TECHNISCHE UNIVERSITÄT
ILMENAU

Faculty of Electrical Engineering and Information Technology
Institute for Media Technology
Audio Visual Technology

MASTER THESIS

Netflix Like Encoding Optimization

Submitted by: Vijaykumar Singh Rana

Major: Media Technology

Advisor: Prof. Dr.-Ing. Alexander Raake

Co-Advisor: M.Sc. Steve Göring

Ilmenau, September 30, 2019

Acknowledgments

optional

Zusammenfassung

maximum of 2400 chars; one paragraph

Abstract

maximum of 2400 chars; one paragraph

Contents

Chapter 1

Introduction

In today's world, media is being offered to the users on a large scale. YouTube[Yt], Netflix[Net] and Prime Video[Pri] are the most popular video hosting websites on internet that provides VOD (video on demand) right now. They have evolved their encoding schemes in a way such that the end user would receive highest possible quality of videos corresponding to the bandwidth they can afford. Providing such data requires various kinds of encoding schemes that can yield the data with best quality for a given bandwidth. These video stream providers mainly aim to reduce the bandwidth, consequently reducing the costs and to satisfy the end user so that they can perceive the best possible quality. One of such schemes that provides streams based on the bandwidth is Dynamic Adaptive Streaming over HTTP (DASH). As the name suggests, it uses a streaming technique that provide contents with highest possible quality based on the network conditions. Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH, is an adaptive bitrate streaming technique that enables high quality streaming of media content over the Internet delivered from conventional HTTP web servers. The content is made available at a variety of different bit rates, i.e., alternative segments encoded at different bit rates covering aligned short intervals of playback time. While the content is being played back by an MPEG-DASH client, the client automatically selects from the alternatives the next segment to download and play based on current network conditions. The client automatically selects the segment with the highest bit rate possible that can be downloaded in time for playback without causing stalls or re-buffering events in the playback. Considering video streaming the main goal of a video stream provider is to reduce bandwidth consequently the cost and to satisfy the end user, so that they are able to perceive the best possible quality. [Das]

YouTube and Netflix are the most popular video hosting websites in the world right now. They provide the users with streaming service which is affordable and provide the best possible quality content based on the fluctuating network conditions. To provide the quality content, YouTube uses chunked video playback. the video is split into smaller parts called chunk. Each of the

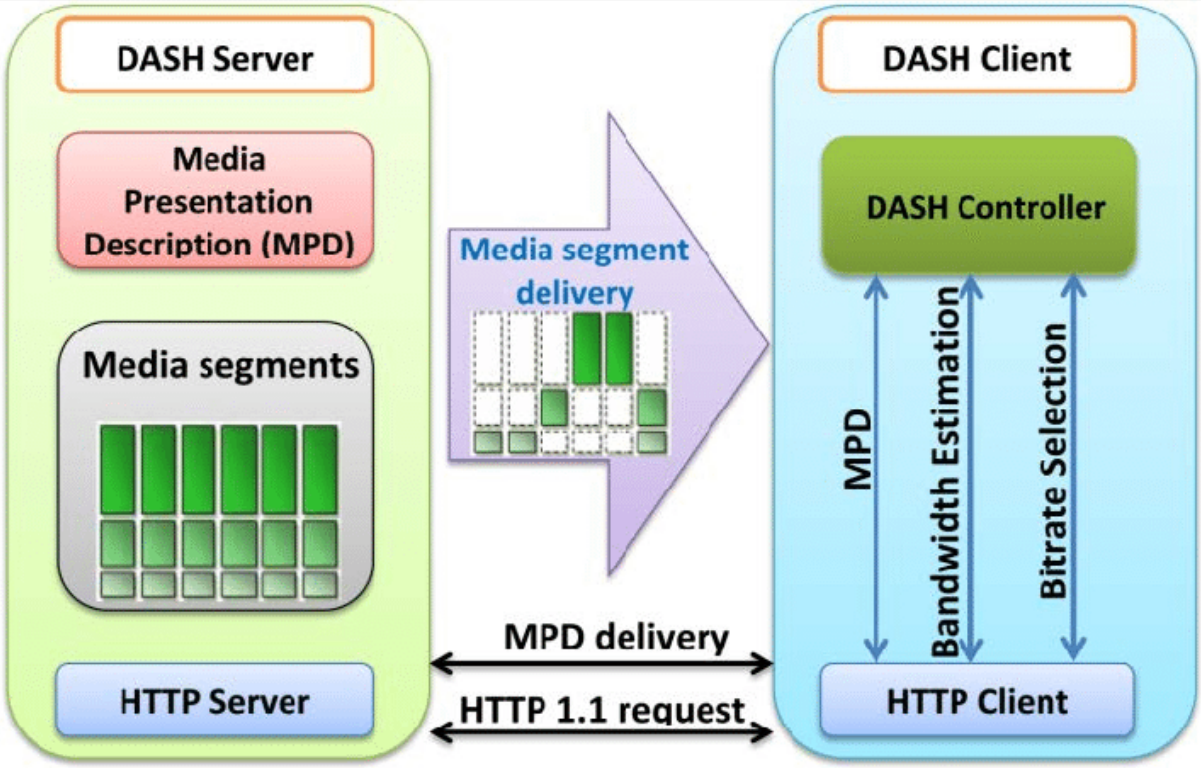

Figure 1.1: The overview of Dynamic Adaptive Streaming over HTTP technique. [VMC17]

chunks are encoded from low quality to high quality. The suitable quality chunk is selected based on network bandwidth. Because of chunking the web browsers are able to manage the smaller pieces of the videos in their dedicated cache for playback. For fluctuating network speeds and optimal user viewing experience, YouTube uses Adaptive Bitrate Streaming (ABS). Figure 1.2 demonstrates how a video stream is selected based on the network condition. When the video starts on YouTube it starts with low quality and then based on the network condition the quality keeps switching. As it can be seen in the Figure 1.2 when the available bandwidth gets higher, the high bitrate stream is then selected. in the next timestamp there appears a network congestion and eventually the quality drops i.e., high bit rate stream is not streamed and based on lower bandwidth low bit rate stream is then selected. Further as the network bandwidth changes, the suitable stream is then selected. YouTube also acquires the screen dimensions and when enough network bandwidth is available then the YouTube would send the stream with the resolution adaptive to screen dimensions.

Netflix also uses the similar adaptive streaming approach to provide streaming service to the users. Netflix also uses granular approach of chunking the video stream based on the shots (a scene with short time period). A shot is a part of a video containing constant background or constant objects which has least movements. Netflix uses techniques of granular optimizations. Netflix uses Per-Title Encode Optimization where a recipe to encode the video optimally is prepared. For every individual video the recipe is produced. The objective is to deliver better

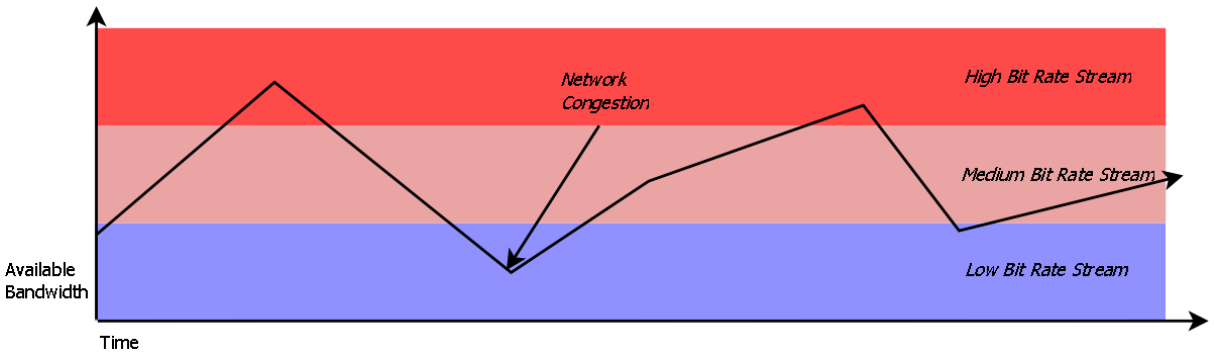

Figure 1.2: Adaptive Streaming [OJ]

experience for users with low network bandwidth. The reason for using per title encoding is because the approach of "one-size-fits-all" fixed bitrate ladder is not a good solution. In many cases the video content differs from video to video. So setting a fixed bitrate for certain resolution is not a good solution and it is inefficient. Figure 1.3 show a table with pairs for bitrate and resolution. Now based on this table 5800 kbps should be good enough for generating a stream of FULL HD resolution(1080p). But certain scenes have dynamic motions in it which leads to camera noise resulting into blockiness artifacts in the noisy areas. On the other hand for generating Full HD stream for a video containing cartoon content 1750 Kbps is more than enough, so utilizing 5800 Kbps is not optimal solution and would leave to bandwidth wastage. [Aar+15].

In case of Netflix, the videos are not just encoded using conventional constant bitrate video encoders. Netflix encodes the videos by employing various encoding schemes and further optimizing the process in order to obtain the high quality data with lowest bitrate. It allows the endusers to stream high quality video with low bitrate and provides the endusers seamless playback of the content. The reason behind achieving this is subjective. From the point of view of an end-user who can afford good network bandwidth would not like to have the blurriness or blockiness in the content. This requires the development of algorithms that could deal with the issues of artifacts in videos encoded with low bitrates. The content of a given video differs with other videos more or less. Assuming a video that contains dynamic and constant scenes. In constant bitrate technique the dynamic scenes may utilize the bitrate efficiently but constant scenes where the background is not changing too much, the bitrate would be more than required thus leading to over utilization of bits and the worst case scenario would be a scene where it is completely dark. The figure 1.4 demonstrates how the quality varies for different video sources. Certain sources never reach high quality even though the bitrate is high enough as 25000 kbps as shown in the figure. 1.4

Also the process of achieving best quality goes further into encoding each of chunks based on the scenes. A scene is a set of frames with similar attributes. A scene could be with fixed

Bitrate (kbps)	Resolution
235	320x240
375	384x288
560	512x384
750	512x384
1050	640x480
1750	720x480
2350	1280x720
3000	1280x720
4300	1920x1080
5800	1920x1080

Figure 1.3: Bitrate-Resolution pairs or Bitrate Ladder [Aar+15]

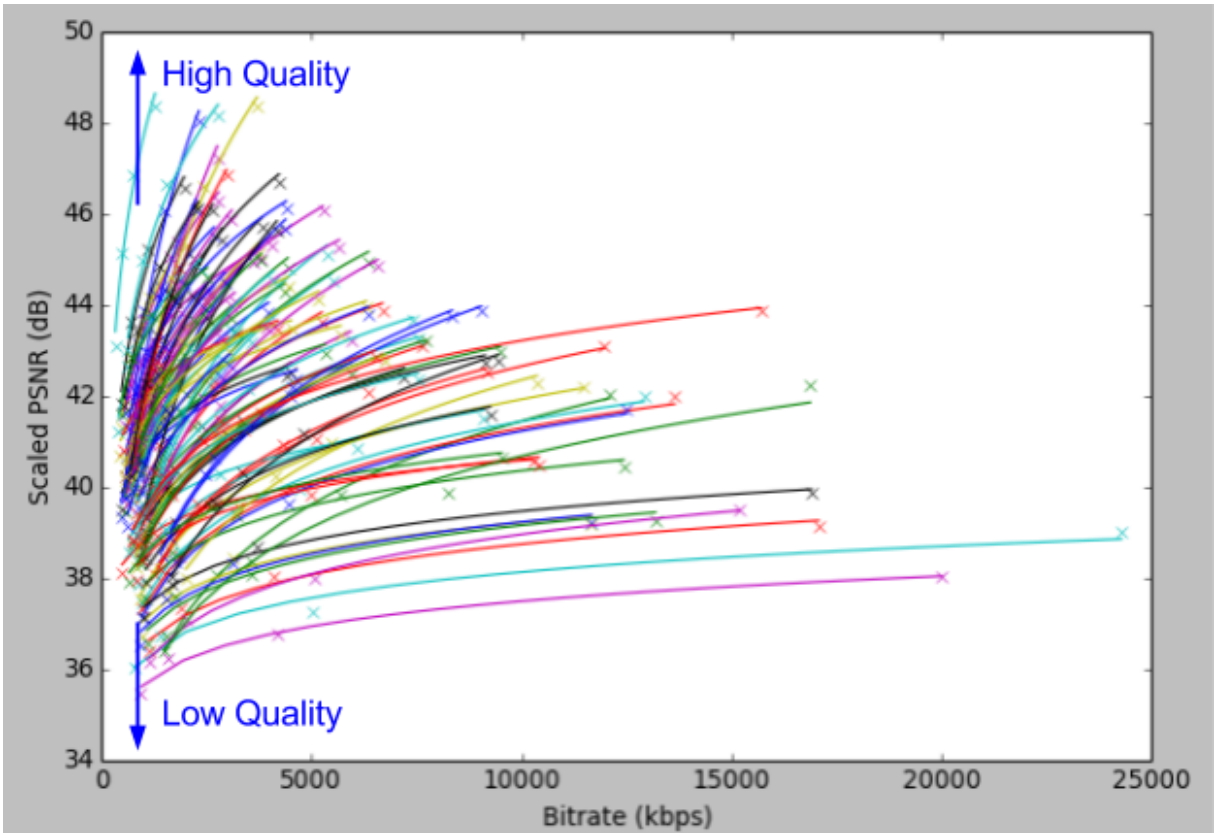

Figure 1.4: 100 randomly sampled sources at 1080p resolution using x264 constant QP (Quantization Parameter) rate control. [Aar+15]

characters in it and a constant background or it could be a dynamic action scene. As each of the scenes have different characteristics so the idea is to encode them individually with different bitrates for different resolutions with highest possible quality.

The metrics like PSNR (Peak Signal to Noise Ratio) [Psn], SSIM (Structural Similarity Metric)[Ssi], VMAF (Video Multi-Method Assessment Fusion) [Blo18] are used for video quality evaluation. The videos encoded for each quality with all possible bitrates are evaluated. Finally those bitrates are chosen which provides high quality amongst the different resolutions on a rate-distortion or rate-quality curve.

1.1 Motivation & Goals

As discussed in the previous section 1 about how each video is processed. This pipeline includes segmenting a video first. Next each of these segments are assessed and divided into scenes or shots. These shots are based on the type of content. These shots have relatively similar kind of frames which makes it easier to encode them with low bitrate compared to encoding a video with heterogeneous content. For this Netflix applies encoding with all possible bitrates for each resolution [Blo17]. Then based on the quality parameters obtained from a rate distortion plot the appropriate bitrate is chosen for each resolution. Netflix applies a brute force technique in order to compute quality parameter for each bitrate for all the resolutions. This requires too much computation power. This leads to research question if there is the way to reduce the overall computation costs and approximate this whole process using optimization techniques. To have such comparison both the approaches will be implemented; a) brute force technique of computing quality parameters by computing quality for each bitrate for different resolutions to generate a rate-distortion curve and b) selecting fewer points from the generated rate-distortion curve and then trying to approximate using mathematical equations e.g., a logarithmic curve.

The motivation here is to do the computation for less number of bitrates and try to obtain other points by the means of any approximation method. Then the overall number of encoding steps would be reduced. The next step would be to compare the resulting points after approximation with points from the brute-force approach. Further analysis may provide error between points obtained using approximation approach and points from the brute-force technique. Based on the error the plausibility of the approach to minimize the overall encoding steps can be analyzed. The smaller steps for resolution will be considered and corresponding to each resolution, the computation of the quality parameters(QP) for all possible bitrate will be done.

The following chapters will include about the fundamentals for the research. Following the fundamentals is the related work to the topic. This will cover all the previous research and

development with regards to topic in this report. After the related work follows the different approaches to design and implement the process chains for

- ▷ Designing the Pipeline for video processing.
- ▷ Brute-Force technique of Netflix to encode a video.
- ▷ Approximation approach, where fewer points are considered from the previous approach and then generating remaining approximated points.

The next part will include analysis and evaluation to calculate the mean squared error between both the approaches. After the evaluation certain conclusions are drawn to validate the results.

Chapter 2

Fundamentals

2.1 Related Work

As described in the previous section 1 about Netflix encoding the videos, firstly using the different encode for each video and then further using granular approach to optimize each of the shots for a the given video. From the point of view of adaptive streamers, the problem was not about encoding a single shot for each video but rather how to do it for collection of all the shots with all the possible bitrate/qualities range that the user would desire the service to receive. This led to development of a framework which considered all the above points from Netflix Technology called Dynamic Optimizer [Kat18]. This framework analyzes an entire video over multiple quality and resolution pairs, finds the best trade off to generate optimal recipe to encode a shot. [Man+18] the idea behind finding the optimal recipe is that a video consists mainly of numerous shots. These shots have little correlation with each other which can be treated an element that can be encoded independently of each other. Each of the shots to be encoded can be considered as visual content which is uniform and homogeneous. This means applying constant or fixed quality for an individual shot fits well. From the human perceptive point of view, humans are least bothered by the coding difference that would occur at the boundaries of adjacent shots. It provides a good deal to adjust the coding resolution and quantization parameter. The decision of choosing optimal resolution and quantization parameter to be used for each of the shots to be encoded, is done on the basis of perceptual video quality metrics like SSIM (Structural Similarity) and VMAF (Video Multimethod Assessment Fusion). Dynamic Optimizer uses cloud computing to generate multiple encodings for a given set of frames from a shot for various resolutions and quantization parameters.[KG18] In this thesis topic the idea is similar to setup a pipeline for video encoding. So in the similar way the videos will be split into shots. Each of the shots will be reproduced for various resolutions. Further for each of the resolutions will be transcoded using several bitrates to generate multiple encodes. Each of the encodes will be

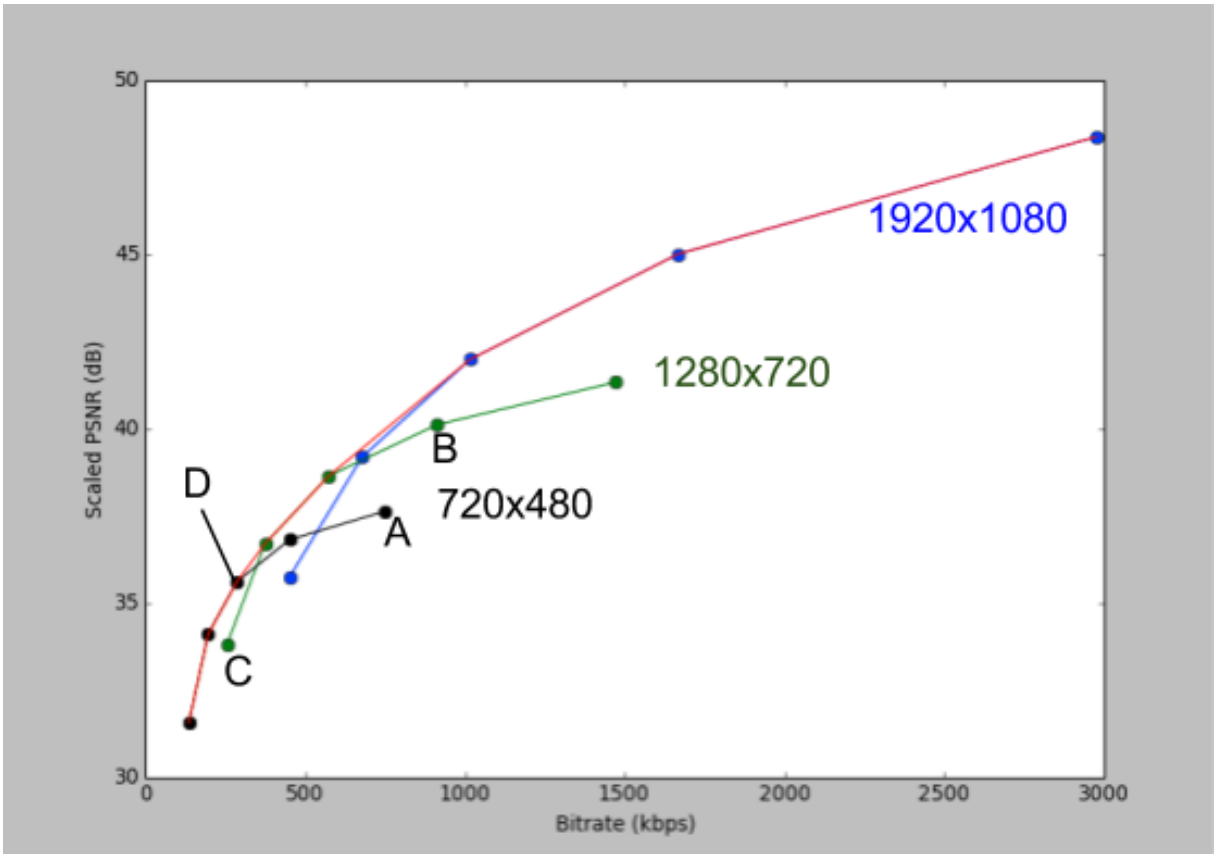

Figure 2.1: Example encodes showing individual R-D curves and convex hull.[Aar+15]

then tested with VMAF for VMAF score (quality parameters) for each bitrate for a given resolution. For each bitrate and its corresponding VMAF score a plot will be generated. The plot would be similar to the Figure 2.1. The labels in the plots are the curves represented in the plot. Curve **A** (in black) is the plot generated using scaled PSNR values as points (R_i, Q_i - set of bitrate points and its corresponding quality score/ parameter for a given resolution) for various bitrates for the resolution 720x480. Similarly curve **B** (in green) is generated using scaled PSNR values for various bitrates for the resolution 1280x720 and finally curve **C** is generated using scaled PSNR values. The red curve **D** is the convex hull obtained over all the curves from the resoltions and its R_i, Q_i points. This curve is used as the basis to select the points for a given bitrate. Although the quality parameters is generated using traditional metric of scaled PSNR in the figure, this thesis topic will include quality parameters generated using VMAF. As discussed in the previous section, the next step will include approximation technique where a fewer points for a curve will be selected and then the rest of the points will be approximated.

The authors from [Sat+19] presented an alternative to per-scene video encoding done by Netflix. They used the CRF (Constant rate factor) encoding recipe for 3 resolutions and for each of those resolutions they computed 4 different (R_i, Q_i) points. For generating quality points

for each of the encodes, they used a full reference metric of PEVQ (Perceptual Evaluation of Video Quality) from OPTICOM [Pev]. The authors tried to explain how the function of (R_i, Q_i) can be used to model the prediction of PEVQ quality scores by using the combination of logistic and negative exponential functions. Using this approach they try to compute all the quality points for all bitrates for a given resolution and finally getting all computations for a given video. Their approach also takes into consideration the tradeoffs between bitrate and quality. In this thesis topic the approach is quite similar. The mathematical approximation would be done using a logarithmic curve instead of inverse exponential function. The fitting of this curve is done with respect to fewer points selected from (R_i, Q_i) for a given resolution.

The pipeline used for the brute force technique by Netflix requires to transcode video at variable bitrates for every resolution of a shot (or a scene) from a given video. The transcoding scheme that is used in this thesis topic is Two-Pass Video Encoding.

2.2 FFMPEG

FFMPEG is an open source cross platform tool which allows for handling audio and video data by the means on encoding, decoding, transcoding, multiplexing, demultiplexing etc. This tool allows for many options for customizing video data by the means of its capabilities. [Ffm]

FFMPEG provides various command line tools, which is described in the table 2.1.

Command Tool	**Usage**
ffmpeg	is used for converting audio or video formats
ffplay	is a command line media player that uses Simple Direct-media Layer(SDL) and FFMPEG libraries
ffprobe	displays text based media information in formats like XML, CSV etc

Table 2.1: FFMPEG Command Line Tools

FFMPEG also provides libraries for various applications described in the table 2.2.

Libraries	Usage
libswresample	provides functions to resample audio
libavresample	provides functions to resample audio from Libav project [tea]
libavcodec	provides native FFmpeg audio/video encoders and decoders
libavformat	provides demuxers and muxers for audio/video container formats
libavutil	includes hash functions like SHA-1, LZO decompressor and Base64 encoder/decoder.
libpostproc	provides functions for old H.263 video postprocessing
libswscale	provides functions for video image scaling and colorspace/pixelformat conversion
libavfilter	is the substitute for vhook which allows the video/audio to be modified or examined between the decoder and the encoder

Table 2.2: FFMPEG Libraries

2.3 Video Quality Metric: Video Multi-Method Assessment Fusion (VMAF)

VMAF was developed by Netflix along with the external researchers which uses machine-learning approach which are fused with existing efficient objective video quality metrics called the elementary metrics.[KG18] [Blo] [Net19]

VMAF is a full reference, perceptual video quality metric. Due to compression and scaling several artifacts result in the video. So VMAF uses the existing perceptual quality scores from multiple quality assessment algorithms to estimate the quality of a given video. This is fused with the support vector machine [Ras17]. As it is full referenced metric the original source and encoded/ compressed source both are taken in to consideration. For multiple encoded versions of a source video the comparison is done using a scaled VMAF score [KG18]. The figure 2.2 shows VMAF system diagram.

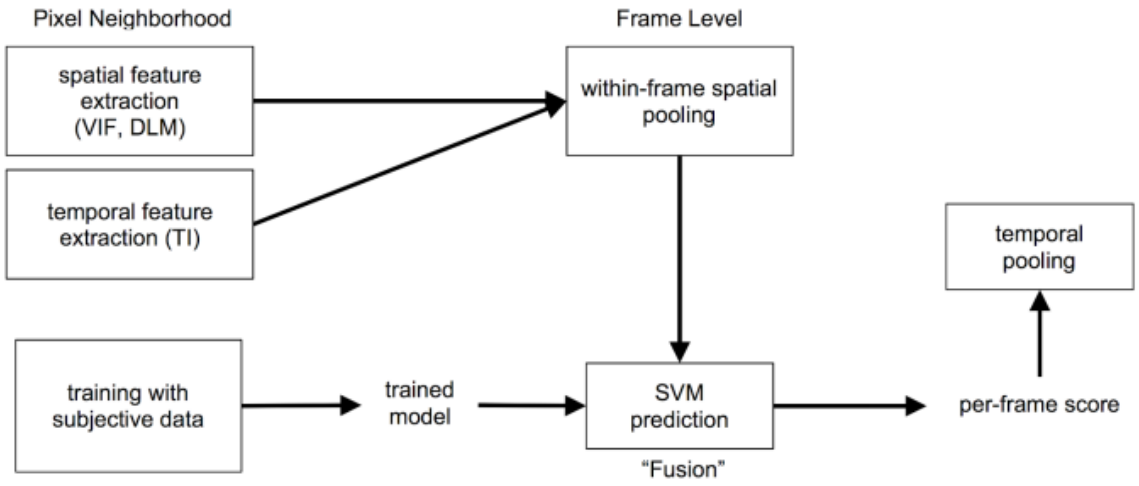

Figure 2.2: VMAF System Diagram [**per title**]

complexity based consistent quality encoding..... Fundamentals / environment and related work: 1/3

- ▷ comment on employed hardware and software
- ▷ describe methods and techniques that build the basis of your work
- ▷ review related work(!)

Chapter 3

Architecture/Implementation

Developed architecture / system design / implementation: 1/3

- ▷ start with a theoretical approach
- ▷ describe the developed system/algorithm/method from a high-level point of view
- ▷ go ahead in presenting your developments in more detail

Chapter 4

Analysis/ Evaluation

Measurement results / analysis / discussion: 1/3

- ▷ whatever you have done, you must comment it, compare it to other systems, evaluate it, using e.g. subjective tests
- ▷ usually, adequate graphs help to show the benefits of your approach
- ▷ caution: each result/graph must be discussed! what's the reason for this peak or why have you ovserved this effect

Chapter 5

Conclusion

Conclusion: 1 page

- ▷ summarize again what your thesis did, but now emphasize more the results, and comparisons
- ▷ write conclusions that can be drawn from the results found and the discussion presented in the paper
- ▷ future work (be very brief, explain what, but not much how)

Bibliography

[Aar+15] Anne Aaron et al. "Per-title encode optimization". In: *The Netflix Techblog* (2015).

[Blo] Netflix Technology Blog. *Toward A Practical Perceptual Video Quality Metric*. https://medium.com/netflix-techblog/toward-a-practical-perceptual-video-quality-metric-653f208b9652.

[Blo17] Netflix Technology Blog. *High Quality Video Encoding at Scale*. 2017. URL: https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746.

[Blo18] Netflix Technology Blog. *VMAF: The Journey Continues*. 2018. URL: https://medium.com/netflix-techblog/vmaf-the-journey-continues-44b51ee9ed12.

[Das] *Dynamic Adaptive Streaming over HTTP*. 2019. URL: https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP.

[Ffm] *FFmpeg*. URL: https://ffmpeg.org/.

[Kat18] Ioannis Katsavounidis. *Dynamic optimizer - a perceptual video encoding optimization framework*. 2018. URL: https://medium.com/netflix-techblog/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f.

[KG18] Ioannis Katsavounidis and Liwei Guo. "Video codec comparison using the dynamic optimizer framework". In: *Applications of Digital Image Processing XLI*. Vol. 10752. International Society for Optics and Photonics. 2018, 107520Q.

[Man+18] Megha Manohara et al. *Optimized shot-based encodes: Now Streaming!* 2018. URL: https://medium.com/netflix-techblog/optimized-shot-based-encodes-now-streaming-4b9464204830.

[Net] *NETFLIX*. https://www.netflix.com/.

[Net19] Netflix. *Netflix/vmaf*. 2019. URL: https://github.com/Netflix/vmaf.

[OJ] Martin Ombura Jr. *How YouTube handles streaming 4,000,000,000+ daily videos without a hitch.* https://medium.com/@martinomburajr/how-youtube-handles-streaming-4-000-000-000-daily-videos-without-a-hitch-8542741e957a.

[Pev] *the Standard for Perceptual Evaluation of Video Quality*. URL: http://www.pevq.com/.

[Pri] *PrimeVideo*. https://www.primevideo.com/.

[Psn] *Peak signal-to-noise ratio*. 2019. URL: https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio.

[Ras17] Reza Rassool. "VMAF reproducibility: Validating a perceptual practical video quality metric". In: *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*. IEEE. 2017, pp. 1–2.

[Sat+19] Shahid Mahmood Satti et al. "Low Complexity" Smart" Per-Scene Video Encoding". In: *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE. 2019, pp. 1–3.

[Ssi] *Structural similarity*. 2019. URL: https://en.wikipedia.org/wiki/Structural_similarity.

[tea] Libav team. *Libav - Open source audio and video processing tools*. https://www.libav.org/.

[VMC17] Van-Huy Vu, Ibrahim Mashal, and Tein-Yaw Chung. "A Novel Bandwidth Estimation Method Based on MACD for DASH." In: *KSII Transactions on Internet & Information Systems* 11.3 (2017).

[Yt] *YouTube*. https://www.youtube.com/.

List of Figures

List of Tables

Appendix A

AppendixExample

Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.
I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly. Ilmenau,

_____

Todo list