#### PROJECT MARCH -APRIL 2021

A REPORT

ON

**Video Compression**

BY

|  |  |
| --- | --- |
| Bir Anmol Singh | 2018A7PS0261P |
|  |  |

On Research Paper

**Learning Binary Residual Representations for Domain-specific Video Streaming**

Prepared for Multimedia Computing

Course No.

CS F401



[link: <https://research.nvidia.com/sites/default/files/pubs/2018-02_Learning-Binary-Residual/Learning%20Binary%20Residual%20Representations%20for%20Domain-specific%20Video%20Streaming.pdf> ]

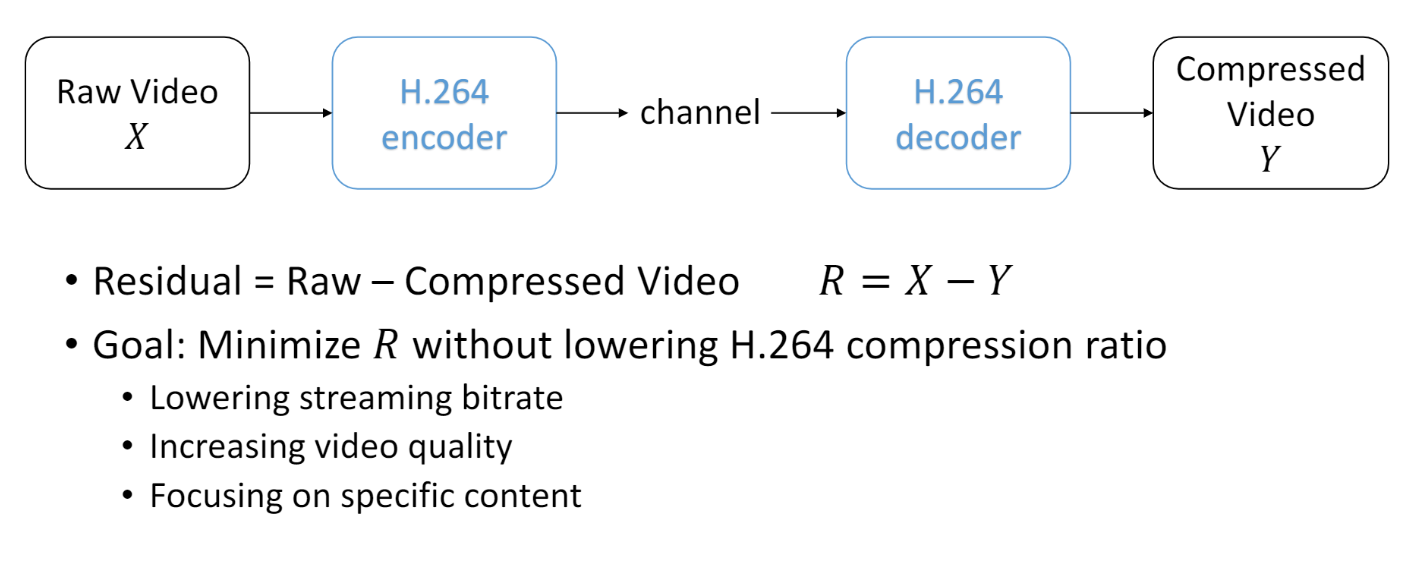
**Intended Work**

Video streaming services, such as Netflix and YouTube, have become primary source of entertainment nowadays. These websites have high resolution, large size videos which have to be sent over limited network bandwidth thus **video compression** is a necessary evil. While video compression can reduce the size of a video, it often comes with undesired compression artifacts, such as image blocking effects and blurry effects.

Video compressing has improved after decades of efforts on delivering the best possible video quality under bandwidth constraint. State of-the-art video compression methods such as **MPEG-4 (Li2001), H.264 (Wiegand et al. 2003), and HEVC (Sullivan et al. 2012)** combine various classical techniques **including image transform coding, predictive coding, source coding, and motion estimation** in a carefully-engineered framework. These methods are generic and can be applied to various video domains for effectively compressing most of the information in a video however, the residual information, which is the difference between the uncompressed and compressed videos, is **difficult to compress because it contains highly non-linear patterns**. This **residual information** can be preserved if compression is applied on stream belonging to a **specific domain**.

Majority of the stream we have, belongs to one kind of domain – **domain specificity**. If we target such a streaming setting where the videos to be streamed from a server to a client are all in the same domain and they have to be compressed to a small size for low-latency transmission, one can leverage this property of domain specificity to achieve better video quality over the conventional compression. A new compression pipeline can be used where we first apply H.264 to compress domain-specific videos, then train a novel **binary autoencoder** to encode the leftover domain-specific **residual information** frame-by-frame into binary representations. Using auto-encoder alongside with a H.264 compressed video will allow to generate a larger residual signal and small encoding of this residual signal as bit stream to be sent along with H.264 packet as metadata. **Auto encoders** are more efficient at preserving and predicting the **non-linear residual information** but very slow and inaccurate for entire video, thus the need for generating this hybrid of H.264 and auto encoder that **can fit well into existing framework**. These binary representations can then be encoded in a **lossless manner by using Huffman encoding** and can be sent along with the H.264 stream to the user.

For example: we have 5MB/s link, we can use high compression ratio-based H.264 to get noisy video at 4MB/s and remaining 1MB/s can be taken up by our Huffman encoded meta data. The auto-encoder generates smaller size byte stream for residual information, which is also significant due to higher compression ratio of H.264. Thus, we can send more information in similar bandwidth levels and club it up to generate better looking video at client side.



**APPROACH TO THE SOLUTION**

We will leverage deep learning models for preserving/predicting the residual information of domain specific content. Neural Networks are powerful non-linear function approximators, which can be used to encode the highly nonlinear residual information.

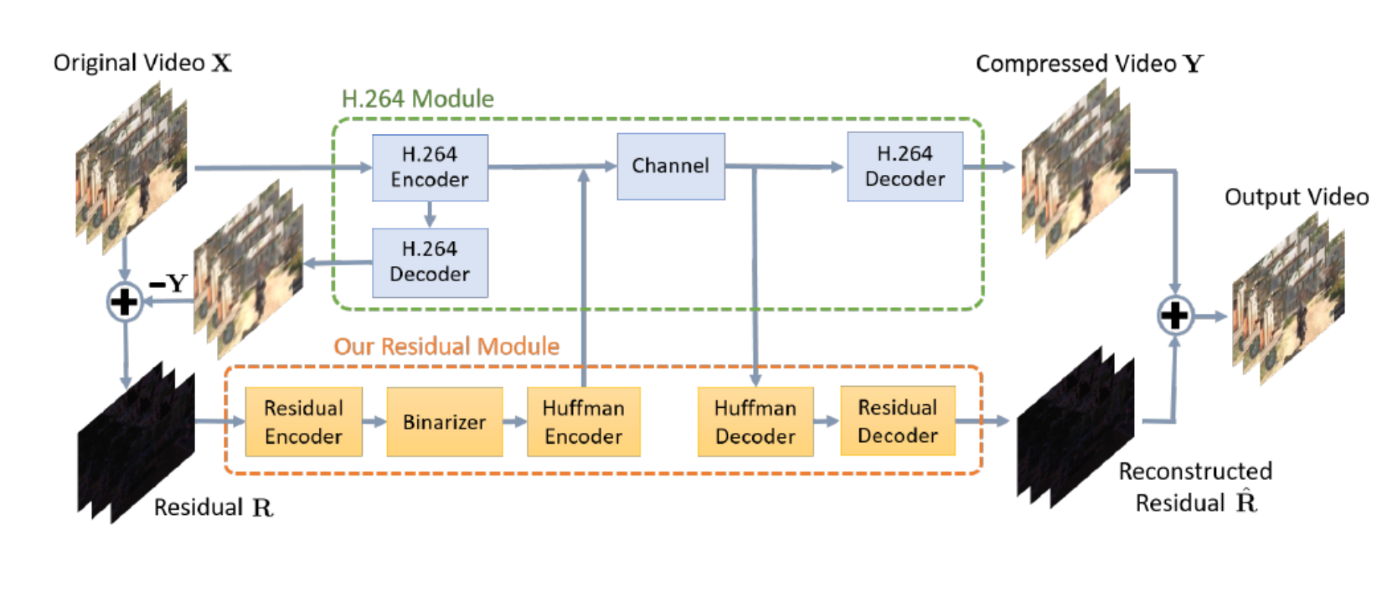
In the video compression pipeline:

1. Apply H.264 to compress videos in a specific domain and train a novel binary autoencoder to encode the resulting residual information frame-by-frame into a binary representation.

2. Apply Huffman coding to encode the binary representations in a lossless manner.

3. The compressed binary representations are sent to the client in the meta data field in the H.264 streaming packet.

4. The decoding of this data can be done and video can be regenerated.



**REFERENCES**

**Research reference:**

* Tsai, {Yi Hsuan} and Liu, {Ming Yu} and Deqing Sun and Yang, {Ming Hsuan} and Jan Kautz; Learning binary residual representations for domain-specific video streaming ; AAAI press.; 32nd AAAI Conference on Artificial Intelligence, AAAI 2018 ; 2018; 7363-7370

[link: <https://research.nvidia.com/sites/default/files/pubs/2018-02_Learning-Binary-Residual/Learning%20Binary%20Residual%20Representations%20for%20Domain-specific%20Video%20Streaming.pdf> ]

**Other References:**

* **NVIDIA YouTube video**: <https://www.youtube.com/watch?v=SCc0SqzkQf4>