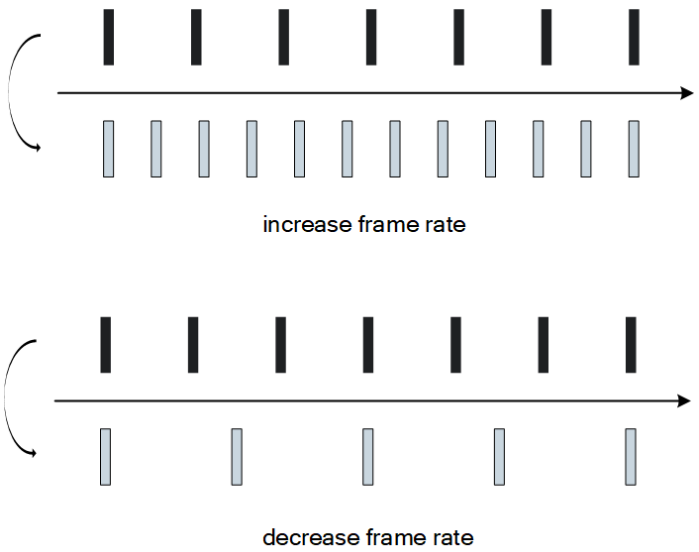


Figure 2.1: Two kinds of frame rate conversion(source:Tai (2019))

Constructing non-existent frames, which we call frame interpolation, have three implementations:

1.Duplication, Copy similar frames

2.Blend, Blend with two adjacent frames

3.Motion Interpolation, Construct intermediate frames based on image motion

2.1.1 Duplication

The frame on the output time node is the frame we need to generate, which is called the intermediate frame here. In the process of generating the current intermediate frame, two input frames are involved. The closest previous input frame to the output time (assuming the n-frame) and the nearest next input frame to the output time (considering the n+1 frame), which we can call them reference frames. Select a frame from these two reference frames that is closest to the intermediate frame output time, and duplicate the frame as the current intermediate frame.

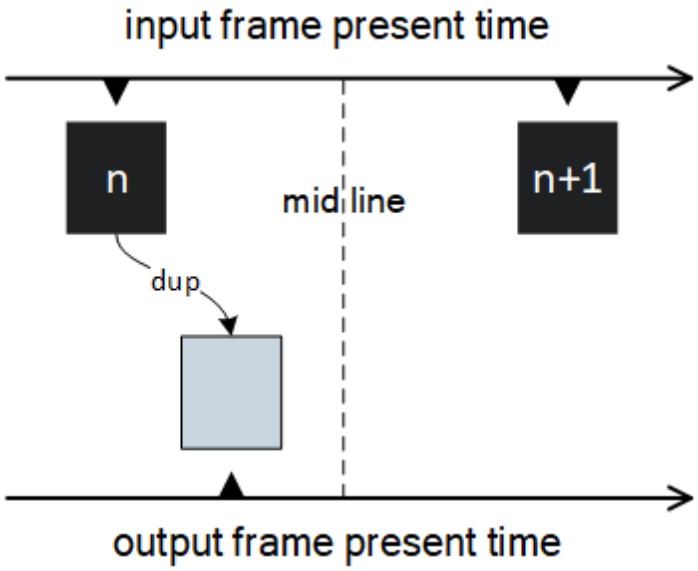


Figure 2.2: Frame interpolation - Duplication(source:Tai (2019))

2.1.2 Blend

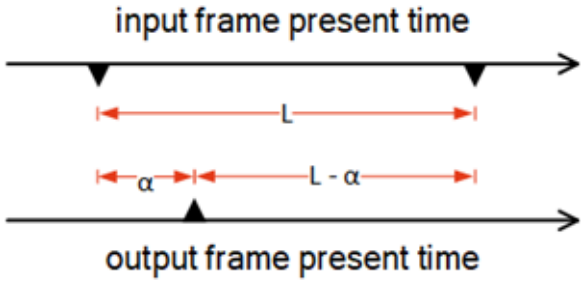
As discussed in the previous Duplication method, the intermediate frame also involves two reference frames adjacent to the left and right. The difference is that the implementation of blend is to mix the two reference frames in a certain way. The approach is to assign weight to two frames. Or we can call it transparency, when opaque, the weight is 1; when fully transparent, weight is 0. After multiplying their respective weights with each pixel of two frames, then add it, you can get the pixel value in the corresponding position of the intermediate frame.These two weights should meet two conditions:

1.The two weights should be the values in the range of [0,1], and the sum is equal to 1.

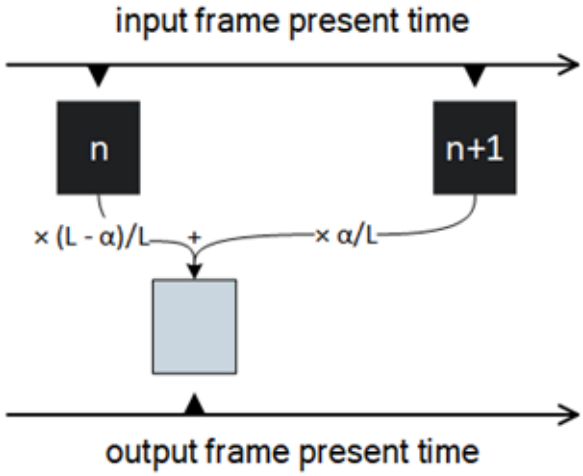
2.The frame which is further from the middle frame is more transparent, would

have a smaller weight; the frame that is closer to the intermediate frame is less transparent, would have a higher weight.

The weight of the reference frame on the left should be $(L-\alpha)/L$, and the transparency of the reference frame on the right should be α/L, by adding two reference frames by their respective weights, we could get the intermediate frame.



(a) Figure 1



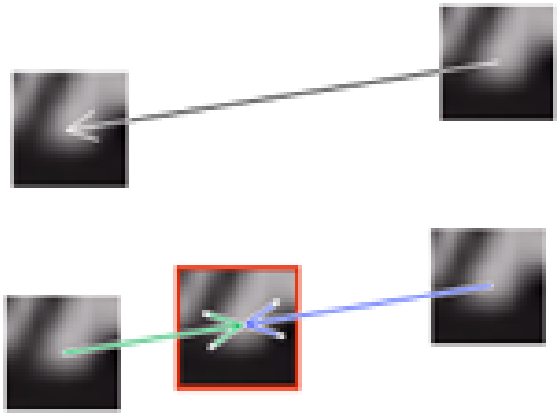
(b) Figure 2

Figure 2.3: Frame interpolation - Blend(source:Tai (2019))

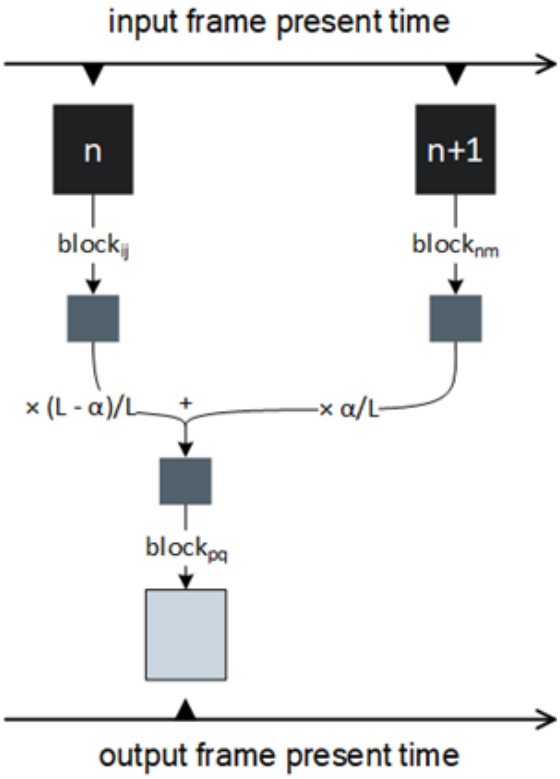
2.1.3 Motion Interpolation

In general, the time interval between the adjacent frames is very short, so if the content in the two frames changes less, we can think of the change as linear motion. If the motion trajectory of the linear motion can be derived, the content position adjustment can be made according to the motion trajectory and the time relationship of the input and output frame, which is called motion interpolation. This method will make the video appear smoother.

In fact, the content in the video is complicated. We can not simply think that all parts of a moving object in one direction as linear motion. If the video frame is divided into small pieces, the moving object is broken down into pieces of moving colour blocks. For these blocks, we can think of them as linear motion. Therefore, the motion relationship (motion vector) between the small pieces on the middle frame and the corresponding small pieces of the reference frame can be obtained according to the time relationship. The motion vector allows you to position the two reference blocks of the two reference frames, as well as the position of the blocks that need to be generated on the intermediate frame. Then synthesize the blocks with the Blend method described above. When all the blocks within the frame have been synthesized, we could get the frame slotted into the desired motion.



(a) Figure 1



(b) Figure 2

Figure 2.4: Frame interpolation - Motion interpolation(source:Tai (2019))

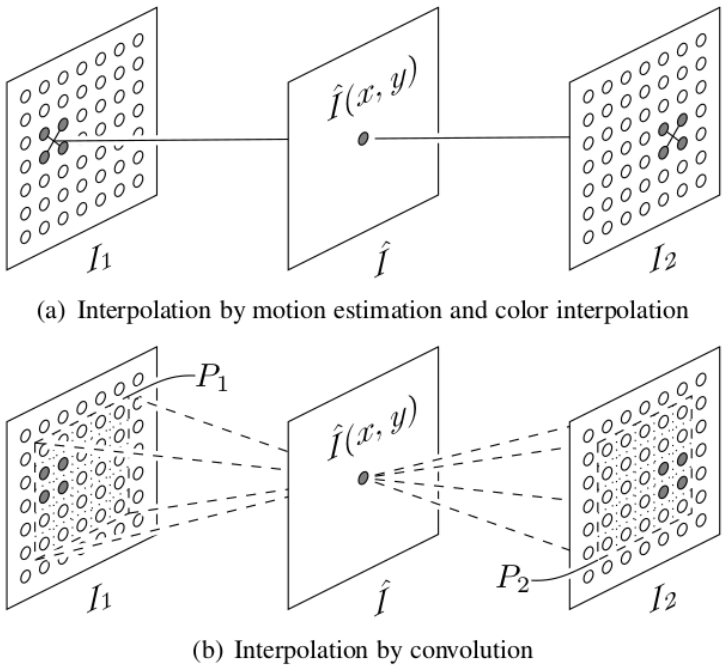
2.2 Frame Interpolation Approaches

Recently, various learning based frame interpolation methods have been proposed based on deep CNNs. The training datasets typically include picture triplets from raw video sequences, with the first and third frames feeding into the network as inputs, and the second intermediate frame working as ground truth (Bao *et al.* (2019)). The latest deep learning-based frame interpolation can be classified into flow-based and kernel-based approaches. They have their own characteristics, and we will present a representative algorithm for both approaches, respectively.

2.2.1 Adaptive Convolution

Video frame interpolation typically involves two steps: motion estimation and pixel synthesis. However, this method can be affected if occlusion, motion blur and lack of texture exist. Niklaus *et al.* (2017) combines these two steps into a process, and the pixel synthesis of interpolated frames is treated as a local convolution of two input frames. Convolution kernels capture the coefficients of local motion and pixel synthesis between input frames. Niklaus *et al.* (2017) uses a deep fully convolutional network (FCN) to estimate the spatially-adaptive convolution kernels for each pixel. Its training process can be done end-to-end in widely available video data without the need of any hard-to-obtain ground-truth(GT) data, such as optical flow.

See the following figure for the principle of pixel interpolation through convolution. For each output pixel $\hat{I}(x, y)$, Niklaus *et al.* (2017) estimate the convolution kernel K and generate interpolated image $\hat{I}(x, y)$ with two patches $P_1(x, y)$ and $P_2(x, y)$ in two input frames centred on (x, y) respectively.



(a) Interpolation by motion estimation and color interpolation

(b) Interpolation by convolution

Figure 2.5: Adaptive Convolution (source:Niklaus *et al.* (2017))

This kernel-based approach achieves high-quality results by applying Spatially Adaptive Convolution Kernels for each pixel. Nonetheless, Jiang *et al.* (2017) indicated that predicting kernel for each pixel is computer-cost and memory-intensive. Also, the motion that can be handled is limited to the kernel size.

2.2.2 Quadratic Video Interpolation

Previous video interpolation work has generally assumed that the motion between frame objects is uniform, so a linear model is usually used for frame interpolation. However, the motion in real-world scenes is complex and nonlinear, and traditional linear models can lead to inaccurate interpolation results. Xu *et al.* (2019) proposed a quadratic video interpolation (QVI) method which exploits the acceleration information in videos. This approach makes it possible to predict the curvilinear trajectory

and a variable velocity and to achieve more accurate results.

Figure 2.6: Xu *et al.* (2019) proposed quadratic model can better reflect the actual reality pixel movement and further generate more accurate results.

Xu *et al.* (2019) proposed two key modules for quadratic video interpolation: the Quadratic Flow Prediction and the Flow Reversal Module, enabling calculation of optical flow in both directions to obtain the final video frame interpolation.

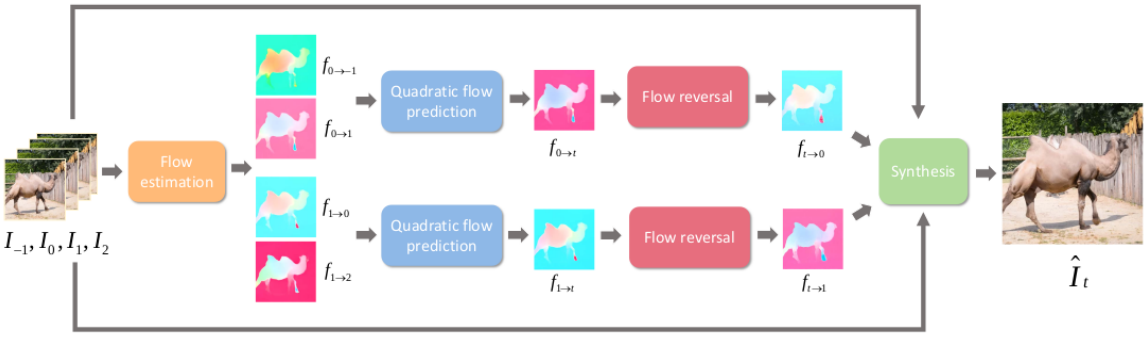


Figure 2.7: Overview of the quadratic video interpolation algorithm (source:Xu *et al.* (2019))

I_{-1}, I_0, I_1, I_2 are four consecutive frames of the input video. Given any moment t, the model will eventually generate the intermediate frame $\hat{I}_t$ where $t \in (0, 1)$. Quadratic flow prediction is the process of getting displacement in the motion of uniform variable speed. Assuming that the motion at the moment of $t \in (-1, 1)$ is a constant acceleration motion, the velocity v_0 at time 0 and acceleration a within the interval can be estimated using the displacement, so that the displacement $f_{0\to t}$ of pixel from 0 to t can be calculated. Symmetrically we can get $f_{1\to t}$. The reverse

flow $f_{t\rightarrow}$ and $f_{t\rightarrow 1}$ are obtained through the flow reversal module. With two backward flow, new video frames can be synthesized.

Although this approach is more reliable for estimating motion trajectories in contrast to the traditional flow-based methods, which typically assume linear uniform movement between two consecutive frames. Shi *et al.* (2020) noted that a single mathematical model could not completely comprehend real-world motions'dynamics and irregularities. And the displacement of the pixel level in flow-based methods is substantially inadequate to handle diffusion and dispersion effects.

2.2.3 Generalized Deformable Convolution Interpolation

A novel mechanism called generalized deformable convolution is proposed to solve those performance-limiting issues mentioned above, which can effectively gather motion information data and freely select sampling points in space-time. Based on this manner, Shi *et al.* (2020) developed a new method for video interpolation.

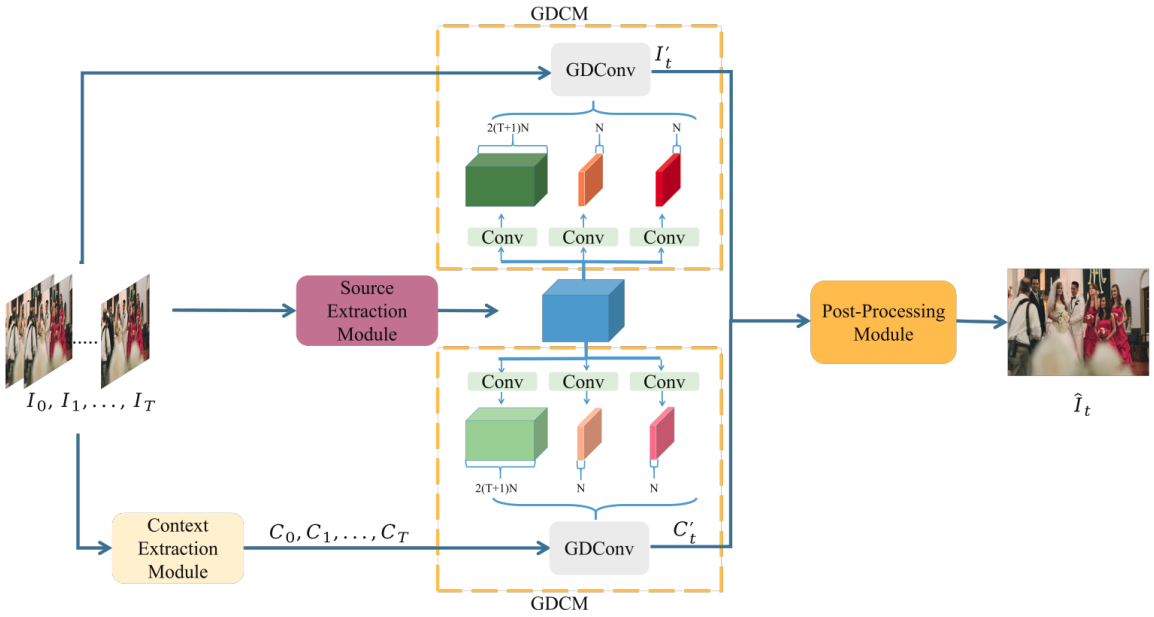


Figure 2.8: Illustration of Generalized Deformable Convolution Network (source:Shi *et al.* (2020))

The GDCN feature is to synthesize an intermediate frame $\hat{I}_t$ where $t \in (0, T)$ with given certain source frames I_0, I_1, ..., I_T. The entire cycle can be divided into four stages roughly. First, using context extraction module (CEM) (Niklaus and Liu (2018)) to extract context maps C_0, C_1, ..., C_T from I_0, I_1, ..., I_T. Next step is to construct a generalized deformable convolution module (GDCM) source feature map from source extraction module (SEM). For getting $C_t{}'$ and $I_t{}'$, the third move would be warping context maps C_0, C_1, ..., C_T and source frames I_0, I_1, ..., I_T respectively through the GDCM source map. Finally, fed $C_t{}'$ and $I_t{}'$ into a post-processing module (PM) to obtain the interpolation result $\hat{I}_t$.

Based on the author's experimental results, the performance of GDCN and QVI is comparable. However, it is worth noting that QVI's model size is six times of GDCN's.

Chapter 3

Video Codec Related Work

3.1 High Efficiency Video Coding (HEVC)

High Efficiency Video Coding (HEVC), also known as H.265, is a new generation of video coding technology after Advanced Video Coding (AVC/H.264). It is jointly released in 2013 by Moving Picture Experts Group (MPEG) and ITU Telecommunication Standardization Sector Video Coding Experts Group(ITU-T VCEG). HEVC has improved the techniques in the coding process while continuing to follow the hybrid coding framework. This makes coding efficiency significantly improved on the basis of AVC, reducing the bitrate by 40%-50% while ensuring the same quality of coding.

The entire architecture of HEVC is divided into intra-prediction, inter-prediction, transform/quantization, adaptive filtering, and entropy coding.The entire structure is shown in Figure 3.1

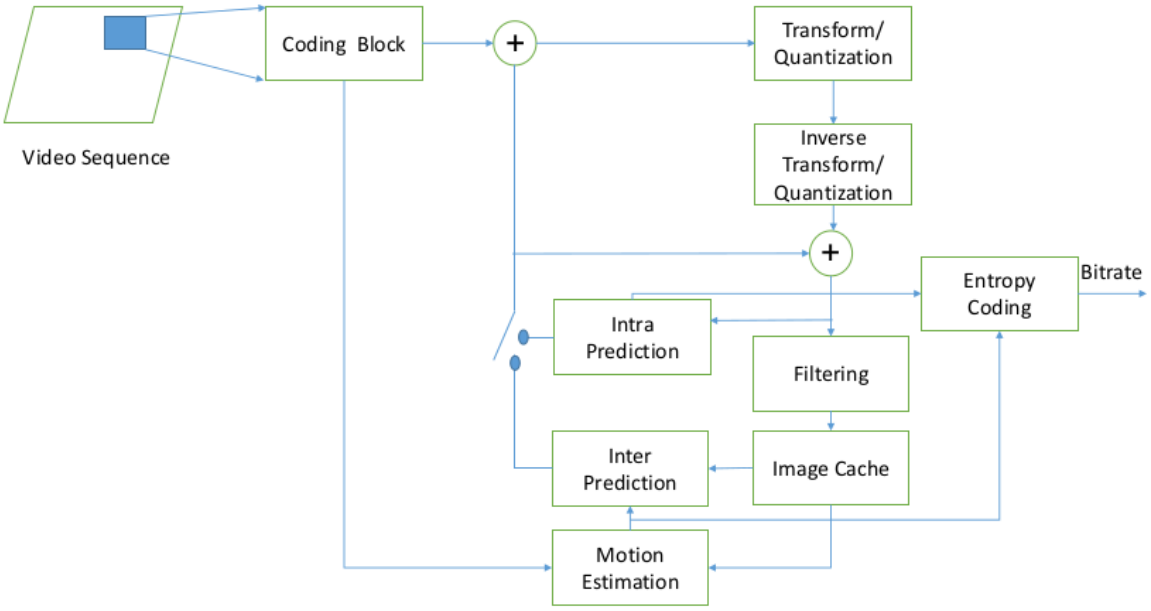


Figure 3.1: High Efficiency Video Coding (H.265/HEVC) hybrid coding framework(source:Jiang *et al.* (2020))

Compared to previous coding standards, the H.265 uses some innovative technologies, such as flexible split and residual coding structures based on large-size quadtree, Adaptive Motion Parameter coding, Adaptive In-loop Filtering and so on. These new technologies improve the coding efficiency of the H.265 standard. In contrast to H.264, here are some highlights of HEVC:

(1) encodes with larger macroblocks called coding tree units (CTU), which can be 16 x 16, 32 x 32, and 64 x 64 pixels in size. Code the CTU to a CU (coding unit) through a quadtree structure. Some CUs are converted to PU (prediction unit) in prediction mode.

(2) In order to compress the encoded information of motion vectors, HEVC uses Advanced motion vector prediction (AMVP), including several candidates on the basis of data from adjacent PBs and the reference picture. Sullivan *et al.* (2012)

noted a merge mode for MV coding allows the inheritance of MVs from temporally or spatially neighbouring PBs.

(3) HEVC's Intra_Angular prediction has a total of 33 directions, covering a range of 180 degrees. It also provides two alternative prediction modes, Intra_DC and Intra_Planar (Sullivan *et al.* (2012)). For a total of 35 prediction modes (Figure 3.2). The multi-directional prediction modes can increase the accuracy of prediction, reduce the residual data of images and increase the efficiency of data compression.

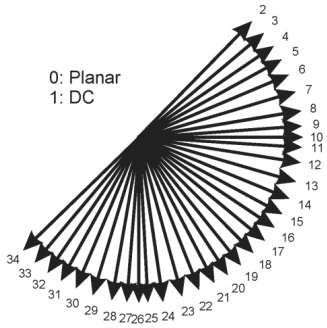


Figure 3.2: Modes and directional orientations for intrapicture prediction(source:Sullivan *et al.* (2012))

3.2 AOMedia Video 1 (AV1)

As a new generation of high-performance open-source and royalty free coding standards, AV1 is favoured by more and more enterprises. The AV1 coding standard is inherited from Google's VP8 in 2011 and VP9 in 2013 and has been developed and maintained by the Alliance for Open Media (AOM). AOM was established by more than 30 companies, including Google, in 2015. AV1 version 1.0 has been released in June 2018 (de Rivaz and Haughton (2018)). With advanced coding tools, AV1 saves 30% bitrate compared to VP9 and HEVC on the premise of same video coding

quality.

Similar to other video coding software, AV1 is divided into a series of continuous modules, including segmentation, prediction, transformation, quantization, entropy coding, in-loop filtering and so on. In the waysome features involved in hybrid video coding using AV1 are highlighted as follows. For more information, Chen *et al.* (2018) made a comprehensive description.

(1) On coding block partition, AV1 supports more partition patterns and larger partition blocks. The predecessor VP9 supported a maximum partition block size of 64x64, which allows the division of subblocks recursively for each block until it reaches the smallest 4x4. AV1 supports a maximum partition of 128x128, with 10 types of partition structure as shown in Figure 3.3.

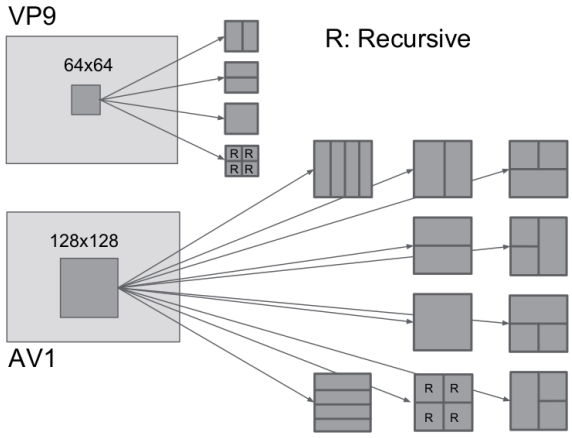


Figure 3.3: Partition tree in VP9 and AV1(source:Chen *et al.* (2018))

(2) AV1 has completely upgraded the former intra prediction technology and has added a variety of new prediction modes. AV1 exploits 56 directional intra prediction modes, also, expands non-directional prediction by adding smooth predictors. It supports predicting chroma from illumination and adds colour palette mode and intra block copying mode.

Predicting chroma from luma, it takes advantage of the high similarity between the luma channel and the chroma channel in the video image and predicts the chroma value by selecting the appropriate parameters to reconstruct the luma channel, which has a good compression effect for video game. Another tool that is important for screen content compression is called Intra Block Copy. It works by searching the image portion of the current frame when predicting the current block, such as the second and the third letter "t" in Figure 3.4. It will find the first "t" in the predicted block that has been encoded, which will be very accurate and achieve higher compression efficiency. The graph contains a huge number of letters, will get excellent prediction results with Intra Block Copy.

Figure 3.4: Intra block copy

(3) Motion compensation in inter prediction is a critical module in video coding. AV1 exploits a range of measures to improve the accuracy of motion compensation. For example, adopted more comprehensive and diverse reference frames. In contrast with 3 references LAST(nearest past) frame, GOLDEN(distant past) frame and

ALTREF(temporal filtered future) frame of VP9 , AV1 adds another 4 references to a total of 7 references, two future frames (BWDREF and ALTREF2) and two near past frames (LAST2 and LAST3). Also, the proposed overlapped Block Motion Compensation (OBMC) improves the accuracy of block edges prediction. In addition to translational motion, AV1 adds two affine motion compensation modes, global warped motion compensation and local warped motion compensation. For the global warping, model is estimated from the encoder side by some feature matching and the parameter will be conveyed at frame level. It works great with camera motions like panning, zoom and rotation. For the local warping, AV1 estimated the local warping models by linear square fitting from the conveyed neighbourhood motion vectors, so the decoder can reproduce the warping models by doing a low-complexity four-parameter least square, works great on graphical contents.

3.3 Comparison of State-of-the-Art Video Codecs

In this section, we draw on the latest "HEVC/AV1 Video Codecs Comparison test report" produced by MSU Graphics and Media Lab (Video Group) Vatolin *et al.* (2019) to see how these video codecs perform. We randomly picked a video sequence to compare the performance of four different codecs based on AV1 and HEVC standards. The following Table 3.1 lists all participated codec for performance comparison:

Codec name	Developed by	Standard
aom	AOMedia	AV1
SVT-AV1	Open Visual Cloud	AV1
SVT-HEVC	Open Visual Cloud	HEVC
x265	MulticoreWare, Inc.	HEVC

Table 3.1: Participated Codec for performance comparison

Classic RD curves are the critical charts for this comparison. This chart displays differences in codec quality by bitrate or file size. A higher value typically indicates better quality for this metric. Based on the following graphs, we can intuitively see the advantages of AV1 standard in coding effects.

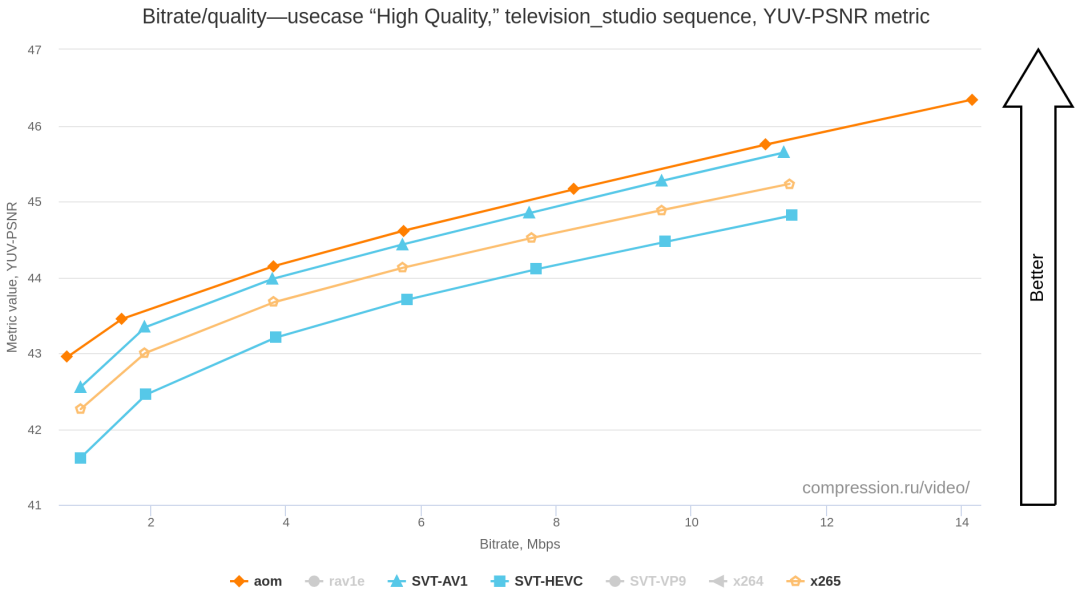


Figure 3.5: RD curves of television_studio video sequence(source:Vatolin *et al.* (2019))

Nevertheless, AV1 also has its apparent drawback, which is the speed of encoding.

We can see that the encoding speed of AV1 and SVT-AV1 based on aom standard is significantly slower than HEVC standard codec.

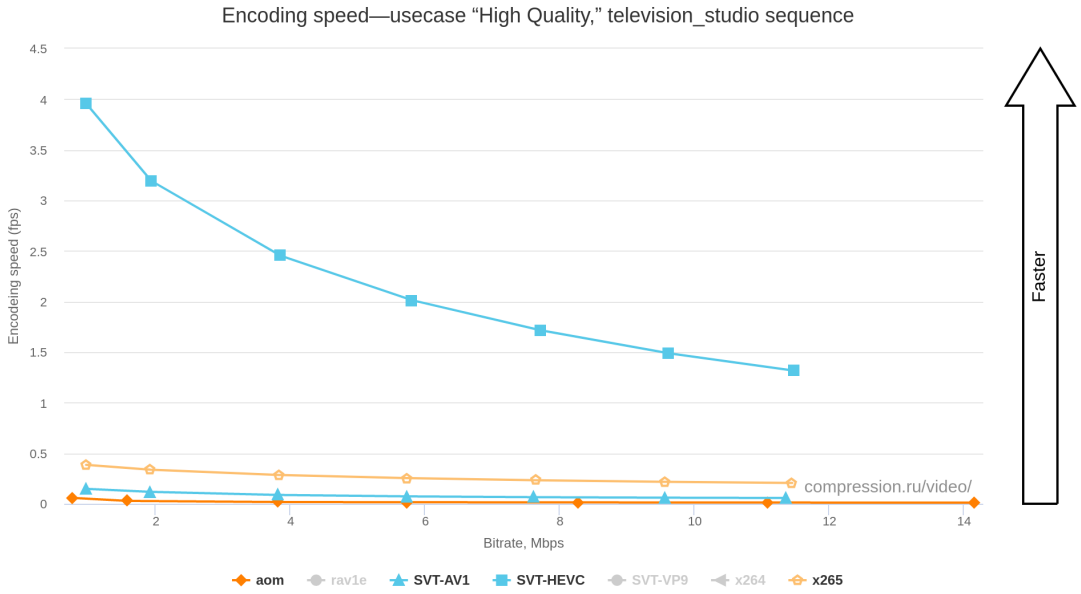


Figure 3.6: Encoding speed of television_studio video sequence(source:Vatolin *et al.* (2019))

Chapter 4

Implementation and Experimental Result

By comparing the different interpolation algorithms and video codecs in chapter 2 and chapter 3, we employed Generalized Deformable Convolution Interpolation (Shi *et al.* (2020)) and AV1 to complete our experiment of video compression over virtual frames.

4.1 Preparation

The entire experiment is split into three parts.

Part I: use FFmpeg to cut video into n frames (I_1, I_2... I_n) and generate $n-1$ virtual frames ($I_{1.5}$, $I_{2.5}$... $I_{(n-1).5}$) via the Generalized Deformable Convolution Interpolation network (Shi *et al.* (2020)).

Part II: encode the video consisting of virtual frames with AV1 codec and use FFmpeg to cut into frames after decoding.

Part III: a pre-trained reconstruction network is used to recreate the $n-1$ frame obtained in the second stage into $n-2$ frames ($\hat{I}_2$, $\hat{I}_3$... $\hat{I}_{n-1}$).

Generate virtual frames

The video database is downloaded from AOMedia google git, we randomly selected one test video for explanation. With the first and second frames I_1, I_2 from raw Garden video sequences feeding into the network as inputs, the intermediate virtual frame $I_{1.5}$ was generated. A raw video sequence with n frames could generate $n-1$ virtual frames.

(a) Raw frame I_1

(b) Raw frame I_2

(c) Virtual frame $I_{1.5}$

Figure 4.1: Generate virtual frames by GDCN

Retrain the interpolation network

In Part I, the training dataset for the network used to generate the virtual frames includes image triplets from raw video sequences, with the first and third frames feeding into the network as inputs, and the second intermediate frame working as ground truth. For the reconstruction network in Part III, we need to retrain the network with different dataset. Every triplet consists of two decoded virtual frames and one raw image as ground truth.

With a large number of encoded virtual frames covering diverse content types, including humans, animals, natural scenes, architectures, etc.(Li *et al.* (2019)), we use two consecutive virtual frames, for example $I_{1.5}$, $I_{2.5}$, and one raw frame I_2 as ground truth to generate a triplet training set sample, as shown in Figure 4.2. A total of 22GB training image datasets and 5GB validation datasets were created to retrain and validate the network. The training process is carried out on a PC with four NVIDIA GeForce GTX 1080Ti. In order to distinguish it from the interpolation network in the Part I, we would note it as "reconstruction network".

(a) Virtual frame $I_{1.5}$

(b) Virtual frame $I_{2.5}$

(c) Ground truth (Raw frame I_2)

Figure 4.2: Example triplet for training our reconstruction network

Reduce the bitrate

Under the premise of ensuring the quality of the video, reducing the bitrate is a significant research problem of video compression. We increased the distortion within a reasonable range in the source code of AV1 to reduce the bitrate. Then compared with traditional video coding method, we will verify whether our approach can effectively improve the quality of coding. The rate-distortion in AV1 source code was set as bellow:

```
rdmult = rdmult * 3 + (rdmult * 2 / 3)
```

We have modified the number $\frac{2}{3}$ to $\frac{4}{3}$ and $\frac{6}{3}$ to see what effects it has on the actual average bitrate (ABR) with specified target bitrate. Different actual ABR and RD

curves with target bitrate are shown as following:

Parameter \ Target	100	140	180	220	260	300
AV1_2/3	145.26	150.75	188.69	235.34	279.83	327.73
AV1_4/3	136.08	145.36	183.28	227.96	272.97	316.12
AV1_6/3	127.68	141.98	180.24	223.21	267.08	309.46

Table 4.1: Actual ABR with different RD parameters

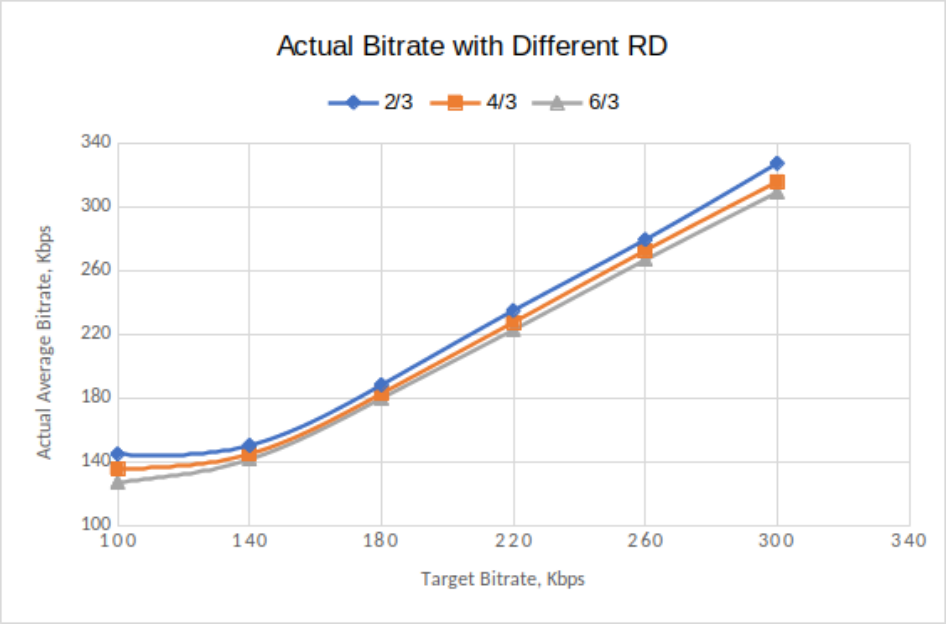


Figure 4.3: Actual ABR Comparison with Different RD parameters

Figure 4.3 intuitively displays that increasing the molecular elements in *rdmult* can effectively reduce the bitrate and increase the compression. In the following experiments, we will verify if it is possible to recover the loss of video quality due to the reduction of bitrate by reconstruction network.

One-pass and Two-pass coding mode

In this Section, we would illustrate the two encoding modes One-pass and Two-pass in AV1 before we go down.

One-pass is mostly used to live streaming media for fast coding purposes. In av1, the 1-pass encoding process uses only two reference frames, LAST and GOLDEN, to minimise the complexity. The latest encoded frame is used as the LAST reference frame and the first frame is still the GOLDEN frame to deal with intra and inter prediction on partition block. 1-pass is encoded under the constraint of a large quantization parameter (QP) which gathers such information without dramatically increasing complexity such as code decisions, motion vectors, etc(Alanqin (2019)). The QP controls the compression level for each Macroblock in a frame. Large values mean higher quantization, higher compression and poorer quality. Lower values reflect the opposite (Robitza (2017)). Once 1-pass encoding process is completed, all these information and operational results will be processed in a .fpf format logfile for 2-pass.

Two-pass, particularly for non-real-time video encoding, in layman's terms, is that we need to encode for a second time. AV1 uses all seven references in 2-pass encoding, LAST(nearest past) frame, GOLDEN(distant past) frame, ALTREF(temporal filtered future) frame, two future frames (BWDREF and ALTREF2) and two near past frames (LAST2 and LAST3). The encoder will measure the cost of encoding for each frame during the first encoding and assign the bits available more effectively for second encoding, which enable the encoder to obtain the best video quality at a certain code bitrate. Twice encoding means twice conversion, the compressed video will have better image quality, better detail and smaller volume. Correspondingly, it

will be much slower than 1-pass coding.

AV1 does 2-pass encoding by default but by modifying the parameters of the execution shell file we can set to 1-pass encoding. We set Average Bitrate to 100 to 300 with an interval of 40 and get the results of 1-pass and 2-pass coding. The comparison was made by two different types of videos, one is a character named "Suzie", and the other one is a natural landscape called "Garden". The results are shown as follows:

Suzie Sequence

Mode / Target	1-pass		2-pass	
	ABR	PSNR	ABR	PSNR
100	206.51	37.42	98.19	36.01
140	235.9	37.85	133.66	37.07
180	263.52	38.29	169.76	37.88
220	296.31	38.81	206.85	38.58
260	324.58	39.16	245.36	39.16
300	352.19	39.57	284.33	39.7

Table 4.2: One-pass and Two-pass encoding results of Suzie sequence

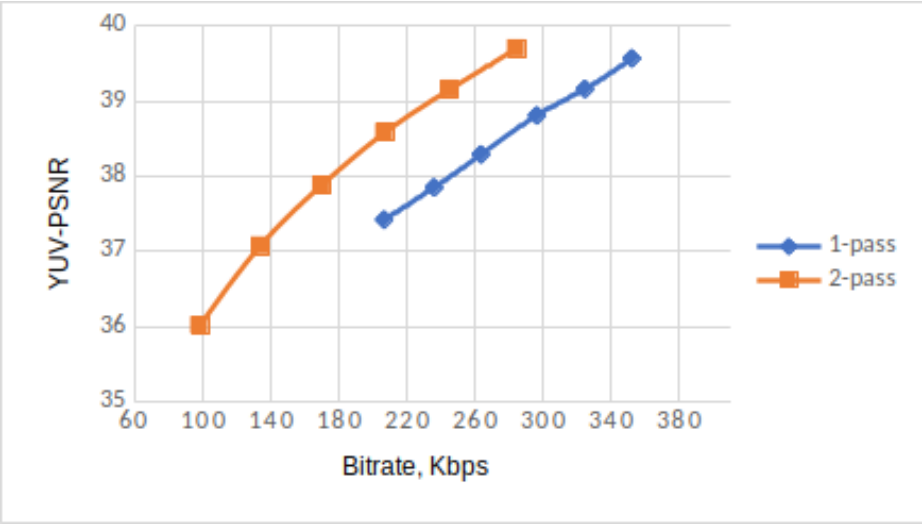


Figure 4.4: One-pass and Two-pass encoding curves of Suzie sequence

(a) Raw frame (b) One-pass (c) Two-pass

Figure 4.5: One-pass and Two-pass visual comparison of Suzie sequence

Garden sequence

Target \ Mode	1-pass		2-pass	
	ABR	PSNR	ABR	PSNR
100	300.13	21.95	145.26	20.46
140	363.67	22.66	150.75	21.08
180	412.89	23.26	188.69	21.87
220	455.92	23.84	235.34	22.66
260	497.56	24.33	279.83	23.33
300	539.75	24.76	327.73	23.94

Table 4.3: One-pass and Two-pass encoding results of Garden sequence

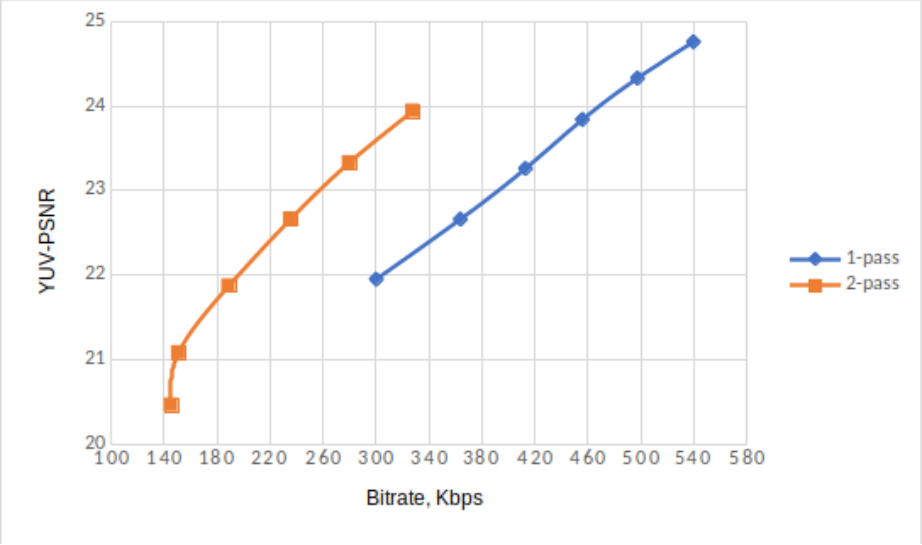


Figure 4.6: One-pass and Two-pass encoding curves of Garden sequence

(a) Raw frame (b) One-pass (c) Two-pass

Figure 4.7: One-pass and Two-pass visual comparison of Garden sequence

As can be seen from Figure 4.4 and Figure 4.6, 1-pass can not effectively compress the video to the specified bitrate interval (100 to 300), while 2-pass can do so under the premise of PSNR assurance. But in the actual coding process, 1-pass is about 5 to 10 times faster than 2-pass.

4.2 Video Compression over Virtual Frames

In this section, two different videos are selected in each encoding mode for presentation, one is a character called "Suzie", and the other one is a natural landscape named "Garden". Snapshots of source video sequences are shown in Figure 4.8

(1) Frame I_1

(2) Frame I_{58}

(a) Snapshots of Suzie video sequence

(1) Frame I_1

(2) Frame I_{45}

(b) Snapshots of Garden video sequence

Figure 4.8: Snapshots of source video sequences

4.2.1 One-pass Encoding Result

The comparative experiment consists of three parts. In the first part, the two videos are encoded with AV1 codec as one-pass mode without changing any parameters; the curve is identified as "AV1_2/3". In the second part, the *rdmult* parameter is changed to $\frac{4}{3}$, and compressed with AV1 again, and the curve is shown as "AV1_4/3". Keep the parameter as $\frac{4}{3}$ and proceed to part three, which is the method we proposed.

First, generate the virtual frames by GDCN, and encode virtual frames with 1-pass mode, finally, the decoded frames are fed into the retrained GDCN to reconstruct the corresponding raw frames, results curve legend is "Ours_4/3".

One-pass encoding of Suzie sequence

Target \ Parameter	AV1_2/3		AV1_4/3		Ours_4/3	
	ABR	PSNR	ABR	PSNR	ABR	PSNR
100	206.51	37.42	201.56	37.24	172.47	37.27
140	235.9	37.85	229.74	37.7	196.31	37.71
180	263.52	38.29	256.31	38.12	220.89	38.06
220	296.31	38.81	286.57	38.57	245.62	38.39
260	324.58	39.16	316.52	38.96	277.24	38.74
300	352.19	39.57	346.96	39.39	310.29	39.04

Table 4.4: One-pass encoding results of Suzie sequence

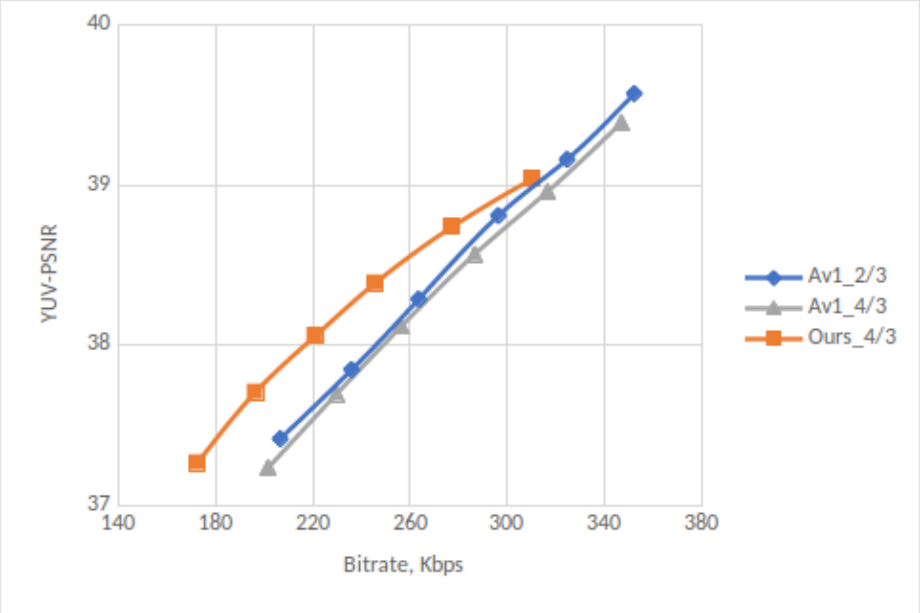


Figure 4.9: One-pass encoding RD curves of Suzie sequence

(a) Raw frame

(b) AV1_2/3

(c) AV1_4/3

(d) Ours_4/3

Figure 4.10: One-pass encoding visual comparison of Suzie sequence

One-pass encoding of Garden sequence

Target \ Parameter	AV1_2/3		AV1_4/3		Ours_4/3	
	ABR	PSNR	ABR	PSNR	ABR	PSNR
100	300.13	21.95	295.91	21.8	260.09	21.74
140	363.67	22.66	356.92	22.5	314.96	22.36
180	412.89	23.26	408.23	23.13	356.95	22.88
220	455.92	23.84	452.51	23.71	402.33	23.3
260	497.56	24.33	492.26	24.21	445.72	23.65
300	539.75	24.76	534.56	244.64	487.64	23.92

Table 4.5: One-pass encoding results of Garden sequence

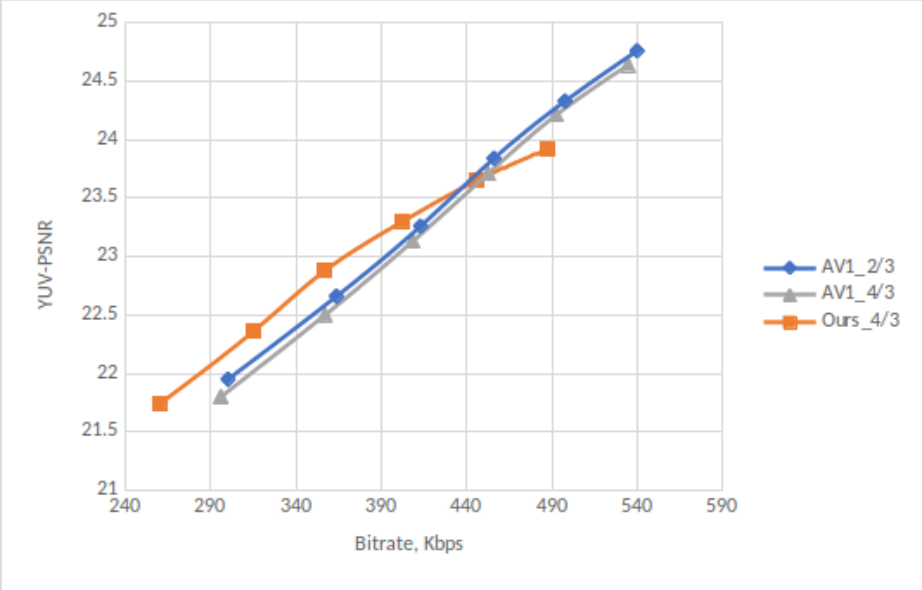
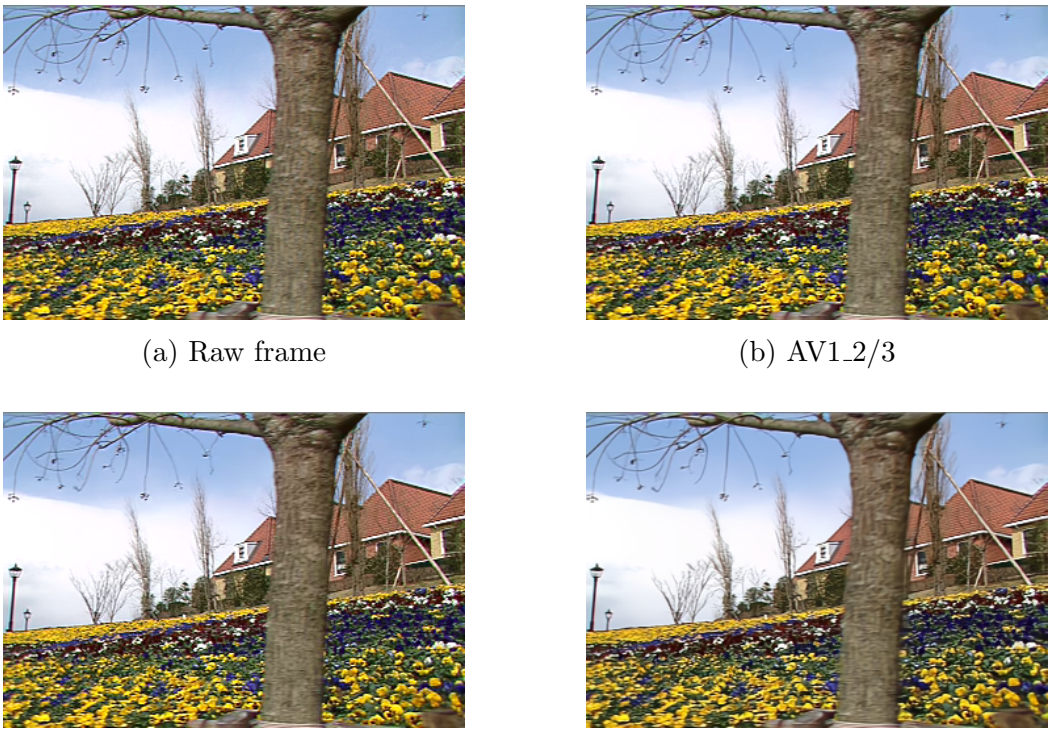


Figure 4.11: One-pass encoding RD curves of Garden sequence



(a) Raw frame (b) AV1_2/3

(c) AV1_4/3 (d) Ours_4/3

Figure 4.12: One-pass encoding visual comparison of Garden sequence

4.2.2 Two-pass Encoding Result

The comparison experiment process of 2-pass mode is similar to that of 1-pass, except that all coding processes are converted to 2-pass mode. As shown in Figure 4.13 and Figure 4.15, 2-pass mode compression could encode the video around the specified bitrate interval (100 to 300).

Two-pass encoding of Suzie sequence

Target \ Parameter	AV1_2/3		AV1_4/3		Ours_4/3	
	ABR	PSNR	ABR	PSNR	ABR	PSNR
100	98.19	36.01	95.61	35.88	93.98	36.6
140	133.66	37.07	131.33	36.82	129.12	37.51
180	169.76	37.88	167.43	37.69	165.14	38.26
220	206.85	38.58	204.41	38.42	202.16	38.81
260	245.36	39.61	240.94	39.01	239.06	39.3
300	284.33	39.7	280.03	39.53	277.69	39.71

Table 4.6: Two-pass encoding results of Suzie sequence

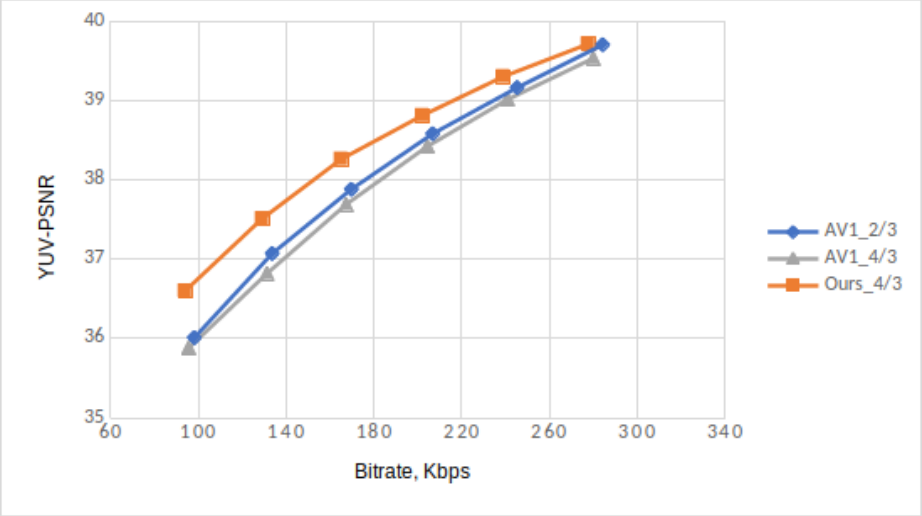


Figure 4.13: Two-pass encoding RD curves of Suzie sequence

(a) Raw frame

(b) AV1_2/3

(c) AV1_4/3

(d) Ours_4/3

Figure 4.14: Two-pass encoding visual comparison of Suzie sequence

Two-pass encoding of Garden sequence

Target \ Parameter	AV1_2/3		AV1_4/3		Ours_4/3	
	ABR	PSNR	ABR	PSNR	ABR	PSNR
100	145.26	20.46	136.08	20.69	103.99	20.55
140	150.75	21.08	145.36	21.01	146.47	21.5
180	188.69	21.87	185.28	21.7	194.59	22.23
220	235.34	22.66	227.96	22.47	236.69	22.83
260	279.83	23.33	272.97	23.13	282.29	23.3
300	327.73	23.94	316.16	23.73	325.21	23.67

Table 4.7: Two-pass encoding results of Garden sequence

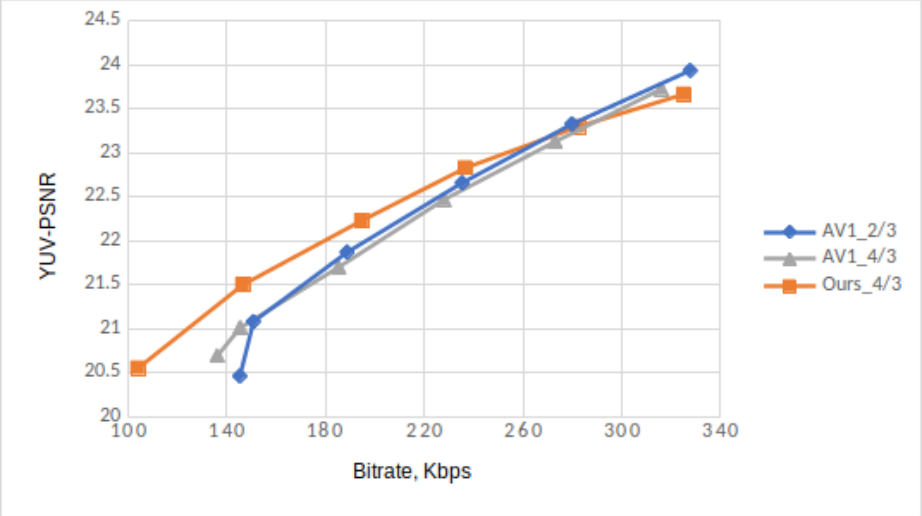
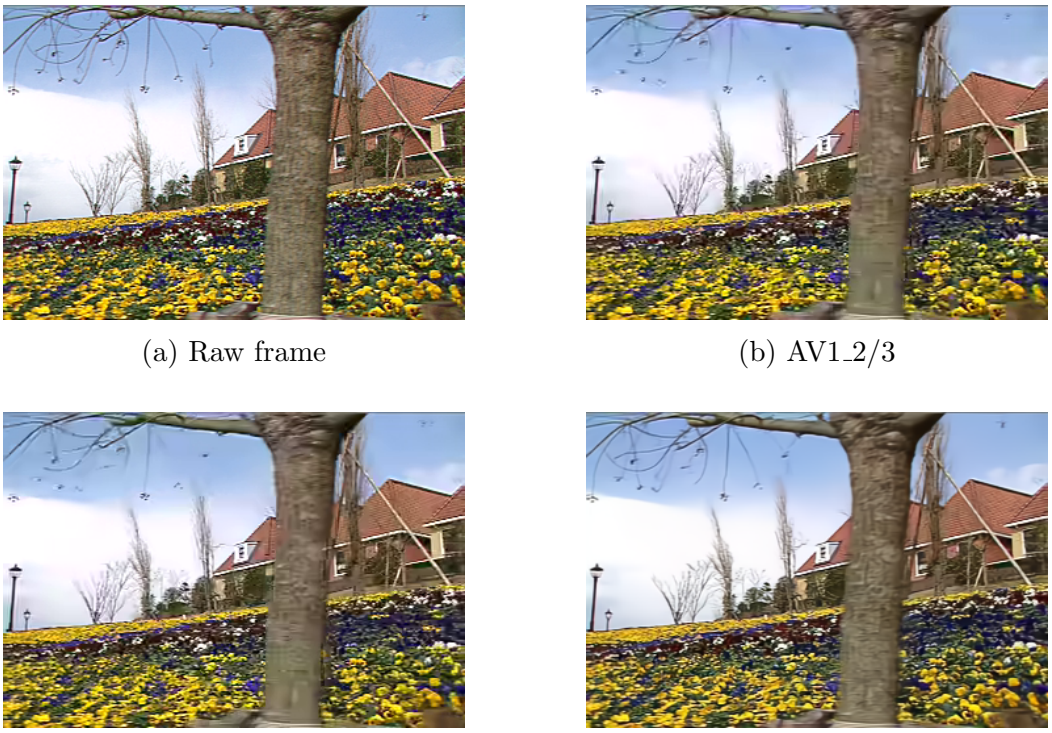


Figure 4.15: Two-pass encoding RD curves of Garden sequence



(a) Raw frame (b) AV1_2/3

(c) AV1_4/3 (d) Ours_4/3

Figure 4.16: Two-pass encoding visual comparison of Garden sequence

Chapter 5

Conclusion and Future Work

5.1 Summary

We proposed an effective way to improve video quality under the condition of saving bitrate. At lower bitrate, the video compression rate is higher and the image loss is more significant. In this case, frames can be refactored in conjunction with the interpolation network. The key concept of our proposed method is the "virtual frames" which is applied to the coding process. Compared with the conventional video coding stated in the advanced video compression standard, AV1, we can effectively improve the effect of frames at low-level bitrate.

Before validating our idea, we encoded the video with one-pass and two-pass modes, and the experiment results showed that one-pass could not adequately compress the video to the target bitrate. On the premise of PSNR assurance, two-pass can do so with almost 50% bitrate saving. As for coding speed, one-pass is about 5 to 10 times faster than 2-pass. One-pass and two-pass have their advantages and disadvantages. They are applied for different situations. One-pass is often used for

live streaming media for fast coding purposes, two-pass for situations where required better image quality smaller volume.

To validate our method, we cut a large number of videos for interpolation, then encode them with AV1 one-pass mode and two-pass mode, respectively. With the decoded frames as dataset, we retrain the interpolation network twice, one with 1-pass data, the other one with 2-pass data. The entire training set is about 22GB, and the validation set is about 5GB. After the training is completed, different types of videos are randomly selected for experiments. The experimental results show that in both one-pass and two-pass modes, our approach can effectively improve the video quality at a low bitrate.

5.2 Future Work

In the above demonstration, we can clearly see that when the bitrate reaches a certain value, the experimental effect starts to drop. In the case of the bitrate gradually increases, the picture quality becomes higher, the neural network's ability to recover the picture begins to weaken. Although this is a reasonable situation, we hope to find a way to improve it. By comparing with the original images I_2, I_3... I_{n-1}, adding error information to the reconstructed frames $\hat{I}_2$, $\hat{I}_3$... $\hat{I}_{n-1}$ may improve the experimental effect. This idea can be deployed in further study.

Bibliography

Alanqin, . (2019). AV1 encode and AV1 first pass. https://blog.csdn.net/qq_30945147/article/details/103477222.

Bao, W., Lai, W.-S., Zhang, X., Gao, Z., and Yang, M.-H. (2019). Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement. *IEEE transactions on pattern analysis and machine intelligence*.

Chen, Y., Murherjee, D., Han, J., Grange, A., Xu, Y., Liu, Z., Parker, S., Chen, C., Su, H., Joshi, U., *et al.* (2018). An overview of core coding tools in the av1 video codec. In *2018 Picture Coding Symposium (PCS)*, pages 41–45. IEEE.

de Rivaz, P. and Haughton, J. (2018). AV1 bitstream & decoding process specification. *The Alliance for Open Media*, page 182.

Jiang, H., Sun, D., Jampani, V., Yang, M.-H., Learned-Miller, E. G., and Kautz, J. (2017). Super slomo: High quality estimation of multiple intermediate frames for video interpolation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9000–9008.

Jiang, X., Song, T., and Katayama, T. (2020). Maximum-entropy-model-enabled

complexity reduction algorithm in modern video coding standards. *Symmetry*, **12**(1), 113.

Li, Z., Duanmu, Z., Liu, W., and Wang, Z. (2019). AVC, HEVC, VP9, AVS2 or AV1?—A Comparative Study of State-of-the-Art Video Encoders on 4K Videos. In *International Conference on Image Analysis and Recognition*, pages 162–173. Springer.

Niklaus, S. and Liu, F. (2018). Context-aware synthesis for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1710.

Niklaus, S., Mai, L., and Liu, F. (2017). Video frame interpolation via adaptive convolution. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2270–2279.

Robitza, W. (2017). Understanding rate control modes (x264, x265, vpx).

Shi, Z., Liu, X., Shi, K., Dai, L., and Chen, J. (2020). Video interpolation via generalized deformable convolution. In *2020 European Conference on Computer Vision (ECCV 2020)*.

Sullivan, G. J., Ohm, J.-R., Han, W.-J., and Wiegand, T. (2012). Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, **22**(12), 1649–1668.

Tai, G. (2019). Frame Interpolation. `https://www.cnblogs.com/TaigaCon/p/10612354.html`.

Vatolin, D., Kulikov, D., Erofeev, M., Antsiferova, A., Zvezdakov, S., Kondranin, D., Sklyarov, E., and Grokholskiy, S. (2019). HEVC/AV1 Video Codecs Comparison. `http://compression.ru/video/codec_comparison/hevc_2019/#hq_report_summary`.

Weijia, S., Li, J., and Liding, J. (2014). Technique comparative analysis of ott tv and iptv. *Telecommunications Science*, **30**(5), 15–19.

Xu, X., Si-Yao, L., Sun, W., Yin, Q., and Yang, M.-H. (2019). Quadratic video interpolation. In *NeurIPS*.

Zheng, W., Weijia, S., and Guqiao, Z. (2014). Multi-screen interaction technologies on internet streaming video. *Telecommunications Science*, **30**(5), 27–32.

Zhou, J., Yang, H., Liu, D., Ma, X., and Wang, T. (2017). Trends and technologies of video coding. *Telecommunications Science*, **33**(8), 16–25.