



universität
wien

MASTERARBEIT

Titel der Masterarbeit

Video Quality Measurement Based On Objective
Quality Metrics

verfasst von

Doris Isabella Braunoeder, BSc

angestrebter akademischer Grad

Diplom-Ingenieurin (Dipl.-Ing.)

Wien, 2014

Studienkennzahl: A 066 935
Studienrichtung: Medieninformatik
Betreuer: Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs

Abstract

The distribution of multimedia content, especially videos and films over the internet, has been growing rapidly over the past years. The number of professional streaming services has been rising for years, while platforms for user generated content, as for example YouTube, still enjoy great popularity. Consumer expectations regarding the quality of the videos are difficult to satisfy. Therefore, it is absolutely essential for providers to choose the right compression parameters and monitor the perceived quality of the compressed video files. In the last years the development of video quality assessment methods on how to meet different requirements has been part of many research studies.

The primary purpose of this study is the identification of critical encoder settings with a focus on mobile streaming. The goal of the experiment is to investigate the impact of different encoder settings such as varying bit rates, resolutions, buffer sizes or group of picture lengths on the perceived video quality.

Data for this study were collected during an experiment, in which 22 clips with seven to 14 seconds length and a resolution between 320p and 1080p were encoded with different encoder settings. All of the 23100 hypothetical reference circuits were performed with the H.264/AVC codec, because it is freely available and one of the most frequently used codecs for video compression. For measuring the perceived video quality of the encoded files the objective full reference evaluation methods PSNR, SSIM and VQM were used.

The results of the study showed that the reduction of the bit rate does have a significant impact on the video quality, while other tested parameter settings are not limiting factors for video quality. The study showed that the quality of videos with bit rates over 400 kBit/s are always rated with good video quality, while bit rates of 200 kBit/s or below are not recommendable. It was concluded that the content of each video does have a major impact on the performance of an encoder and therefore also on the perceived video quality. The results of the three objective algorithms PSNR, SSIM and VQM are comparable for overview results - they review the files with similar quality - but on a closer look each of the metrics show different results.

Zusammenfassung

Die Verbreitung von multimedialen Inhalten, vor allem aber von Video- und Filmmaterial über das Internet hat in den letzten Jahren beständig zugenommen. Die Anzahl der Video-Streaming Dienste wächst, während sich Plattformen für die Verbreitung von nutzergenerierten Inhalten weiterhin höchster Beliebtheit erfreuen. Die Benutzer erwarten Videos in bestmöglicher Qualität, unabhängig von diversen Einschränkungen wie zum Beispiel der Bandbreite. Für Provider ist es deshalb um so wichtiger, die richtigen Parameter bei der Videokompression zu verwenden und die Qualität der produzierten Videos ständig zu überwachen. In den letzten Jahren gab es viele Forschungsprojekte mit dem Ziel neuere oder bessere Algorithmen zur Qualitätsbestimmung in Videos zu entwickeln.

Das primäre Ziel dieser Studie ist die Identifikation der kritischen Encoder-Einstellungen mit dem Fokus auf Streaming über mobile Endgeräte. Das Ziel des Experimentes ist es, den Einfluss von unterschiedlichen Encoder-Parametern auf die wahrgenommene Qualität zu untersuchen.

Die Daten für diese Studie wurden durch ein Experiment mit 22 Videoclips, die alle zwischen sieben und 14 Sekunden lang sind, gewonnen. Die Videos mit einer Auflösung zwischen 320p und 1080p wurden mit unterschiedlichen Encoder-Einstellungen encodiert. Durch die Kombination verschiedener Einstellungen ergeben sich 23100 theoretische Möglichkeiten der Encodierung. Als Basis für das Experiment diente der H.264/AVC Encoder, welcher frei verfügbar und außerdem heute auch noch einer der am häufigsten verwendeten Encoder ist. Um die Qualität der encodierten Videos zu messen wurden objektive Methoden wie PSNR, SSIM und VQM zwischen Referenzvideo und encodierten Video berechnet.

Die Ergebnisse der Studie zeigten, dass nur eine Reduktion der Bitrate die Videoqualität signifikant beeinflusst, während GOP Länge, Buffergröße und Auflösung keinen Einfluss auf die Videoqualität nahmen. Den Ergebnisse zufolge erhalten Videos mit einer Bitrate über 400 kBit/s das Prädikat sehr gut hinsichtlich ihrer Qualität, während Videos, die mit einer Bitrate von 200 KBit/s und darunter encodiert wurden, schlechtere Ergebnisse erzielten. Der Inhalt eines Videos hat maßgebenden Anteil an der Leistung der Encoder und der daraus folgenden Videoqualität des Endproduktes. Die Ergebnisse der drei Algorithmen PSNR, SSIM und VQM liefern auf den ersten Blick ein ähnliches Ergebnis, detaillierte Einblicke zeigen dann aber, dass die Metriken nicht vergleichbar sind.

Acknowledgements

First, I would like to thank my supervisor Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs for his support and guidance along this way. My gratitude also goes to Dipl.-Ing. Werner Robitza for his thoughts, ideas and support. Special thanks go to my fellow students Christina, Silvia and Evelyn for many useful discussions and support through all the years of study.

I am especially grateful to my parents, for always supporting me and giving me the opportunity to attend university. Finally I would like to say thank you to Michael for his patience and support through all the years.

Contents

Abstract	iii
Zusammenfassung	v
Acknowledgements	vii
Contents	ix
List of Figures	x
List of Tables	x
Listings	xi
1 Introduction	1
1.1 Motivation	2
1.2 Outline	3
2 Related Work	5
2.1 Video Quality Metrics	5
2.2 Video Quality Experts Group (VQEG)	6
2.3 Video Quality Analysis Databases	6
2.4 Performance Studies	7
3 Background	11
3.1 Video Streaming	11
3.1.1 Technical Background	12
3.1.2 Protocols	13
3.1.3 Problems in Streaming	13
3.2 Quality of Service	14
3.3 Quality of Experience	15
3.3.1 Subjective QoE	15
3.3.2 Objective QoE	17
3.4 Video Encoding	22
3.4.1 The Need for Compression	22
3.4.2 What is Compression?	23

3.4.3	Requirements of a Compression System	24
3.4.4	Basics of Compression	24
3.4.5	Video Compression Standards	28
3.4.6	Video Transcoding	32
3.5	Compression Artifacts	34
3.6	Human Visual System	34
4	Experimental Setup	37
4.1	Hardware	37
4.2	Software	38
4.3	Use Cases	39
4.4	Encoder Settings	39
4.5	Footage	41
4.6	Encoding Process	42
4.7	Comparison Process	44
4.8	Problems	46
5	Experimental Results	47
5.1	Analysis of Encoding Duration	47
5.2	Analysis of Resulting File Size	50
5.3	Analysis of Objective Evaluation Results	55
5.3.1	Impact of Bit Rate	55
5.3.2	Impact of GOP Length	59
5.3.3	Impact of Buffer size	64
5.3.4	Impact of Resolution	68
5.3.5	Impact of Content Type	70
5.3.6	ANOVA Validation	71
5.4	Summary	73
6	Conclusion	75
6.1	Future Work	77
A	Curriculum Vitae	79
	Bibliography	81

List of Figures

3.1	Simple example of a video streaming architecture	12
3.2	FR video quality framework	18
3.3	Process diagram of SSIM calculation	19
3.4	RR video quality framework	21
3.5	NR video quality framework	22
3.6	Block diagram of a video encoder	23
3.7	Video compression architecture	25
3.8	Macroblock structure	26
3.9	Typical GOP structure	27
3.10	Block diagram of a H.264/AVC encoder	30
3.11	H.264/AVC macroblock partitions	31
3.12	H.264/AVC profiles	32
3.13	Transcoding operations	32
3.14	Transcoding classification	33
3.15	The human eye	35
3.16	Visual pathway in the human brain	36
5.1	Duration of encoding 1 st pass	48
5.2	File size of the input footage	49
5.3	Duration of encoding 2 nd pass	50
5.4	Impact of bit rate reduction on the file size	51
5.5	Impact of resolution reduction on the file size	52
5.6	Impact of variation in the GOP length on the file size	53
5.7	Impact of different buffer sizes on the file size	54
5.8	PSNR for different bit rate values	56
5.9	PSNR for different bit rate values grouped by the input files	57
5.10	SSIM for different bit rate values	58
5.11	SSIM for different bit rate values grouped by the input files	58
5.12	VQM for different bit rate values	59
5.13	PSNR for different GOP lengths	60
5.14	PSNR showing the influence of the GOP length	61
5.15	PSNR for different GOP lengths, Bitrate: 100 kBit/s	62
5.16	PSNR for different GOP lengths, Bitrate: 200 kBit/s	62
5.17	PSNR for different GOP lengths, Bitrate: 400 kBit/s	63
5.18	SSIM for different GOP lengths	63

5.19	VQM for different GOP lengths	64
5.20	PSNR for different buffer sizes	65
5.21	SSIM for different buffer sizes	66
5.22	VQM for different buffer sizes	66
5.23	SSIM for different buffer sizes grouped by the bit rate	67
5.24	PSNR for resolution	68
5.25	SSIM for resolution	69
5.26	VQM for resolution	69
5.27	PSNR grouped by resolution	70
5.28	PSNR for each file	71

List of Tables

3.1	Time line of compression standards	29
4.1	Hardware setup	38
4.2	Video details of the source footage	42
4.3	Overview and description of the source footage	43
5.1	Mapping PSNR, SSIM and VQM to subjective MOS values	55
5.2	ANOVA on PSNR with bit rate, buffer size, GOP length, resolution and content type	72
5.3	ANOVA on SSIM with bit rate, buffer size, GOP length, resolution and content type	72
5.4	ANOVA on VQM with bit rate, buffer size, GOP length, resolution and content type	72
6.1	HRC of the experiment	76

Listings

4.1	Sample AVS input file	42
4.2	Sample x264 encoder call 1 st pass resolution: 1080p	42
4.3	Sample x264 encoder call 2 nd pass resolution: 1080p	44
4.4	Sample x264 Encoder call 1 st pass resolution: 360p	44
4.5	Sample x264 Encoder Call 2 nd pass resolution: 360p	44
4.6	Sample .stat file	44
4.7	Sample x264 encoder output	45
4.8	Sample MSU Tool statement for VQM calculation	45
4.9	Sample MSU Tool output for VQM calculatiuon	45

Chapter 1.

Introduction

The number of video-on-demand (VOD) platforms has been rising in the last years. To name some well-known services: YouTube, Netflix, Hulu, Amazon Instant Video and iTunes Video. According to Cisco's "Global Mobile Data Traffic Forecast Update" [1] the mobile data traffic reached 1.5 exabytes per month at the end of 2013. Mobile video consumption accounts for 53% of this data traffic. Cisco predicts an increase in monthly mobile traffic of up to 15 exabytes by the end of 2018 - more than two-thirds of this overall mobile traffic will be produced by mobile video consumption. There is no doubt that video streaming will continue to dominate data traffic also in the future.

The extensive use of video streaming on mobile phones correlates with the rising number of users of mobile devices. Cisco [1] reports that more than half a billion mobile devices were added in 2013, so the number of devices increased to 7 billion and will grow up to 10.2 billion by 2018. On average, a smart phone generated 29 times more traffic than a basic cell-phone. It is clear that with the rising number of smart devices the amount of user generated video content on social networks and download sites increases. According to [2] more than 6 billion hours of videos are watched per month on YouTube. Even more incredible is the fact that the video portfolio is increasing every minute by about 100 hours.

The Global Internet Phenomena Report [3] reported that real-time entertainment was the biggest network traffic source in North America in the first half of 2014. According to that report about 30% of the traffic are generated by Netflix and about 20% by other streaming or entertainment services like YouTube, iTunes, Amazon Instant Video or Hulu. These figures are different to those from Europe where only about 3% of the traffic is generated by Netflix, owing to the fact that the service was only officially launched in the United Kingdom and Ireland at the time of the experiment. Reviewing the report in a year would probably show a completely different result for Europe, since Netflix has now been launched officially in many other locations in Central Europe. YouTube remains the largest single source of real-time entertainment traffic.

With the introduction of high-capacity network connections such as 3G, 4G or LTE in the last years, the usage of VOD applications on mobile devices has increased and the

new distribution networks apart from television, cinema and blue-ray or DVD are finally established on the market. According to Cisco's study [1] a fourth-generation network already generates 30% of mobile traffic. By 2018, 15% of all devices will be ready to handle 4G connections and half of the mobile traffic will be generated through high speed connections.

Streaming services are still challenged by the need to provide all users with videos in a steady high quality despite several limitations such as low bandwidth, lower processor capability or smaller screen resolutions on mobile phones and tablet PCs on the user's side. Therefore, it is absolutely essential to choose the right settings regarding encoding and processing, depending on the user's device and to monitor the quality of the compressed video files as well as the user-perceived quality-of-experience (QoE). On the other hand, network providers are also challenged by the need to design networks that support these services with adequate quality-of-service (QoS).

Quality is a broad concept - it has to be considered that every human being defines quality according to his/her personal interests, different expectations regarding quality and content, experience and viewing conditions for example lighting, display type and quality. Therefore, it is not easy to evaluate what is good and what is bad - which makes this field so interesting and challenging. According to [4] quality is defined as:

“The totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs.”

The evaluation of video quality is classified into two methods: subjective and objective analysis. As humans are always the end-users of videos, subjective analysis means that a large number of human observers review the video quality and thereby generate the Mean Opinion Score (MOS), which is the most reliable form of video quality measurement. Unfortunately these reviews are expensive, time consuming and not possible for example in real-time applications like streaming and for large amounts of data. Because of all these disadvantages objective quality metrics were developed. For measuring the objective video quality, which should cover all aspects of the Human Visual System (HVS) there are three basic methods, depending on the amount of information on the reference video available: Full-Reference (FR), No-Reference (NR), Reduced-Reference (RR). Typical Full-Reference metrics are for example the Mean-Squared-Error (MSE), Structural-Similarity (SSIM) and Peak-Signal-to-Noise-Ratio (PSNR).

1.1. Motivation

A central aspect of this thesis is the identification of critical encoder settings focusing on mobile streaming as part of an experiment. Each encoding process goes along with a loss of information. The goal is to find out which settings result in image artifacts

and too much information loss. Therefore I will encode video clips with various encoder settings to obtain information on how certain factors, such as different resolutions, bit rates, buffer sizes or group-of-picture (GOP) lengths may impact the video quality. I will focus on the H.264/AVC video coding standard, which is freely available.

To evaluate the quality of the resulting files I will focus on the objective FR metrics PSNR, SSIM and VQM, since a subjective evaluation is not possible with over 20000 generated files. These values will be calculated by an automated process with the help of external tools, which will make it easier to interrelate the critical settings and get indications through plots and figures on for example which of the encoder settings result in the biggest quality loss or the best relation between quality and file size.

1.2. Outline

This thesis is structured into 6 chapters. Chapter 1 “Introduction” provides some major background information and describes the idea and motivation behind this thesis.

Chapter 2 “Related Work” gives an overview of related publications in this field and tries to reveal the differences between existing studies and this thesis.

Chapter 3 “Background” covers the theoretical background of this master thesis with an overview of the video streaming paradigm. Different metrics for subjective and objective video quality assessment are presented and an insight into the field of video compression with focus on H.264/AVC encoding and the resulting artifacts while compressing is provided. To understand how compression works a short overview of the Human Visual System is given to illustrate the process of human perception.

The experimental part is described in chapter 4 “Experimental Setup”, providing detailed information about the software tools used, the base footage, the design of the experiment with all the encoder settings used, insights into the encoding process with sample statements, details about the quality comparison process as well as a short description of problems that occurred during the whole process.

Chapter 5 “Experimental Results” contains the evaluation of the data gathered during the experiment, illustrated with plots and figures, as well as a recommendation on which encoding parameters should be used and which not.

Finally, chapter 6 “Conclusion” summarizes the thesis and gives an outlook on future work.

Chapter 2.

Related Work

In this chapter an excerpt of the most relevant literature in the context of this thesis is presented. The first section 2.1 lists the most useful literature regarding the metrics available and relevant for this experiment. The VQEG as one of the biggest institutions in the field of video quality assessment is introduced in section 2.2. The next section 2.3 gives a short overview of common freely available databases for evaluation and benchmarking objective methods. Different outcomes of previous performance studies on encoding settings and quality metrics are mentioned in section 2.4.

2.1. Video Quality Metrics

In the past years a lot of research in the context of video quality measurement, especially the development of new objective video quality metrics, has been done. A description with details on the calculation of the metrics available can be found in chapter 3. As I have chosen SSIM, PSNR and VQM as metric for the evaluation of the experiment, the following three works are of importance for this thesis.

Wang et. al [5] developed an image quality assessment method based on the extraction of the structural similarity of two images. The goal is to estimate the degradation of structural similarity based on the statistical properties of local information between a reference and a distorted video file. The **SSIM** is very easy to implement and very fast to calculate and is very often used as a metric for video quality evaluation. SSIM calculation is the basis of many extensions developed in the years after its publication - some of them are discussed in chapter 3 section 3.3.2 [5].

The **Peak-Signal-to-Noise-Ratio (PSNR)** is a very popular metric for video processing as it is fast to calculate and very easy to understand. The metric is based on the MSE (Mean-Squared-Error) calculation between a pair of reference and encoded video frames, both calculation methods are introduced in [6]. The outcome is the similarity level of the luminance in the scale of dB.

The **Video Quality Metric (VQM)** developed by the NTIA in [7] is an objective reduced-reference method that predicts subjective quality very well. The calculation is based on different impairment filters for example blurring, jerkiness, noise and block distortion. All these parameters are combined into one measure. The performance in the VQEG validation phase II [8] was outstanding so the metric was adopted as an international standard. As the reference file is not necessary in its entirety, this metric can be used in real-time quality measurements. The developers of the VQM published a very user-friendly tool for VQM calculation free of charge at ¹.

2.2. Video Quality Experts Group (VQEG)

The VQEG ² was founded in 1997 with the goal of validating and testing objective perceptual quality metrics. The group consists of experts in the field of video quality assessment from industry, universities, and international organizations. Different publications of the VQEG were used as a basis for many other studies and experiments. Many studies of the VQEG resulted in International Telecommunication Union (ITU) standardization. Developing more accurate metrics that are easy to calculate is currently an area of active research. The VQEG continuously enhances available metrics and develops new metrics for objective video measurement, which will be validated through extensive tests.

2.3. Video Quality Analysis Databases

For measuring the performance of a newly developed quality metric or the comparison of different encoder settings, a subjective study is necessary to validate the results. As the subjective experiments do have several limitations, which are discussed in chapter 3 section 3.3.1, the existence of predefined databases helps a lot in bypassing the step of finding human reviewers and building a suitable laboratory for evaluation. In the following only the most relevant databases with relevance to today's testing areas are mentioned. The length of the videos in all the databases is around 10 seconds.

The **LIVE** database [9] developed by the Laboratory for Image and Video Engineering contains 10 reference videos with a resolution of 768×432 pixels. Each of these reference files has 15 impaired clips based on either the MPEG-2 or H.264/AVC compression algorithm. This results in 150 test files. A single stimulus continuous test was performed with 38 subjects and the difference mean opinion scores (DMOS) were recorded. The database is freely available and can be downloaded from ³. The LIVE database is one of

¹<http://www.its.bldrdoc.gov/resources/video-quality-research/software.aspx>

²<http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx>

³http://live.ece.utexas.edu/research/quality/live_video.html

the most frequently used references for evaluation and benchmarking in this field. An advanced version, called the **LIVE Mobile Video Quality Database** was published in 2012 ⁴ and contains reference HD content.

The VQEG published a database ⁵ within the **VQEG FRTV Phase 1** program in 2000, containing 20 video sequences, each with 16 distortion types in a resolution of 720×486 pixels, and is the oldest video database available. Unfortunately, the coding is only based on MPEG-2, which is nowadays not longer standard. The database also contains the subjective evaluation by a DSCQS method of 287 subjects. In 2010 the company released a database for testing HD content, the so called **VQEG HDTV database** ⁶, containing 6 videos with about 9 distortions each. Test conditions include MPEG-2 and H.264/AVC compression.

The **IRCCyN/IVC 1080i database** ⁷ contains 24 high quality video sequences in a resolution of 1080i, each video was distorted with different H.264/AVC compression methods (e.g. reduction of the bit rate). To summarize, the database contains 192 processed HD video sequences with corresponding MOS values.

Another database with HD content is the **IVP Subjective Quality Video Database** ⁸, which contains 10 source videos in HD resolution and 128 files encoded with either MPEG-2, Dirac wavelet, and H.264/AVC codecs as well as H.264/AVC streams affected by simulated packet loss.

Different databases, such as the **EP FL/PolIMI video database** ⁹ or the **Poly@NYU Video Quality Database** ¹⁰ contain only videos in CIF and QCIF resolution, which is not state-of-the-art nowadays and not useful for further experiments but were used in different previous works.

2.4. Performance Studies

In the research field of video quality assessment many studies have been published in the past years. The PSNR and SSIM values were criticized in some studies, as they do not correlate well with the HVS, while on the other hand different papers were published, in which the authors prove that these values can still be used for quality estimation under different circumstances. In the following I would like to introduce the most relevant papers regarding performance studies.

⁴http://live.ece.utexas.edu/research/quality/live_mobile_video.html

⁵<http://www.its.bldrdoc.gov/vqeg/projects/frtv-phase-i/frtv-phase-i.aspx>

⁶<http://www.its.bldrdoc.gov/vqeg/projects/hdtv/hdtv.aspx>

⁷<http://130.66.64.103/spip.php?article541>

⁸<http://ivp.ee.cuhk.edu.hk/research/database/subjective/>

⁹<http://vqa.com.polimi.it/>

¹⁰<http://vision.poly.edu/index.html/index.php>

In [10] the authors tested for scenarios in which the PSNR is a reliable indication of video quality and when it correlates with subjective evaluation. The result is that the PSNR performs better for evaluation of videos with a fixed content and codec and no variance in the frame rate. This means that correlation is much better if each video scene is considered separately.

Another study in [11] examined the performance of the SSIM, PSNR and MS-SSIM in digital cinema environments. They compared subjective values with the results of the objective quality analysis of the same footage and found out that the PSNR has the highest correlation with subjective data followed by the MS-SSIM and then SSIM in a 2K video environment.

The authors of [12] developed a framework for QoE evaluation with the FR metrics SSIM and VQM. The goal was to identify the main factors that influence the perceived video quality. For this purpose the influence of content type, video resolution, frame rate and the scaling method on the QoE were examined. The base footage was encoded with the H.264/SVC codec with different frame rates and resolutions. The study showed that, regardless of the content, the higher the resolution the better the SSIM and VQM, while the VQM estimates the QoE better than the SSIM. These results can also be mapped on reducing the frame rate of the videos. The experiment also showed that it is absolutely essential to consider the content type for controlling the QoE. Similar results are shown in [13] with the help of the FR metrics PSNR and SSIM. Zinner et. al figured out that the reduction of the file size to less than half still results in a clip with decent video quality.

A quite similar experiment was conducted in [14] with much smaller video resolutions and only the calculation of the PSNR metric for comparison. The results showed that a lower resolution results in lower PSNR values for all bit rates.

In [15] the authors generated two databases each of them containing 90 video clips with distortions, carried out subjective tests on these test files and afterwards compared the performance of different objective quality metrics (PSNR, VSNR, SSIM, MS-SSIM, VQM, SMOVIE, TMOVIE, MOVIE and FMSE) to the subjective results. All files were encoded with the H.264/AVC and Xvid v.1.1.0 codec based on the MPEG-4 Part 2 specification and different bit rates from 73 to 1737 kBit/s. The results showed that resolution as well as content and distortion type have a significant influence on the performance of the objective VQA metrics. E.g. VQM and TMOVIE seem to be less adaptable to the change of resolution. Another outcome was that the metrics MOVIE and FMSE achieve very good results in testing the distortions.

The authors of paper [16] used various watermarking techniques to add logos in a video and used the metrics PSNR, VQM and SSIM to evaluate the video quality. The experiment was conducted in three categories, reference and encoded video are nearly the same, reference and encoded video are different and the encoded video contained watermarks. The overall findings are that the SSIM is more accurate and closer to the

results obtained through subjective studies but is in general not the metric of choice for finding distortions.

Paper [17] summarizes and classifies the quality assessment methods in a very clear structure. Furthermore, the authors ran a performance evaluation of recent objective metrics (SSIM, VSSIM, MS-SSIM, VIF, VQM) on the LIVE database and found out that the VQM, MS-SSIM and MOVIE correlate very well with subjective results in this case.

The paper [18] presented an encoder parameter recommendation for adaptive streaming applications to achieve the best video quality. For this purpose the authors transcoded a H.264/AVC main profile HD source video clip with about 240 seconds length containing different motion and texture content to different H.264/AVC baseline profiles with varying resolution, frame rate and bit rate. The results of the experiments showed that the compression artifacts in lower resolution videos were much more perceptible than those in higher resolution encoded with low bit rates. From this it can be concluded that the perceived video quality is better when low resolution videos are encoded with low bit rates and are up-scaled to the desired resolution suitable for a mobile device. However, a resolution of 35% smaller than the desired size is not useful.

Only a few studies and experiments have been done on the HEVC codec until now. The authors of [19] analyzed the differences in the performance of the metrics PSNR, SSIM, MS-SSIM and VQM between H.264/AVC and HEVC. The findings are that the PSNR performs very well on HEVC while there are difficulties with VQM. Unfortunately, compared to the subjective results no metric seems to work perfectly with this encoder and content type.

The authors of [20] summarized all available image and video quality metrics. They also benchmarked the different metrics with the LIVE Video Quality Database. The results of this experiment showed that the ST-MAD, MOVIE and VQM metrics correlate very well with subjective evaluation results. The authors also give an outlook on future metrics, what is still needed and how the research field might develop in the future.

In [21] a learning system was developed that can automatically extract the perceived quality from a H.264/AVC encoded video. To achieve this goal the authors encoded a large set of H.264/AVC video clips with different source rates and calculated the SSIM index for each video and trained different machine learning algorithms with a data set containing frame sizes of different GOPs. The validation showed that the calculation works very well. The method might be extended for further metrics, which unfortunately needs a larger training data set to provide enough information to extract the data.

[22] presented the results of a subjective and objective quality evaluation study. PSNR, VQM, SSIM and locally averaged PSNR were used as objective metrics. The authors compared source sequences in CIF resolution encoded with the H.264/AVC as well as the MPEG-4 Part 2 codec with several bit rates. The outcome of this study was that first

the H.264/AVC codec has a much better performance than the older MPEG-4 Part 2 standard and second that the metrics local PSNR and SSIM show a good correspondence with subjective evaluation results. This, in my opinion is very surprising because the VQM is better rated in other experiments, for example in [20], [12] or [7].

The authors of [9] evaluated the performance of several objective video quality measurement methods available (PSNR, SSIM, MS-SSIM, VSNR, VQM and MOVIE) with a subjective study of about 38 human subjects and came to the result that the MOVIE index was the best performing algorithm followed by the VQM and MS-SSIM index. The authors also prove that the often used PSNR values do not correlate very well with subjective results under different circumstances. A very similar result can be found in [23] for non-HD content but could not be proven for HD content.

The experiments in [24] once again showed that the results of objective metrics depend on the content and that their performance cannot be compared. The authors used the VQM metric for quality evaluation of H.264/AVC encoded files with dropped P-frames. The performance of the VQM does not depend on the resolution or used bit rate but rather on the number of lost P-frames and the video content.

The challenges in developing a new metric are summarized in [25]. The authors also discuss common errors and show that a wrong verification process affects the results of a metric. To avoid errors five steps that should be followed in the verification process of video quality metrics are presented. Using a large database that covers a wide range of content with diverse distortion types is strongly recommended. The goal is to reach a broad range of video quality from excellent to bad with different HRCs to cover authentic scenarios. A second very important step is the subjective evaluation of the results. For that reason the tests must be conducted very carefully according to the specifications of for example the ITU [26]. A proper documentation with all information is recommended so that everybody can reproduce the experiment according to the instructions. At least as important is the clear, meaningful and comparable presentation of the results.

Nearly all studies presented above are based on the evaluation of non-HD content. The resolutions are mostly limited to VGA or only CIF, which is nowadays not longer standard. Unfortunately, all experiments have so different results they barely share any outcome so that some theses could be proven. As the literature research showed the performance of objective metrics strongly depends on which database or which videos were used. Therefore, there are no valid hypotheses on which metric is the best and which completely fails to evaluate video quality, because all of them depend on the content they are tested on. There still remain challenges for developing reliable video quality metrics that perform in the same way on different databases and all sorts of content.

Chapter 3.

Background

This chapter provides insights into the most relevant technologies and concepts that are useful for this work. Section 3.1 gives a brief overview of the video streaming paradigm and the technical background behind streaming. Different transport protocols and problems and challenges of this field are discussed. A short description of the term QoS (Quality of Service) is given in section 3.2, while section 3.3 focuses on the algorithms available for measuring perceived video quality (QoE). An overview of subjective methods and insights into different objective methods are given. I will also try to figure out the occurring problems of subjective and objective methods. Furthermore, the basic concepts of video coding are introduced in section 3.4 with a focus on H.264/AVC. During video compression, different artifacts distorting the image quality can occur, an overview of the artifacts is given in section 3.5. Section 3.6 focuses on the Human Visual System to describe how perception works and why different features can be used for video encoding.

3.1. Video Streaming

By definition video streaming means that source content, which is not available locally on the hard disc of the client, is continuously received and presented to an end-user while being delivered by a provider. The fact that the user can start playing the video file before the entire file has finished downloading distinguishes the delivery method streaming from download. [27]

Figure 3.1 [27] shows a very basic example of a streaming architecture. Usually a video is captured, encoded and afterwards delivered to a receiver. At the client side the data is buffered and after decoding the received bitstream the playback starts. Streams can either be live or on demand. In case of **live streaming** the video files are sent straight to the user's device (computer, smart phone, etc.) without saving them on a server, while **on-demand streaming (VOD)** means that the content is pre-encoded and stored on a server for later use. If a user requests this file the content is played on the user's

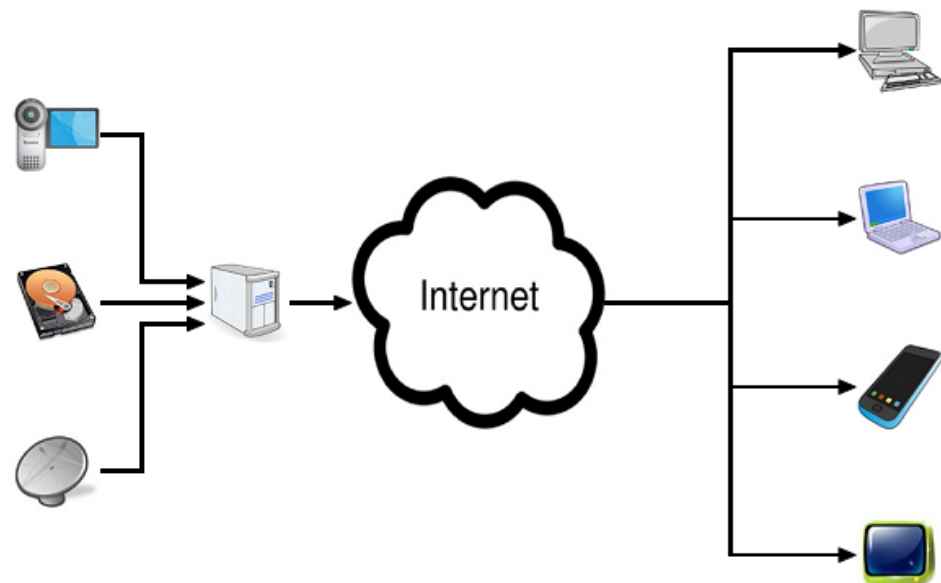


Figure 3.1.: Simple example of a video streaming architecture

device from this source. Live streams of football games for example are only available at the point of time when they actually take place while video-on-demand streams can be requested at anytime [27].

3.1.1. Technical Background

Broken down to the basics, streaming follows the following principles: First of all, the media server has to packetize the bitstream into access units for transmitting them. Typical access units can be audio blocks or frames. These units are delivered from the streaming server to the client via a transport protocol. On the client side the received packets are buffered, decoded and if there is enough data available the playback starts. This process results in a short delay between the request and start of the playback as the client collects the sent data to make a uninterrupted playback possible [27]. Packet loss might occur in every streaming environment, strategies to handle loss are discussed later in this chapter.

According to [28] a basic architecture for a streaming system is divided in the following six steps:

- **Video compression:** Finding a suitable compression codec and the right bit rate for encoding is the main challenge in this task.
- **Application-layer QoS control:** involves congestion control for preventing packet loss and reducing delays as well as error control mechanisms, which are

discussed in one of the following subsections.

- **Media distribution services:** In order to reduce delay and packet loss an adequate network support is necessary.
- **Streaming servers:** These servers process the content and provide the functionality for user interaction. Usually the server waits for a request and then starts streaming.
- **Media synchronization at the receiver side:** Received video as well as audio files should be decoded and synchronized to avoid irregularities, for example between audio and video.
- **Protocols for streaming:** In general there are protocols for network addressing, transport and session control, which are discussed below.

3.1.2. Protocols

Usually the Real-Time Transport Protocol (RTP) is used for transport of multimedia data via User Datagram Protocol (UDP) and is combined with the Real-Time Control Protocol (RTCP) for feedback and session control. RTP is a standardized packet format for delivering video over the Internet Protocol (IP). The RTCP monitors the transmission statistics and the Quality of Service (described in section 3.2).

The Real-Time Streaming Protocol (RTSP) is used for basic media control statements like pause, resume, play and seek in video streams. For describing the session the Session Description Protocol (SDP) is used. Usually this protocol provides information about the duration, codec, bit rate and the content type of a stream.

Nowadays content is mostly streamed directly via HTTP. Content distribution networks (CDN) provide servers in different locations and allow users to stream videos from a server that is geographically close to the user. This was for sure one of the factors causing the extensive growth in video streaming over the web in the last years, seeing as these CDN's can be used effectively for content of any size and type. Streaming via HTTP does not require an extra client - the streams can be requested directly in a common web browser.

3.1.3. Problems in Streaming

Streaming has to deal with different problems because the quality of a network cannot be predicted and does not work with a fixed quality. Since all of this influences the

streaming performance, the long-term goal is to design a system that is able to deliver high-quality content over the internet with a reliable quality despite several unknown factors. In general, we do not know which **bandwidth** can be used between two points on the internet. For example, if the sender is transmitting the files faster than the bandwidth available, packets get lost, while transmitting with insufficient speed also leads to a reduction of the video quality. The goal is to estimate the available bandwidth and adapt the bit rate of the transmitted file to the available bandwidth with rate control mechanisms. This can be achieved by either transcoding the file (described in 3.4) with different bit rates, frame rates or compression formats or by using a scalable video compression standard.

Delay jitter occurs if a packet is delayed and the receiver has to wait for the stream, which slows down the entire process. Using buffers helps to alleviate the problem by playing the media at a constant rate.

Another problem in streaming environments are **losses** that appear in each network. To avoid significant impact on the video quality, video streaming systems have different approaches for error control like forward error correction, retransmission, error concealment and error-resilient video coding.

The **retransmission** approach uses back channels to communicate with the sender and let them know which packets were correctly received and which got lost. The sender is able to react on this information and can resend different missing packets. The advantage of this approach is that the available bandwidth is used very efficiently, seeing as only lost packages are transferred and reactions to changing conditions can also be handled very quickly. This method is not useful if either no back channel is available or the additional delay from using retransmission is not acceptable. **Forward Error Correction** (FEC) adds redundant information, so that in case of packet loss the original bitstream can be reconstructed through this information. The approach of **error concealment** tries to estimate the amount of lost information and replaces the missing content by spatial and temporal interpolated content. **Error-resilient video coding** is an approach that tries to improve the robustness of a compressed video to packet loss [29].

3.2. Quality of Service

The ITU defines QoS in the Recommendation E.800 [30] as the

“Totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service. ”

However, the QoS parameters do not have any relevance for predicting user satisfaction.

Seeing as the terms QoS and Quality of Experience (QoE) (which is part of section 3.3) are often used together it is not always easy to distinguish between them. QoS focuses on measuring the quality of telecommunication services and the physical aspects of a system, while QoE measures user satisfaction which might not only lie within the scope of telecommunications.

Common aspects for measuring the QoS are error rates, bandwidth, throughput, transmission delay, availability and jitter [31].

3.3. Quality of Experience

Quality of Experience (QoE) describes [32]

“The overall acceptability of an application or service, as perceived subjectively by the end-user”.

To guarantee customer satisfaction, monitoring the video quality and the related QoE is an essential but also very challenging task. In the following I will describe the two methods subjective and objective quality assessment, further I will try to figure out the different problems and methods available, as well as different approaches.

3.3.1. Subjective QoE

Since the human observer is always the end user of each video application the best and most reliable solution is to let him/her review the video quality. Subjective method reviews the quality perceived from a number of subjects and covers the Human Visual System (HVS) very well [17]. According to the ITU specifications [26] on subjective tests a group of trained and untrained users view and rate the video files. There are strict evaluation requirements, e.g. viewing conditions such as the distance, display brightness and contrast, room illumination, duration and content of the test files, as well as the number of observers that has to be followed to guarantee comparable results.

As listed in the ITU-Rec. [26] there are different methods covering many applications. The choice of method to be used for an experiment depends on the context and purpose. A description of the most important methods is given below:

- **Double Stimulus Impairment Scale (DSIS):** This method is cyclic and the subject is first presented the original video followed by the encoded one and is asked to rate the encoded video with respect to the quality of the original video. This method is used for example for measuring the robustness of systems or for evaluating new systems. DSIS also provides good results for evaluating different artifacts.

- **Double Stimulus Continuous Quality Scale (DSCQS):** This method requires two sequences, one from the reference and another one from the impaired sequence. The sequences are shown twice in random order and the subject is asked to rate each video separately and as they do not know which one is the reference each sequence independently. DSCQS experiments are often used to measure the quality of systems where the quality of test and reference files are similar.
- **Single Stimulus Method (SS):** Subjects are presented encoded sequences with a length of about ten seconds and are asked to rate their quality. This method works without reference files and is often used if, for example, there are different encoder settings to test for which one performs better.
- **Stimulus Comparison Method (SC):** For this method the subjects are shown two different encoded scenes in random order - their task is to rate the quality of one specific scene with respect to the other ones.
- **Single Stimulus Continuous Quality Evaluation (SSCQE):** The subjects are only shown the encoded video sequence, normally with a longer duration than in the other experiments (about 20 to 30 minutes) and asked to rate the perceived quality.

After finishing the experiment with the methods and requirements described above the gathered data is ready for statistical analysis. All human ratings are combined to a single overall value called the Mean-Opinion-Score (MOS). The MOS value is a numerical indication of the perceived quality, the value normally reaches from 1 “Bad” to 5 “Excellent”.

The MOS for a test scene j is calculated as follows:

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} m_{ij}$$

where N_j is the number of test subjects in the experiment. This value represents the averaged subjective score m_{ij} obtained from subject i after viewing the test scene j [26].

Problems

On the one hand subjective evaluation is a reliable parameter for video quality but on the other hand this method is unfortunately not useful in real-time scenarios where the quality review has to be available right after encoding or sending the files to the customer [17].

In general, there are many influencing factors on the availability of persons for reviewing, which might have an impact on the results of subjective evaluation. Subjects for example can be easily manipulated by different instructions or prior information about the content. Choosing the right method for evaluation is also essential but not always easy. The subject's personal interests, different expectations regarding the question "what is quality", the experience with videos and rating them, their HVS as well as display type and viewing conditions can also have an impact on the results of different studies [33].

Another problem is that these experiments are extremely time consuming and consequently also very expensive. It might be difficult to recruit a large number of subjects to get a variation in the individual viewing opinions. The specifications require an appropriate laboratory for the experiment, which is why subjective tests cannot be carried out anywhere [15].

3.3.2. Objective QoE

As described in section 3.3.1, subjective evaluation is not suitable and useful in certain situations. A good alternative to subjective evaluation are objective quality metrics, which are much more economical. The main goal of an objective metric is to automatically predict the perceived video quality close to the one reviewed by subjects without human action by means of mathematical models [15].

Common application fields for objective methods can either be monitoring the video quality in a video control system, benchmarking different video processing systems and algorithms and optimizing algorithms and parameter settings in video processing systems [34].

Objective quality measurement methods can be classified into full-reference (FR), reduced-reference (RR) and no-reference (NR) methods, all depending on the available amount of information about the reference video [15], as described in detail below.

Full-Reference Methods

FR methods require the complete reference video for comparison and are used in applications where the original video is available, e.g. for measuring the encoding performance. For this purpose the reference and the encoded test video will be compared to each other frame-by-frame [15].

Figure 3.2 [4] shows a sample FR application for testing different codec settings.

FR methods are mostly used for evaluation of non-real-time scenarios where the original and the encoded video are available, such as codec tuning or testing [17].

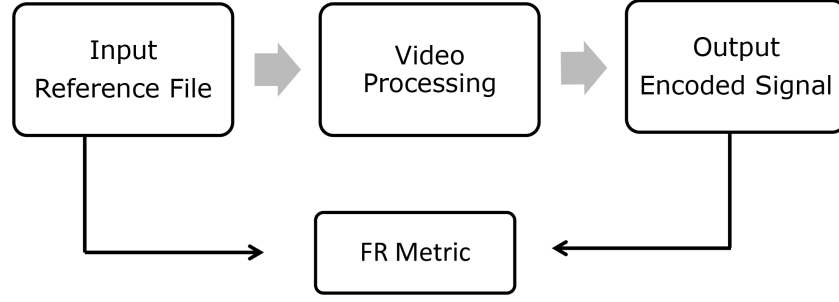


Figure 3.2.: FR video quality framework

Below a list of the most common available FR metrics:

- **Mean Square Error (MSE)** is the simplest implementation of a measuring algorithm - it objectively quantifies the level of distortion or impairment between a reference and the impaired video sequence. MSE calculation is often used because it is easy to calculate and has a clear physical meaning. According to [6] the MSE can be calculated as follows:

$$MSE = \frac{\sum_{i=1}^N \sum_{j=1}^M [X(i, j) - Y(i, j)]^2}{M \times N}$$

where $X(i, j)$ is the original signal and $Y(i, j)$ is the processed signal at pixel (i, j) . M and N are the numbers of horizontal and vertical pixels. The MSE is small if Y is similar to X . Therefore, the greatest similarity is achieved if the MSE value is near or equal to 0.

- **Peak Signal to Noise Ratio (PSNR)** is well known and the most frequently used objective metric, it is easy to calculate and can provide a simple and efficient approach to distortion estimation. The value is calculated with the MSE by averaging the squared intensity differences of distorted and reference pixels [35]. According to [6] the PSNR is calculated as follows:

$$PSNR = 10 \log_{10} \left(\frac{L^2}{MSE} \right)$$

where L is the maximum possible pixel value of the image, which means that for a typical 8-bit value $L = 2^8 - 1 = 255$. The result is expressed in decibels in values between 0 and 100 dB. Achievable values are around 30 to 50 dB for fair quality - the higher the PSNR value the better the quality [6].

- **Structural Similarity (SSIM)** This method is based on the extraction of structural information from images. The SSIM index measures the similarity of the following three elements: luminance (brightness), contrast and the local structure.

SSIM covers the HVS system much better but is unfortunately more complex and needs much more time for calculation [5].

As described in [5] the SSIM value is calculated as follows:

$$S(x, y) = l(x, y) \cdot c(x, y) \cdot s(x, y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \cdot \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right) \cdot \left(\frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \right)$$

Assuming that x and y are images taken from a video at the same location, $l(x, y)$ is the luminance similarity, $c(x, y)$ is the contrast similarity and $s(x, y)$ the structural similarity. μ_x and μ_y are the local mean values of x and y while σ_x and σ_y are the local sample deviations of the two images x and y . σ_{xy} presents the sample cross correlation. C_1 , C_2 and C_3 are constants for stabilizing the calculation.

Figure 3.3 [5] shows a diagram for a better understanding of the SSIM calculation process.

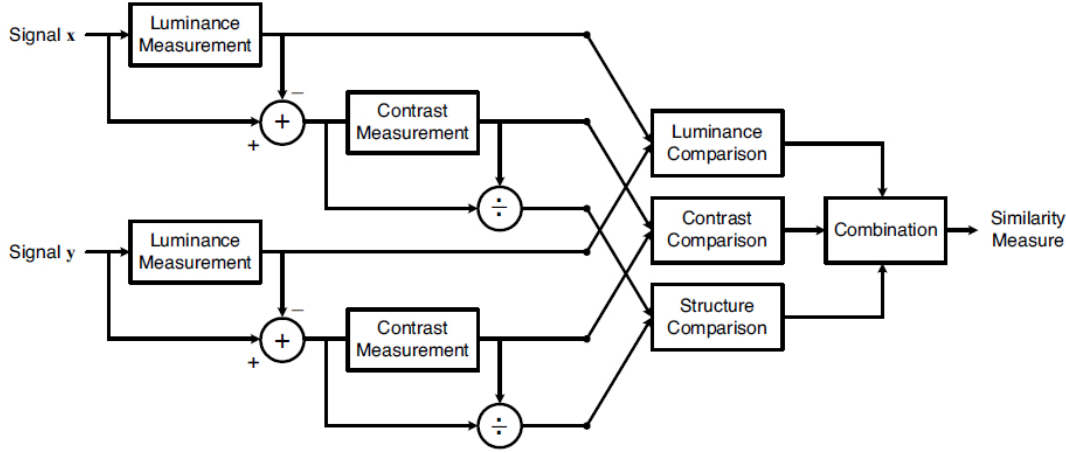


Figure 3.3.: Process diagram of SSIM calculation

The calculated values reach from -1 to 1. The greatest similarity is achieved with a SSIM of 1. A SSIM of 1 means that the distorted picture is identical with the reference frame [6].

- **Video Quality Measure (VQM)** was developed by the National Telecommunications and Information Administration (NTIA) and presented in [7]. It divides sequences into spatio-temporal blocks, then a number of perceptual features (e.g. blurring, jerkiness, noise, block distortion and color distortion) is extracted with special filters from these blocks and afterwards compared with the reference video. Due to an excellent study performance this method was adopted as a national standard by the American National Standards Institute (ANSI)[7].

The VQM value is calculated as follows [7]:

$$\begin{aligned} VQM = & -0.2097 \times \text{si-loss} + 0.5969 \times \text{hv-loss} \\ & +0.2483 \times \text{hv-gaint} + 0.0192 \times \text{chroma-spread} \\ & -2.3416 \times \text{si-gain} + 0.0431 \times \text{ct-ati-gain} \\ & +0.0076 \times \text{chroma-extreme} \end{aligned}$$

where different features are used. A detailed description of the filter used to extract each of the values needed can be found in [7].

The calculation above results in output values from 0 to 1. A VQM value of 0 means no distortion and good quality.

- **Multiscale SSIM (MS-SSIM)** The MS-SSIM is an extension of the SSIM. It is calculated in the same way as the SSIM at different resolutions, so the quality is calculated for different viewing distances. As there is much more calculation work to do the MS-SSIM needs more time for calculation than the SSIM [36].
- **Visual PSNR (VPSNR)** As described in [37] this measure is block based and easy to calculate as it is based on the PSNR calculation. VPSNR incorporates the HVS much more than the PSNR.
- **Visual Information Fidelity (VIF)** presented in [38] measures the loss of perceivable information in the distortion process. The visual information is measured by computing mutual information between reference and distorted video. The authors described three models for measuring the visual information: Gaussian scale mixture model (GSM), distortion model and HVS model.
- **Motion-based Video Integrity Evaluation(MOVIE)** tries to combine spatial values that are similar to several VQA algorithms with temporal values. The metric analyzes the reference as well as the encoded content with special filters. The resulting values contain a temporal quality component based on motion information and a spatial quality component [39].
- **Visual Signal-to-Noise-Ratio (VSNR)** is computed by a two-stage approach for detection of distortions through wavelet-based models [40].

The list is not a complete listing of all available FR methods, seeing as many studies cover this field and the number of objective FR metrics is increasing rapidly. I have listed the ones which are relevant for my experiment PSNR, SSIM and VQM in more detail and mentioned other well known but more difficult to calculate metrics.

Problems of FR Methods

Unfortunately the PSNR and MSE values have been criticized in literature [6] for not covering the HVS system very well and not correlating well with subjective evaluation results [33]. criticized that the PSNR is based on a byte-by-byte comparison and does not consider the actual content.

But there are also studies [11] stating that the PSNR correlates well with subjective test results in case of high quality digital cinema content or is useful for comparing similar codec settings based on the same test data [10]. An overview of different studies is given in chapter 2.

Reduced-Reference Methods

RR methods are not based on the full reference video but on partial information extracted from the reference video. Hence, the encoded test video and the extracted features from the reference file are compared. The metric specifies which information should be extracted for comparison [15]. An example for a comparable feature is the amount of motion or the spatial detail [33]. These metrics are used in networks where partial information about the original content as well as a back channel for transmitting this data are available. Therefore it is necessary to find a good balance between the number and details of extracted features and that result in data rate of extracted features and the accuracy of the quality prediction. Common reduced-reference video quality assessment methods are for example the RR VQA based on natural image statistical models in the wavelet transform domain proposed in [41], as well as the normalization transform-based RR VQA presented in [42] and the method based on digital watermarking described in [43].

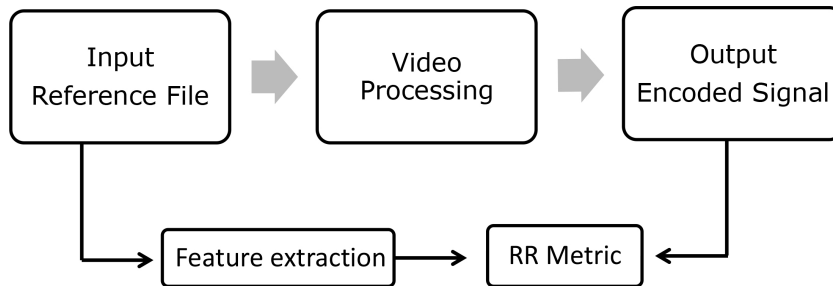


Figure 3.4.: RR video quality framework

Figure 3.4 [41] shows a basic framework for RR VQA. It requires the feature extraction at the sender side as well as the feature extraction process to be followed by the quality assessment on the receiver side. In general extracted features have a much lower data rate and can be sent over an ancillary channel. RR methods are a good compromise

between FR and NR methods and can be used in real-time systems for controlling the video quality [41].

No-Reference Methods

NR methods are used if the reference video is not available, they are also called “blind” quality assessment and only analyze the encoded video. In general these metrics are less accurate than FR methods [15]. Figure 3.5 shows how NR methods work.

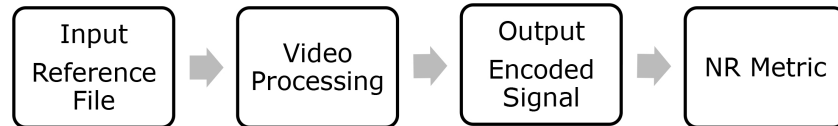


Figure 3.5.: NR video quality framework

NR methods are mostly used in real-time applications for measuring the impact of lossy transport at the receiver side. Generating such metrics is extremely difficult and the use is limited to specific scenarios and distortion types. Common artifacts are listed in section 3.5.

There are different approaches for NR methods. The method proposed by Yang et. al in [44] uses motion vectors for calculating the quality. They follow the approach that the HVS is sensitive to sharp changes and review the respective fields. A metric for calculation of blocking is proposed in [45].

3.4. Video Encoding

Uncompressed video requires large amounts of storage space and in consequence a large bandwidth at transferring the files via internet. Therefore, for streaming purposes encoding with a very effective but also lossy algorithm is necessary with respect to the video quality, which should always be as high as possible. The challenge of content creators and service providers is finding new possibilities to transfer the highest quality with the lowest possible bit rate.

3.4.1. The Need for Compression

The following short calculation shows which enormous amounts of data an uncompressed video produces.

A standard resolution of 720×576 pixels with a frame rate of 25 frames-per-second (fps) and 8-bit color depth for each component in the RGB color space or 3 bytes per pixel would require $720 \times 576 \times 25 \times 3 = 29.66\text{Mb}$ per second. A one-hour uncompressed video would result in a file size of 104.28 GB. For HD resolutions a second of uncompressed video will require 148.32 MB - an hour about 521.42 GB.

The extremely high storage requirements are not the only the reason for why we need compression. Processing uncompressed video files causes an extreme high workload and is in general not possible with common processor and graphic cards.

3.4.2. What is Compression?

Video encoding is the process of converting an uncompressed input stream in a way that it is more efficient for further use, e.g. streaming in terms of file size, video specification and packaging. Encoding packs the data in a smaller space which can be achieved by reducing the amount of data by eliminating redundancies in a video file. Perceptual, temporal, spatial and statistical redundancies can be minimized.

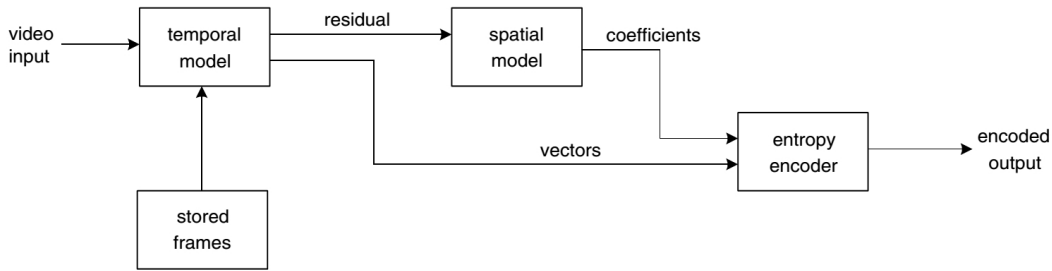


Figure 3.6.: Block diagram of a video encoder

Figure 3.6 [46] shows the sequence of encoding. The uncompressed source is the input file, which is first compressed using the **temporal model**, where temporal redundancies, found by predicting the frames, are discarded. The output of this process results in a residual frame and a set of motion vectors. The residual frames, which are generated by subtracting the prediction from the actual frame, are the basis for the **spatial model**, in which the frames are transformed into another domain. The results are transformation coefficients that are quantized to reduce the file size. Then the resulting transformation coefficients and the results from the temporal model are compressed by an entropy encoder, which removes statistical redundancy and produces a compressed video file. In order to reconstruct the file a video decoder that decodes the coefficients and motion vectors with an entropy decoder is needed. The motion vectors are used to create predictions of the current frame [46].

Compression can be classified into two methods: lossy and lossless. **Lossy compression** means that the encoded file contains less data than the original file. Normally this results in lower file sizes due to the fact that some information may have been lost during the encoding process. The original file cannot be extracted from the encoded file. As described in section 3.6 the HVS usually does not even notice the loss of information and the perceived quality stays almost the same. **Lossless compression** tries to compress the video without discarding of information and allows a 100% recovery of the original data stream. This method is used if massive reduction of the file size is not the primary goal. A typical area of application for lossless compression are video editing environments where video files have to be available in the best quality for further use but should not require massive amounts of file storage.

3.4.3. Requirements of a Compression System

The main requirement for a video compression system is to produce the highest quality with the lowest bit rate. According to [47] there are various other features that should be covered by a compression system:

- **Robustness to loss**
- **Low complexity**
- **Flexibility**
- **Low delay**
- **Authentication**
- **Standardization**

3.4.4. Basics of Compression

While figure 3.6 shows a very basic block diagram of video encoding, figure 3.7 [47] shows which steps are requested in a video encoding system. The input file is encoded at the sender side with inter-frame motion prediction, transmitted via a communication network and decoded on the receiver side.

Almost all video coding methods are based on the following concepts, which are described below: prediction, inter- and intraframe compression, motion compensation and entropy coding.

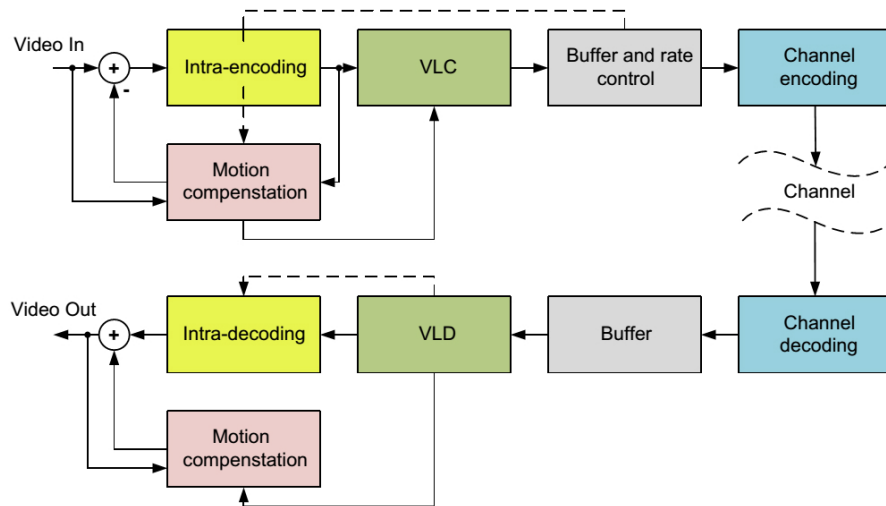


Figure 3.7.: Video compression architecture

Macroblock

A typical video frame is divided into macroblocks. The macroblock contains all information for reconstructing the signal, this might be motion vectors or quantizer information. Each of these macroblocks consists of three components, a 16×16 array for Y (Luminance), and two 8×8 arrays for Cr and Cb (color information) [48]. A typical macroblock structure is shown in figure 3.8 [47].

Prediction

Usually, each encoder starts with prediction and searches for redundancies and predictable parts in a video sequence. If such redundancies are available only the difference between the consecutive frames is stored.

According to [47] there are four classes available for prediction:

- Intra Prediction
- Forward Prediction
- Backward Prediction
- Bidirectional Prediction

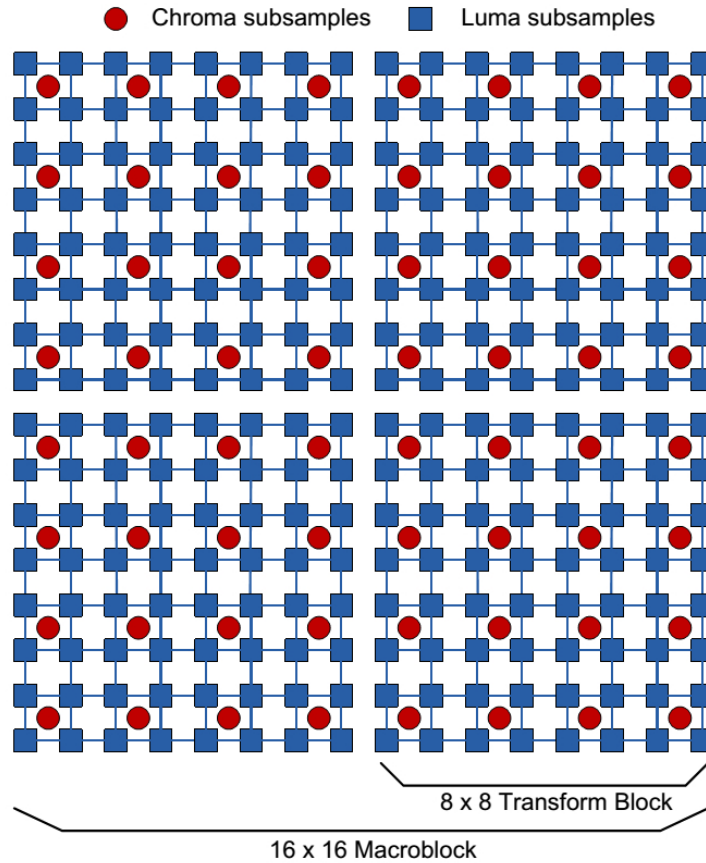


Figure 3.8.: Macroblock structure

I-,P-,B-frames

There are three major types of frames specified in each coding standard:

- **I-frames** are the so-called intra coded frames which are coded without reference to other frames. Each of these frames is individually encoded and is used to create a starting point for decoders.
- **P-frames** The predicted frames are encoded by prediction from other I- or P-frames. P-frames provide more compression than I-frames and are also used as reference for B-frames and following P-frames.
- **B-frames** (Bi-directional frames) use the nearest I- and/or P-frame as reference and are not used for any further prediction [47].

Group of Pictures

The coded pictures described above are arranged in a sequence which is known as Group of Pictures (GOP). A typical GOP structure is shown in 3.9 [47] and can have a variable length depending on the type of video being encoded. Each GOP in a video stream contains an I-frame followed by zero or more P- and B-frames [47].

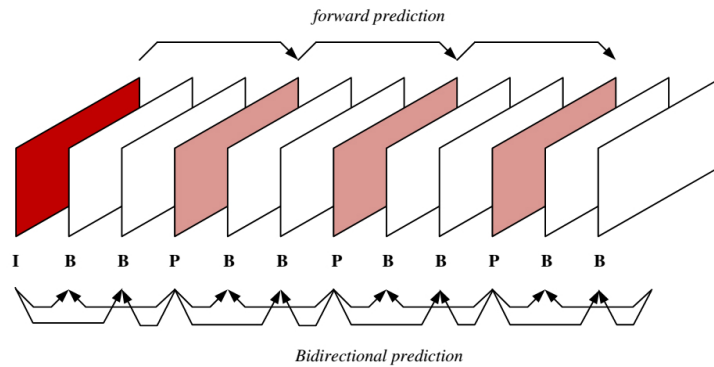


Figure 3.9.: Typical GOP structure

Keyframes

Keyframes are special types of I-frames which are used as reference points for subsequent I-frames. These keyframes are used for searching within a video file [47].

Inter- and Intraframe compression

Interframe compression is a method in which consecutive frames are compared and the differences between these frames are encoded. **Intraframe compression** handles each frame separately and compresses each frame without reference to the others. Typically, the first picture of a video clip is intra-frame encoded. The size of the frame will depend on the nature of the image [47].

Motion Estimation

Motion estimation measures the movement of elements within a frame sequence. The key of a compression system is to use the motion vectors from the motion estimation process in combination with a motion compensation process, which is described in the next section [47].

Motion Compensation

The estimated motion vectors are used in the motion compensation process. The MC subtracts the reference frame from the following frame to produce a prediction of the current frame [47].

Discrete Cosine Transformation

As described above each image is divided into macroblocks. Each of these macroblocks is transformed into another domain called the frequency domain, using the discrete cosine transformation (DCT), which generates 64 coefficients for the input signal. The coefficients are reordered in a one-dimensional array with a zigzag mechanisms from the top left to the bottom right - so the smaller coefficients can be found at the end [47].

Quantization

Quantizers reduce the number of bits in a stream. This step is lossy because information which is not perceivable by the HVS is discarded and cannot be reconstructed [49].

Entropy Coding

The entropy encoding step removes statistical redundancies from the quantized output stream. The entropy is the smallest number of bits needed to represent a frame. Common entropy coders are the Huffman Coding and Arithmetic Coding [47].

3.4.5. Video Compression Standards

Since 1980 the field of video compression evolved to the successful coding standards that are used nowadays. Standardization is needed to ensure interoperability with all available services. A standard defines the goals and constraints a coding environment has to deal with. Different video coding standards normally only define the bitstream format, the syntax and the decoding process. Organizations that developed video coding standards in the past are the International Telecommunication Union (ITU) and the International Standards Organization (ISO) with the joint ISO/IEC (International Electrotechnical Commission). The Moving Picture Experts Group (MPEG) is a working group of the ISO/IEC, that was founded in 1988 and is responsible for the most popular standards in video coding [50].

Year	Standard	Application Area
1988	H.261	video conferencing
1993	MPEG-1	consumer video on CD
1994	MPEG-2	broadcast of SD content
1997	H.263	video-telephony, video on mobile phones
1998	MPEG-4 Part 2	video on the internet
2003	H.264/AVC	improved video compression for Blu-ray, video broadcasting of HD content
2013	High Efficiency Video Coding (HEVC) (H.265)	Ultra HD, 4K content

Table 3.1.: Time line of compression standards

Table 3.1 gives a short overview of available standards and their first appearance. **H.261** was the first codec with the macroblock concept on which many of the later standards are based. H.261 supported only two video sizes, CIF and QCIF as well as I- and P-frames. Subsequent developments, for example the **MPEG-1** standard, were based on the H.261 codec and supported other resolutions as well as B-frames. The **MPEG-2** standard introduced profiles and levels. **H.263** was designed for low bit rate applications, which was originally developed for video-telephony but was used also in flash video applications, as for example used by YouTube. The **MPEG-4** standard was designed for encoding videos at lower bit rates with higher qualities than its ancestor MPEG-2. With the implementation of **H.264/AVC** a new standard, which is still one of the most frequently used codecs, was born. H.264/AVC is used as standard in many internet streaming environments, as well as format for blu-ray players and in many mobile services. The **HEVC standard H.265** developed in 2013 is designed to provide the same video quality as H.264/AVC but with a 50% higher compression efficiency. HEVC was designed to deal with ultra HD and 4K content which is advancing to be used by a majority of normal users and no longer only by digital cinemas for example [50].

H.264/MPEG-4 AVC Codec

H.264/AVC was standardized by the Joint Video Team (JVT) of experts from ISO/IEC (International Electrotechnical Commission) and ITU. The H.264/AVC compression scheme follows common coding structures as described above with slight modifications, figure 3.10 [48] shows a simple block based diagram of a typical H.264/AVC encoder. Changes with respect to other coding standards are, that in order to employ a higher efficiency, multiple reference frames, variable block sizes for motion estimation and compensation are used and the DCT is replaced by an integer transform. To reduce the block artifacts a deblocking filter is used within prediction [48].

The H.264/AVC consists of two conceptual layers. The video coding layer (VCL) and the

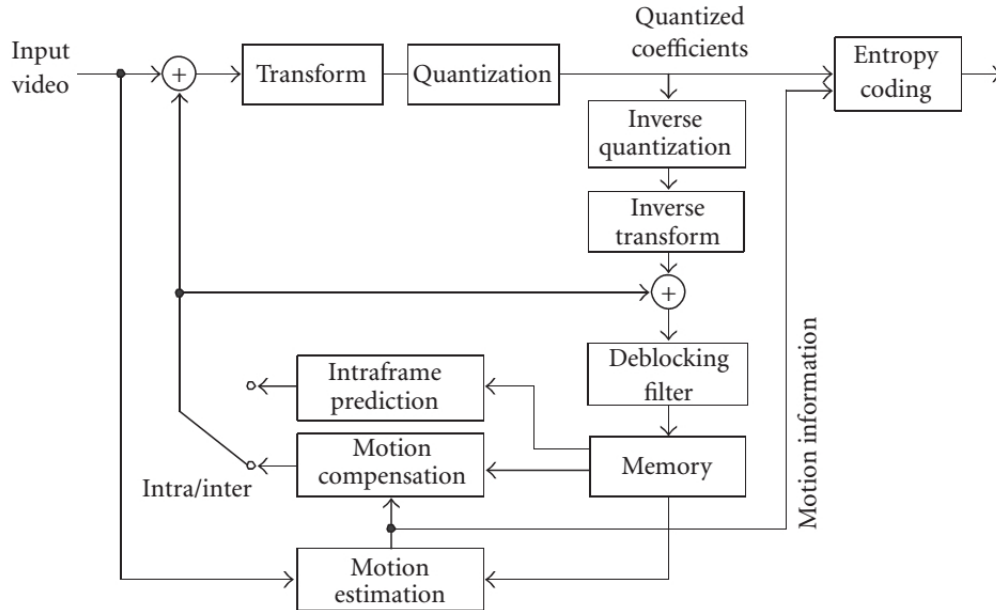


Figure 3.10.: Block diagram of a H.264/AVC encoder

network adaption layer (NAL), which converts data from the VCL into a format usable for networks or storage devices [48].

In **intra prediction** the information of already encoded macroblocks of the same image is used for the prediction of the luminance component. There are two types depending on the size of the available macroblock: 4×4 and the 16×16 . In general nine different prediction modes are supported, which are automatically chosen by the encoder depending on the content [46].

For **motion compensated prediction** with multiple reference frames a macroblock is predicted from the frame that is already being transmitted. Therefore one macroblock can be divided into smaller partitions, and if necessary each of these samples can be partitioned further. Figure 3.11 [48] shows how different macroblocks can be coded. These different ways are called modes. In the 16×16 mode one motion vector is estimated for the macroblock. All other modes divide the macroblock into rectangular or square parts and estimate a motion vector for each of these parts. Additional picture reference parameters are necessary to link the parts together [48].

Instead of the commonly known DCT, which is used in other standards, a **transform coding** algorithm is used to reduce the spatial redundancy of the prediction error signal. For this purpose different integer transforms available are applied on the bitstream. There are three types of transforms, which in general have a size of 4×4 [48].

Instead of the known **entropy-coding** schemes the H.264/AVC standard introduces two

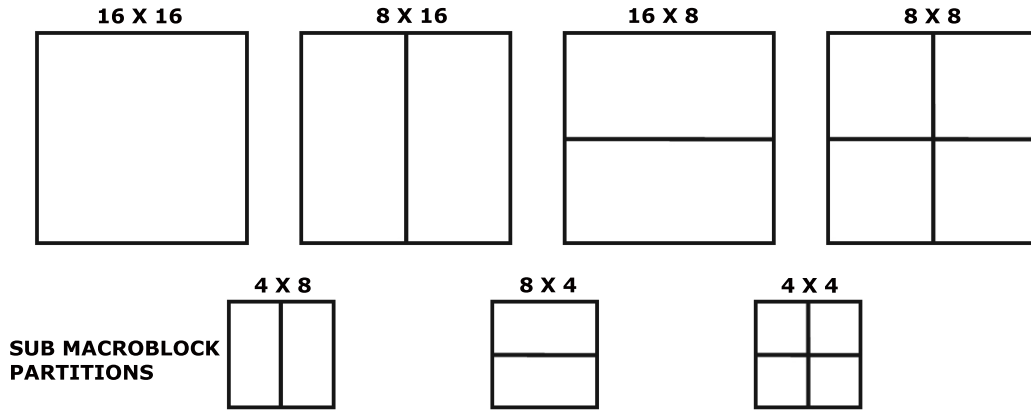


Figure 3.11.: H.264/AVC macroblock partitions

new methods. The context-adaptive variable length coding (CAVLC) and the context-adaptive arithmetic coding (CABAC). Experiments showed that both methods result in a significant increase of encoding efficiency compared to earlier methods. The CABAC is more complex and requires more processing time than CAVLC but is also 5 - 15% more efficient. CAVLC is supported in all H.264/AVC profiles while CABAC is only supported in the main profile [48].

Encoding profiles and levels

As each application has different requirements, profiles and levels are a possibility to maximize the interoperability while limiting the complexity. A profile is

“a subset of the entire bitstream syntax” [48]

- which can be called the basic coding tool. A level specifies the range of allowed values for the parameters in the bitstream as specified in the profile. H.264/AVC defined three profiles - Baseline, Main and Extended Profile, which can be combined with any of the specified 15 levels [48].

Figure 3.12 [48] shows how the profiles are based on each other and which tools they support.

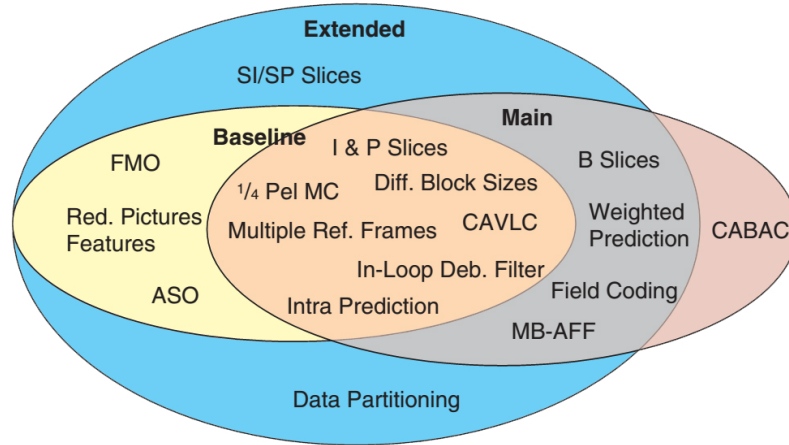


Figure 3.12.: H.264/AVC profiles

3.4.6. Video Transcoding

Video transcoding is specified as:

“The operation of converting a video from one format into another format. A format is defined by such characteristics as the bit rate, frame rate, spatial resolution, coding syntax, and content.” [51]

In general, transcoding is used to convert a compressed video file to another compressed video file with different settings as shown in figure 3.13 [52]. Settings that can be changed are the format, bit rate, spatial resolution, temporal resolution and the error resilience [53].

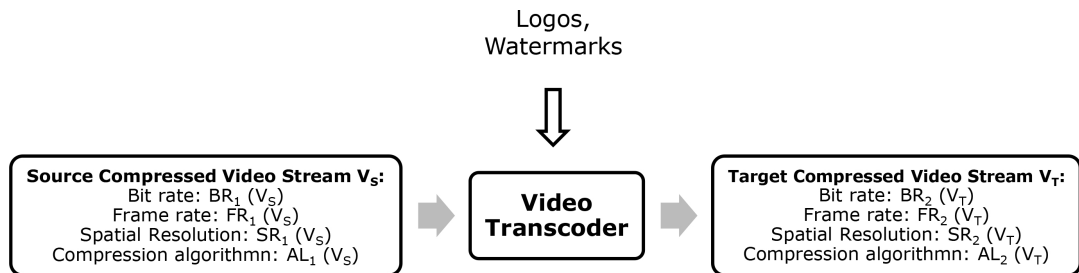


Figure 3.13.: Transcoding operations

According to [52] there are three requirements a transcoding system should meet: The information in the original bitstream should be exploited as much as possible, the quality of the transcoded video should be as high as possible and the delay as well as the memory requirements should be as low as possible.

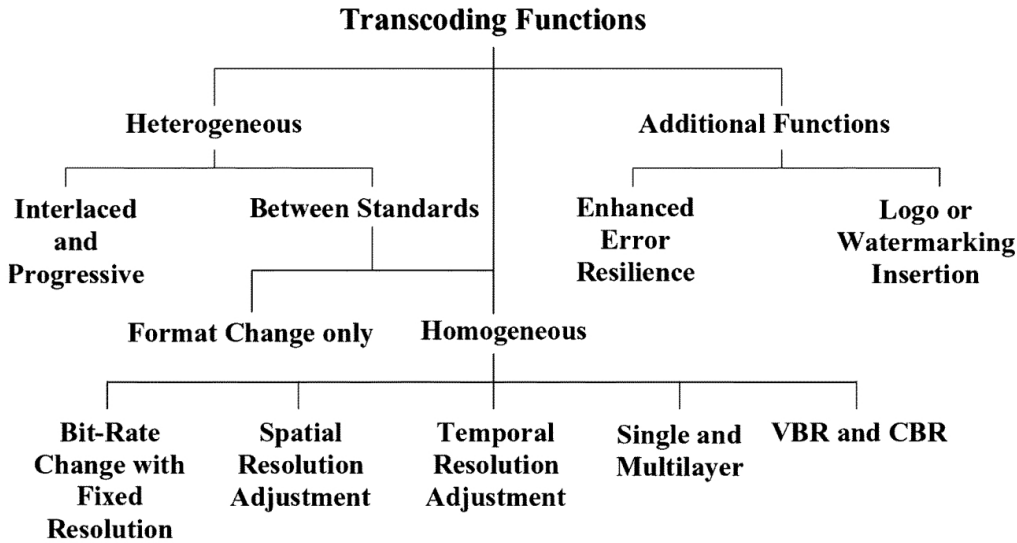


Figure 3.14.: Transcoding classification

Figure 3.14 [52] shows the different application areas for transcoding. One is for example multimedia streaming networks. As it is very important for each supplier to deal with different channel characteristics like bandwidth, packet loss ratio or error rates, transcoding is used for adapting the multimedia content before delivering the file over the network to multiple receivers. Another area of application is watermarking to insert new information in a video, e.g. company logos and watermarks. Transcoding is also used in video conferencing systems to adapt content to different network resources [51].

H.264/AVC Scalable Video Coding Extension

Scalable encoding is another approach besides transcoding for adjusting different settings in a video file. The Scalable Video Coding Extension (SVC) allows subsets of the stream to be removed. Each of these subsets can represent a lower spatial resolution, temporal resolution or a lower quality video signal. SVC solves the resource problem of transcoding or saving a video stream in different resolutions, for example in video streaming, video conferencing and even broadcasting. Once the SVC scheme is set up, for example in a video streaming service, the source content has to be encoded only once with the highest required resolution and bit rate. The output allows to transmit files with a lower resolution or bit rate without transcoding or encoding the file for a second time. Some of the subsets in the bitstream can be discarded and so the desired bit rates can be achieved [54].

3.5. Compression Artifacts

Artifacts are distortions in videos and occur due to loss of details in the video signal either during transmission or compression. There are several types of artifacts that can be classified as follows [55] [49]:

- **Blocking** is a typical distortion caused by common block-based DCT compression mechanisms. A subject perceives block structures. The blocking effect is one of the most known artifacts, recent codecs have implemented so-called deblocking filters to reduce the visibility of this artifact.
- **Ringing** causes rippings along sharp edges or contours of objects.
- **Flickering** appears in scenes with high texture content.
- **Blurring** is caused by the loss of spatial information and manifests in the reduction of edge sharpness. Blurring occurs when transforming the coefficients quantization while encoding or when reducing the bit rate for encoding.
- **Noise** is an uncontrolled pattern that distorts the video file and can be noticed along high-contrast edges of objects or moving objects and occurs on DCT based compression methods.
- **Jerkiness** means that sequences with smooth and continuous motion appear as a series of still pictures with jerky motion. This artifact is normally caused by low temporal resolutions or by insufficient motion compensation.
- **Color Bleeding** occurs if the edge of one color bleeds into or overlaps another color.

3.6. Human Visual System

Since digital images are viewed by humans it is important to consider the Human Visual System (HVS) when designing video compression systems. The knowledge of the HVS can also be useful for the development of VQA methods. Each image contains perceptual redundancies, which are details of an image that are not perceivable by the human eye. Compression is based on the discarding of information that humans cannot perceive without affecting the visual quality.

Figure 3.15 [49] shows the major components of a human eye. Light is focused by the *cornea* and the *lens*, which projects a visual stimulus onto the *retina*. To focus objects from different distances accommodation of the lens is needed. The retina contains two

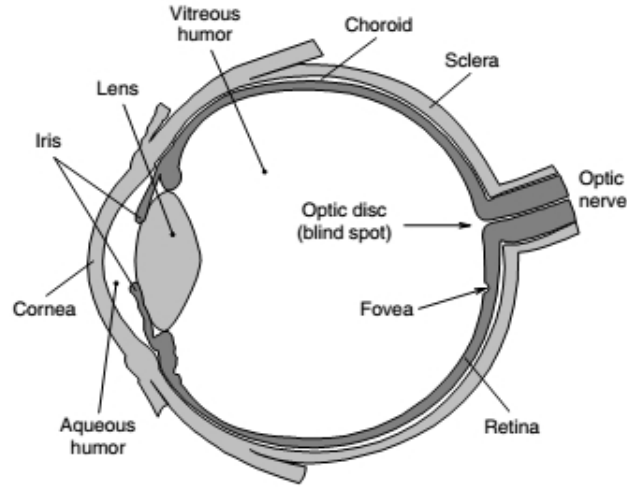


Figure 3.15.: The human eye

types of photoreceptor cells. One called the *cone* cells, which are sensitive to color vision under normal light conditions and are divided in the L- M- and S-cones, as well as the *rod* cells, which are sensitive to variations in intensity and are used under low light conditions. The incoming light is converted into neural signals, which pass through several layers in the retina. Each retina contains 120 million rods and six million cones, which makes the human eye more sensitive to variations in intensity than in color. The *fovea*, a small area on the visual axis, has the highest density of cone cells. As there are more green cones than red and blue ones, the perception of green color leads to more brightness than red and blue colors. The colors are represented in the YCbCr color space where Y indicates the brightness and Cb and Cr represent the chrominance components [34].

Figure 3.16 [49] illustrates the pathway from the retina through the cortex. Before images are relayed from the *optical nerves* to the *visual cortex* they are reorganized in the *optical chiasm* and the *lateral geniculate nucleus (LGN)*. The LGN contains six layers of cells, three for receiving input from each the left and right eye. Significant computation and transformation of the visual system is done in the area of the primary visual cortex, which contains 200 million neurons all containing different receptive fields [34].

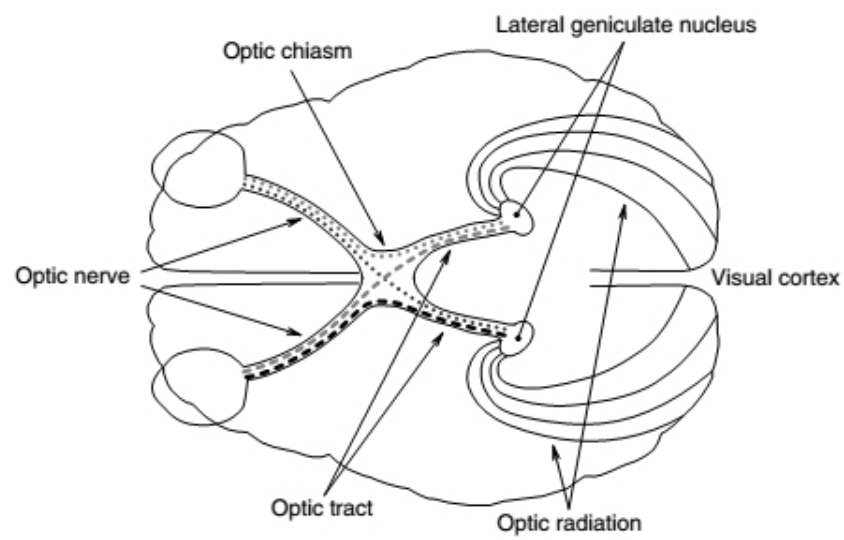


Figure 3.16.: Visual pathway in the human brain

Chapter 4.

Experimental Setup

The main goal of this experiment was to map the influence of different encoder settings on the objective quality of the resulting videos. The following tasks were necessary to collect the data.

First of all programming all necessary scripts to encode the footage files in different variations. Therefore all parameters and settings of the encoding tools were studied and the key features the experiment should cover were extracted. Details on all encoder settings used and a description of those can be found in section 4.4. All cases that should be covered by this experiment are listed in section 4.3. The software tools used, as well as the hardware for this experiment can be found in the sections 4.1 and 4.2. Further details on the used footage can be found in section 4.5.

The second task was running the prepared scripts and gathering data. This was one of the most time consuming parts of this experiment more detailed information regarding this can be found in section 4.6, as well as a short overview of occurred problems and limitations in section 4.8.

After all videos had successfully finished encoding the next task was to compare the quality of all generated clips with the original footage to get some measurable data. This step was also initiated with an automated script. More details on which software was used for that can be found in section 4.7. A detailed description of available and used objective quality metrics is included in chapter 3.3.2.

The last task was to inspect all gathered data and try to break them down to one level and get some conclusions out of it. Results of this experiment will be discussed in chapter 5.

4.1. Hardware

The experiment was performed on a PC with the setup as presented in table 4.1.

OS	Windows 7 Home Premium x64 SP1
CPU	Intel Core i5 430M (2,27 GHz)
RAM	4GB DDR3
GPU	ATI Mobility Radeon HD5650 (1 GByte GDDR3)
Hard Disk	640 GB Western Digital WD6400BEVT 5400 rpm Sata

Table 4.1.: Hardware setup

For storing the original files as well as all output files and encoded video files an external hard drive from Western Digital with USB 2.0 connection and 2 TB storage space was used.

4.2. Software

The following listing gives an overview of which software tools were used to perform this experiment.

- **x264**¹ is a free software library for encoding video files into the H.264/MPEG-4 AVC codec with a console based tool. A detailed description of the codec and how it works can be found in section 3.4. A sample encoder call can be found in section 4.6.
- **Avisynth**² is a simple script based filter collection for video post-production tasks. It was mainly used for changing the resolution of the footage or for converting the videos to the right color spaces.
- **Avisource Filter for Avisynth**³ The avisource filter plugin loads a video in the “.avi” container from a location. It was used for opening the footage file with the x264 encoder. The scripts can also be opened in common media players.
- **Microsoft Visual Studio**⁴ The IDE was used to programm the script for encoding all videos with the different settings and the automated process for comparing the videos with the MSU VQM Tool.
- **Notepad++**⁵ is a free text editor that is, in my opinion, much more powerful and user friendly than windows notepad. The tool was mainly used for having a quick look at different statistical files generated from the x264 encoder.

¹<http://www.videolan.org/developers/x264.html>

²<http://sourceforge.net/projects/avisynth2/>

³<http://avisynth.nl/index.php/AviSource>

⁴<http://www.visualstudio.com/>

⁵<http://notepad-plus-plus.org/>

- **MSU Video Quality Measurement Tool**⁶ is a tool for objective video quality measurement. It was used to compare the reference video with the encoded video and for calculation of the desired metrics. The tool provides usage via the console within the PRO version and gives detailed feedback regarding the video quality.
- **Eclipse**⁷ I have extracted all information from the “.log” and “.stats” files generated from the x264 encoder and the MSU VQM Tool with Java and merged all information into a single “.csv” files for further use.
- **R Statistics**⁸ R is a freely available software environment for statistical computing and graphics. The tool was used for generating the plots in chapter 5.

4.3. Use Cases

When defining the general use cases for this experiment I tried to extract which parameters and settings of the x264 encoder are interesting and might have an impact on the resulting quality. A detailed listing of all encoder settings used can be found in section 4.4. In general, I tried to focus on mobile video streaming, meaning generating files that could be consumed on a mobile device and under different circumstances, such as perhaps very bad and low internet connection. The goal was to gather as much data as possible, so it was necessary to combine different encoder settings and encode all videos with these different settings. The whole process resulted in thousands of clips and statistical files.

The goal was to test the impact of different bit rates starting at 100 kBit/s up to 1500 kBit/s, as well as different GOP lengths, beginning with 25 up to 325 and different buffer sizes to cover common bandwidth limitations, eg. EDGE, 3G and HSDPA. All settings were tested on several resolutions starting at 320p and going up to 1080p.

4.4. Encoder Settings

Below a list of the used encoder settings and a short description of each encoder statement according to [56] is given:

- **profile** as described in chapter 3 section 3.4.6 is a set of predefined settings that should cover different areas of application. I used the “baseline” profile for encoding, which is designed for lower complexity with limited computing resources and covers the following settings:

⁶http://compression.ru/video/quality_measure/video_measurement_tool_en.html

⁷<https://www.eclipse.org/>

⁸<http://www.r-project.org/>

- **no-8x8dct** disables the 8x8 DCT.
- **bframes** normally sets the maximum number of B-frames between an I- and P-frame - in this case B-frames were deactivated.
- **no-cabac**: CABAC (Context Adaptive Binary Arithmetic Coder) is the normally used x264 compression method and improves the efficiency within the baseline profile. This method is not used and replaced by the CAVLC coding.
- **cqm** sets the quantization matrix. Used: flat
- **weightp**: the used value 0 disables the use of explicit weighted prediction for B-frames.
- **preset** is a template of settings to exchange encoding speed with compression efficiency (eg.: file size or quality). Preset “fast” was used in this experiment to cover the following settings:
 - **ref** makes it possible to specify how many reference frames can be used. Used: 2
 - **rc-lookahead** defines the number of frames used for mb-tree ratecontrol and vbv-lookahead. Used: 30
 - **subme** controls the subpixel estimation strength. Used: 6
- **threads** specifies the number of parallel used threads while encoding. Used: auto
- **resize** The resize filter processes the input video before encoding and resizes the frames to the desired size. Used Values: 240, 360, 480, 720, 1080, Method: Bicubic
- **bitrate** tries to reach a target total bit rate in the final file in kBit/sec. Used: 100, 200, 400, 800, 1500.
- **keyint** sets the maximum interval between keyframes, also known as GOP length and indicates the maximum distance between I-frames. Used keyint values were 25, 75, 125, 175, 225, 275 and 325.
- **vbv-maxrate** defines the maximum local bit rate. Used: 0, 80, 220, 380, 800, 1000 kBit/s.
- **vbv-bufsize** sets the buffer size of the VBV (Video Buffer Verifier). Used: vbv-maxrate/2.

- **pass** enables multipass encoding. The first pass creates a stats file, the second one reads this stat file, which contains information about every input frame and creates the output video from this stat information. Used: 1 and 2
- **stats** defines the output folder and filename of the stats file containing information on every input frame.
- **psnr** enables the PSNR calculation, which will be logged after the end of encoding.
- **ssim** enables the SSIM calculation, which will also be logged at the end of encoding.
- **output** defines the output filename and folder. Per default the values are stored in a “.264” stream but can also be stored in an “.mp4”, matroska or flash container. I used the “.mp4” container for the output file.
- **input file** was in this case an “.avs” file referencing the “.avi” footage.

To figure out the influence of the different settings on the objective video quality a combination of resolution, bit rate, GOP length and buffer size settings were tested with each run. That leads to 1050 Hypothetical Reference Circuits (HRC) and multiplied with the 22 source sequences (SRC) the number of encoded videos will reach 23000 files.

Some sample encoder calls, encoder outputs, stat files as well as a sample “.avs” file can be found in section 4.6.

4.5. Footage

As footage for this experiment I took a representative sample of 22 video clips from the Made for Mobile database presented in [57], which was available at the time when I started working on my research. All files were recorded in two different versions - one according to the content creation guidelines described in [58] and a second one not considering these guidelines for mobile usage. The clips were produced with high-definition equipment from a professional media production company.

The mobile version prefers close-ups for static speakers, larger subtitle fonts for interviews and a shorter duration of each shot [57]. The raw clips were stored in a “.mov” container encoded with the Avid DNxHD codec. For easier handling some scenes with different content and a length no longer than 20 seconds were extracted and saved in an “.avi” container. As there is no relevance for audio in this experiment the sound of the video files was deleted.

In general, the footage captures the life of a dog in an animal shelter. The dog is adopted by three different owners but never stays long with his new parents. He always returns to the shelter as he fell in love with the animal keeper there. Tables 4.2 and 4.3 give an overview of the different clips and technical details of the videos used in this experiment.

Framerate	25 fps
Video Codec	YUV 4:2:0 uncompressed
Container Format	" .avi"
Bit rate	200 kBit/s
Resolution	1920 × 1080

Table 4.2.: Video details of the source footage

4.6. Encoding Process

The experiment was initiated with an automated script via a C# command line tool with the settings as defined in section 4.4. After starting the script everything ran fully automated.

To monitor the whole automated process a log file with a timestamp at the beginning and the end of each encoding process was generated.

Since it is not possible to load footage in the ".avi" container, which was in this case the base footage, directly to the x264 encoder an avisynth ".avs" script had to be generated and used as input file. The .avs file references the base footage file and Avisynth opens the file. Listing 4.1 shows a sample script.

```
AviSource("D:\Footage\video1.avi")
```

Listing 4.1: Sample AVS input file

After creating an avisynth script for the input clip the encoding process with different arguments can be started directly from the console. The listings 4.2 and 4.3 give examples of how the statement for a two pass encoding step can look like. In that case the target resolution will be 1920 × 1080 pixels as there is no resize statement in there. Listings 4.4 and 4.5 show how to use the resize filter.

```
x264.exe --profile baseline --preset fast --threads auto --bitrate 100 --keyint 25
--vbv-maxrate 80 --vbv-buFSIZE 40 --pass 1 --stats "D:\Statfiles\video1_pass1.stats"
--psnr --ssim -o NUL "D:\Footage\video1.avs"
```

Listing 4.2: Sample x264 encoder call 1st pass resolution: 1080p

Filename	File size (GB)	Duration (seconds)	Mobile Classic	Description
10 – <i>video_a</i> – 01 – <i>sami</i> – <i>ep01 – var01</i>	0.95	13	Mobile	2 text passages and motion
10 – <i>video_b</i> – 02 – <i>sami</i> – <i>ep01 – var02</i>	1.11	15	Classic	
11 – <i>video_a</i> – 01 – <i>sami</i> – <i>ep01 – var01</i>	0.79	10	Mobile	2 intro text passages no motion dog lying on the floor
11 – <i>video_b</i> – 02 – <i>sami</i> – <i>ep01 – var02</i>	0.99	13	Classic	
12 – <i>video_a</i> – 01 – <i>sami</i> – <i>ep01 – var01</i>	1,15	16	Mobile	high motion - running dogs and human feet
12 – <i>video_b</i> – 02 – <i>sami</i> – <i>ep01 – var02</i>	0.94	13	Classic	
13 – <i>video_a</i> – 01 – <i>sami</i> – <i>ep01 – var01</i>	0.82	11	Mobile	low motion grass details
13 – <i>video_b</i> – 02 – <i>sami</i> – <i>ep01 – var02</i>	1.03	14	Classic	
14 – <i>video_a</i> – 01 – <i>sami</i> – <i>ep01 – var01</i>	0.56	7	Mobile	low motion only 2 camera positions
14 – <i>video_b</i> – 02 – <i>sami</i> – <i>ep01 – var02</i>	0.98	13	Classic	
15 – <i>video_a</i> – 05 – <i>sami</i> – <i>ep02 – var01</i>	0.92	12	Mobile	birthday cake with lightning candles low motion
15 – <i>video_b</i> – 06 – <i>sami</i> – <i>ep02 – var02</i>	0.80	11	Classic	
16 – <i>video_a</i> – 05 – <i>sami</i> – <i>ep02 – var01</i>	0.72	9	Mobile	2 camera positions almost no motion
16 – <i>video_b</i> – 06 – <i>sami</i> – <i>ep02 – var02</i>	0.82	11	Classic	
17 – <i>video_a</i> – 05 – <i>sami</i> – <i>ep02 – var01</i>	1.03	14	Mobile	high motion - dog gone wild
17 – <i>video_b</i> – 06 – <i>sami</i> – <i>ep02 – var02</i>	0.63	8	Classic	
18 – <i>video_a</i> – 09 – <i>sami</i> – <i>ep03 – var01</i>	1.04	14	Mobile	steady motion - walking dog and a couple
18 – <i>video_b</i> – 10 – <i>sami</i> – <i>ep03 – var02</i>	1.04	14	Classic	
19 – <i>video_a</i> – 09 – <i>sami</i> – <i>ep03 – var01</i>	0.92	12	Mobile	high motion scenes - the dog is hunting a cat
19 – <i>video_b</i> – 10 – <i>sami</i> – <i>ep03 – var02</i>	0.99	13	Classic	
20 – <i>video_a</i> – 09 – <i>sami</i> – <i>ep03 – var01</i>	1.02	14	Mobile	steady motion containing credits
20 – <i>video_b</i> – 10 – <i>sami</i> – <i>ep03 – var02</i>	0.99	13	Classic	

Table 4.3.: Overview and description of the source footage

```
x264.exe --profile baseline --preset fast --threads auto --bitrate 100 --keyint 25
--vbv-maxrate 80 --vbv-buftype 40 --pass 2 --stats "D:\Statfiles\video1_pass1.stats" --psnr
--ssim -o "D:\Results\video1_1080p_100kbit_25gop_80buftype.mp4" "D:\Footage\video1.avs"
```

Listing 4.3: Sample x264 encoder call 2nd pass resolution: 1080p

```
x264.exe --profile baseline --preset fast --threads auto
--video-filter resize:width=640,height=360,method=bicubic --bitrate 100 --keyint 25
--vbv-maxrate 80 --vbv-buftype 40 --pass 1 --stats "D:\Statfiles\video1_pass1.stats"
--psnr --ssim -o NUL "D:\Footage\video1.avs"
```

Listing 4.4: Sample x264 Encoder call 1st pass resolution: 360p

```
x264.exe --profile baseline --preset fast --threads auto
--video-filter resize:width=640,height=360,method=bicubic --bitrate 100 --keyint 25
--vbv-maxrate 80 --vbv-buftype 40 --pass 2 --stats "D:\Statfiles\video1_pass1.stats"
--psnr --ssim -o "D:\Results\video1_1080p_100kbit_25gop_80buftype.mp4"
"D:\Footage\video1.avs"
```

Listing 4.5: Sample x264 Encoder Call 2nd pass resolution: 360p

The “.stat” file generated for each first pass contains a summary of the encoder settings used, followed by a line with information (framenumbers, frametypes, number of macroblocks, number of bits spent) on each frame shown for example in listing 4.6. These files are the input for a second pass encoding. The encoder uses the statistical values to reproduce the video clip and can reach a better compression ratio.

```
in:0 out:0 type:I dur:2 cpbdur:2 q:35.92 tex:71601 mv:25689 misc:5662 imb:1200 pmb:0;
in:1 out:1 type:P dur:2 cpbdur:2 q:37.21 tex:4435 mv:2602 misc:763 imb:54 pmb:204;
in:2 out:2 type:P dur:2 cpbdur:2 q:37.21 tex:6857 mv:3468 misc:955 imb:134 pmb:252;
in:3 out:3 type:P dur:2 cpbdur:2 q:37.21 tex:5625 mv:3739 misc:996 imb:143 pmb:257;
in:4 out:4 type:P dur:2 cpbdur:2 q:37.21 tex:6207 mv:4163 misc:1086 imb:165 pmb:273;
in:5 out:5 type:P dur:2 cpbdur:2 q:37.30 tex:7741 mv:4188 misc:1119 imb:200 pmb:272;
.
.
```

Listing 4.6: Sample .stat file

The whole encoding process lasted about 207 hours. 46200 encoding passes generated 23100 video clips with a total size of about 20 GB.

4.7. Comparison Process

Seeing as the x264 encoder is also able to calculate PSNR as well as SSIM, I used this output for the PSNR and SSIM values. An example of what this looks like can be found in listing 4.7.

```

x264 [info]: frame I:14      Avg QP:33.67  size:  3701  PSNR Mean Y:33.42 U:39.31 V:42.04
Avg:34.68 Global:33.57
x264 [info]: frame P:316      Avg QP:40.82  size:   309  PSNR Mean Y:31.89 U:38.92 V:41.47
Avg:33.24 Global:31.69
x264 [info]: mb I   I16..4:  65.7%   0.0%  34.3%
x264 [info]: mb P   I16..4:   7.6%   0.0%   0.0%  P16..4: 10.8%   0.0%   0.0%   0.0%   0.0%
skip:81.6\%
x264 [info]: final ratefactor: 34.51
x264 [info]: coded y,uvDC,uvAC intra: 33.1% 25.5% 7.7% inter: 2.1% 1.0% 0.0%
x264 [info]: i16 v,h,dc,p: 63% 19% 13% 6%
x264 [info]: i4 v,h,dc,ddl,ddr,vr,hd,vl,hu: 38% 27% 17% 2% 4% 3% 3% 3% 3%
x264 [info]: i8c dc,h,v,p: 77% 8% 14% 1%
x264 [info]: SSIM Mean Y:0.8960071 ( 9.830db)
x264 [info]: PSNR Mean Y:31.956 U:38.938 V:41.492 Avg:33.297 Global:31.752 kb/s:90.56

```

Listing 4.7: Sample x264 encoder output

To get all results in one file a script which parsed all of the 46200 files (first pass encoding also generates statistical values) helped me to merge the SSIM and PSNR in a “.csv” file, which can be processed in further use without problems.

For calculating the VQM value I used the powerful MSU Tool command line interface and automated the calculation through an C# script.

Therefore, the statement listed in section 4.8 is used for each encoded video and the referencing unimpaired input file in the right resolution.

```

msu_metric.exe -f "D:\Footage\video1.avi"
-f "D:\Results\video1_1080p_100kbit_25gop_80buffsize.mp4" -metr vqm -cc ALL -sc 1
-cod "D:\MSUTest\Stats" -lp "D:\MSUTest\Stats\video1_compare_error.txt"
-af2 "D:\MSUTest\Stats\video_compare_average.txt" -sv 0

```

Listing 4.8: Sample MSU Tool statement for VQM calculation

The parameter “-metr” indicated the metric for calculation. As SSIM and PSNR were computed through the encoder the MSU Tool was used for VQM calculation. More metrics can be listed with comma. “-cc” indicates the color components for which the calculation should be performed. “-sc 1” indicates that a “.csv” file with the results should be saved at the path specified in “-cod”. “-lp” and “-af2” specify the path and filenames for error messages, such as “could not open file” or “could not calculate the metric”. The parameter “-sv” indicates if a visualization file has to be saved or not.

Each statement generated a “.csv” file with the resulting VQM values, an example is given in listing 4.9.

```

VQM_YYUV
D:\Footage\....avi
D:\Results\..._1080p_100kbit_25gop_80buffsize.mp4
AVG: 1.39900
1.62744

```

```
1.88285
2.07602
1.94790
.
.
.
```

Listing 4.9: Sample MSU Tool output for VQM calculatiuon

Each measurement corresponds to one column in the CSV file. The first line describes the metric and the color component used for the comparison. The second and third line indicate the name of the compared files. The fourth line contains an averaged result of the whole video clip. The following lines contain the VQM value for each frame in the compared videos.

I joined the average values to the “.csv” file generated above with the PSNR and SSIM values for further use in any statistical tool.

A detailed description of all parameters and all functions of the MSU Tool can be downloaded from the website of the developers. ⁹

4.8. Problems

As described in section 4.5 the raw material, produced with high-definition equipment was exported in a “.mov” container compressed with Avid DNxHD codec with an average bit rate of 1020 Mbit/s and a resolution of 1920×1080 pixels and a frame rate of 25 frames per second. All clips have an average length of about three minutes. The file size of each clip lies between 21 and 22 GB. First of all I started to test the whole process with these files but unfortunately it was almost impossible to get results in an acceptable amount of time. The encoding process of one video with only one pair of settings lasted about 50 minutes and was very unstable which unfortunately resulted in many crashes, incomplete videos and artifacts due to processing problems. The same problem occurred when calculating metrics with the MSU Tool, which lasted about 65 minutes for one pair. Therefore, I decided to use the 22 clips described in section 4.5 that were already available at this point of time for another experiment. They contained very interesting scenes extracted from the original footage and are still available in 1080p resolution.

⁹http://compression.ru/video/quality_measure/src/MSU_VQMT_Documentation.pdf

Chapter 5.

Experimental Results

In the following chapter the results gathered during the experiment as described in chapter 4 are analyzed. The aim is to figure out which parameter settings have an impact on the objective video quality and how distortions affect visual quality, as well as which encoder settings could be used for producing a good quality. Section 5.1 gives a short overview of the duration of the encoding process while section 5.2 examines the performance of the file size in relation to the different HRC's. Section 5.3 shall help to answer the question which of the tested settings have a major impact on the perceived video quality and above all how much will the videos be distorted. This section also gives an overview on how the values of the objective methods can be classified to make the results easier to compare.

5.1. Analysis of Encoding Duration

This section analyzes the duration of the encoding process for all HRC's. The overall encoding process for all videos with every setting combination lasted 207 hours, which are more than eight days for encoding the video files.

As you can see in figure 5.1, which presents the data of a first pass encoding with a bit rate of 400 kBit/s, a buffer size of 380 kBit and a GOP length of 125 frames, as well as in figure 5.3 the video clip 14 – *video_a* – 01 – *sami* – *ep01* – *var01* was the one that performed very well in terms of encoding speed while encoding of the clip 12 – *video_a* – 01 – *sami* – *ep01* – *var01* always took significantly longer than the others. Figure 5.2, which shows the file size of the original input footage gives the reason for that. The reference file of clip 12 – *video_a* – 01 – *sami* – *ep01* – *var01* has the biggest file size with 1.15 GB and 16 seconds duration compared to the clip 14 – *video_a* – 01 – *sami* – *ep01* – *var01*, which has only a duration of 7 seconds and a file size of 0.56 GB. So in summary, the encoding speed depends on the file size of the footage that is used as input video. Figure 5.3 shows the encoding duration of all videos encoded with a bit rate of 400 kBit/s, a buffer size of 1000 kBit, and a

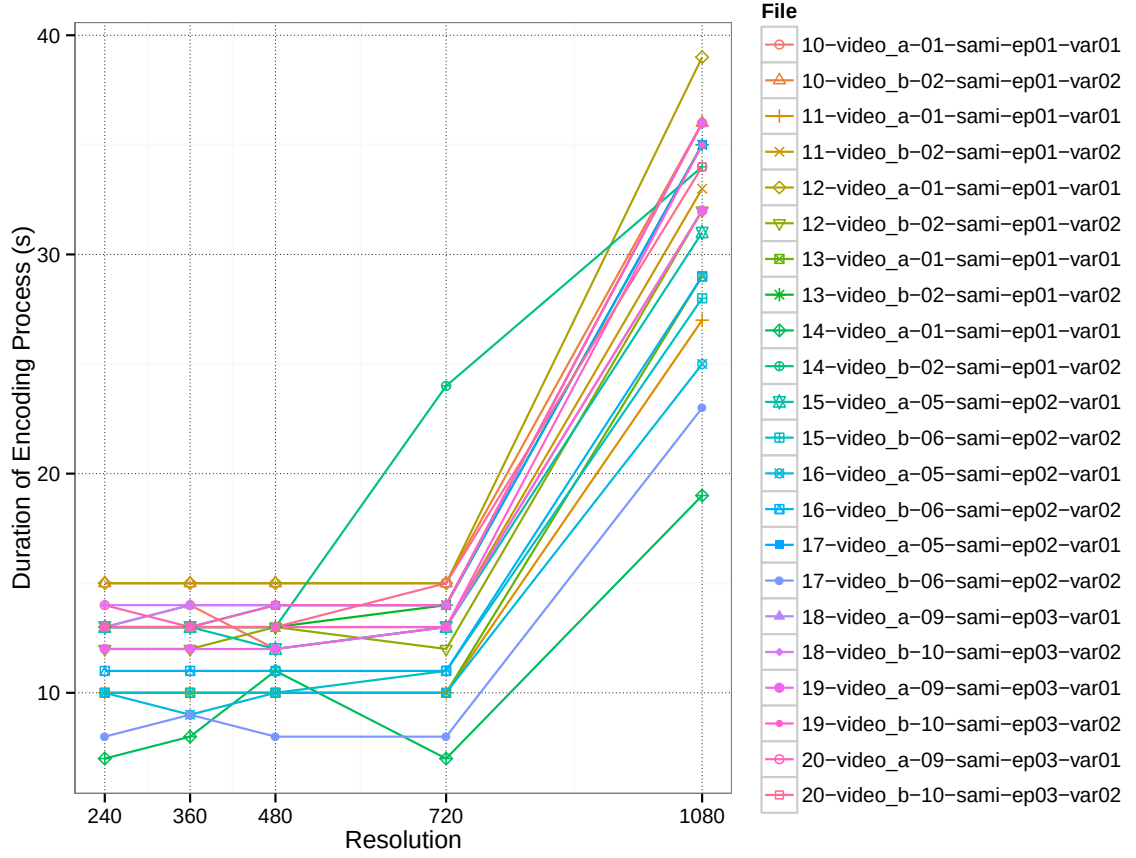


Figure 5.1.: Duration of encoding 1st pass

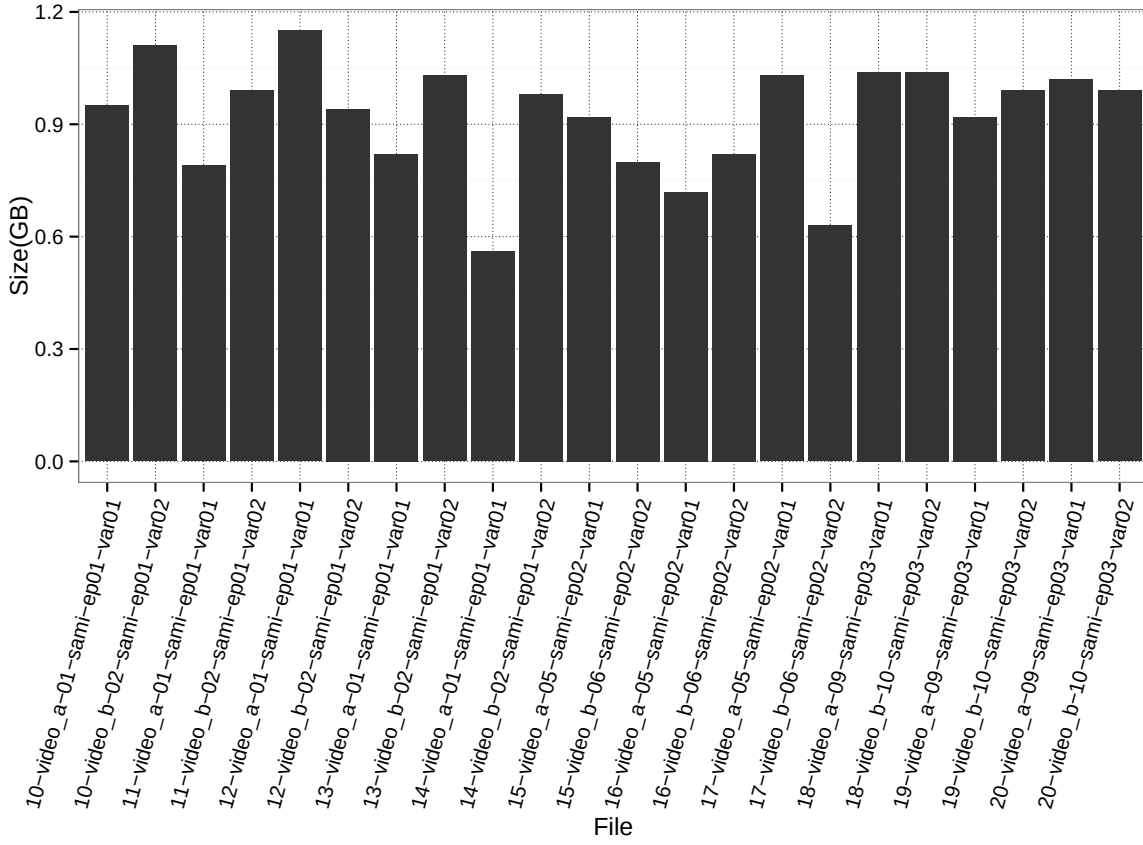
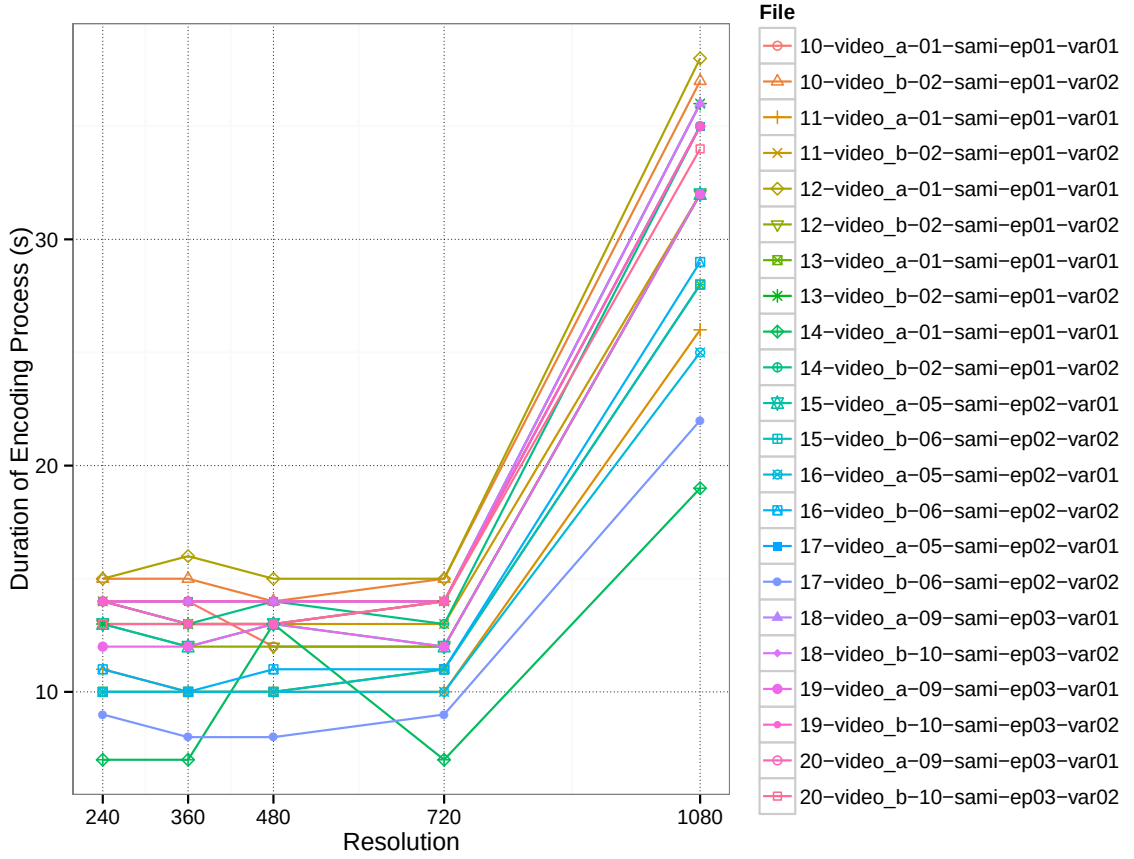


Figure 5.2.: File size of the input footage

Figure 5.3.: Duration of encoding 2nd pass

GOP length of 325 frames in the second pass for all available resolutions. Different plots for other HRC combinations show nearly the same results so this figure is exemplary for other setting combinations. The minimum duration for encoding a file was 7 seconds while the longest time needed for encoding was 44 seconds. There is no difference regarding the duration in first and second pass encoding, they both need more or less the same amount of time. A big increase in encoding duration can be noted for high resolution videos. The mean duration is about 31 seconds while the mean encoding duration of the low resolution video clips is only about 12 seconds.

5.2. Analysis of Resulting File Size

Does a reducing of the bit rate or resolution of video clips have an impact on the resulting file size? This question will be answered with the following figures. In general, the file size of the encoded video files varies between 0.06 MB and 2.9 MB with a mean storage

amount of 0.88 MB per video. Summing up the total of about 20 GB of video clips was generated in this experiment, distributed among 23100 clips.

Figure 5.4 shows the file size of a 480p clip at different bit rates and presents very well how encoding of a clip with half of the bit rate results in a file size that is only half of the file with higher bit rates. While files with a bit rate of 1400 kBit/s have an average size of about 2.1 MB, a file with a bit rate of only 400 kBit/s results in an average size of about 0.6 MB.

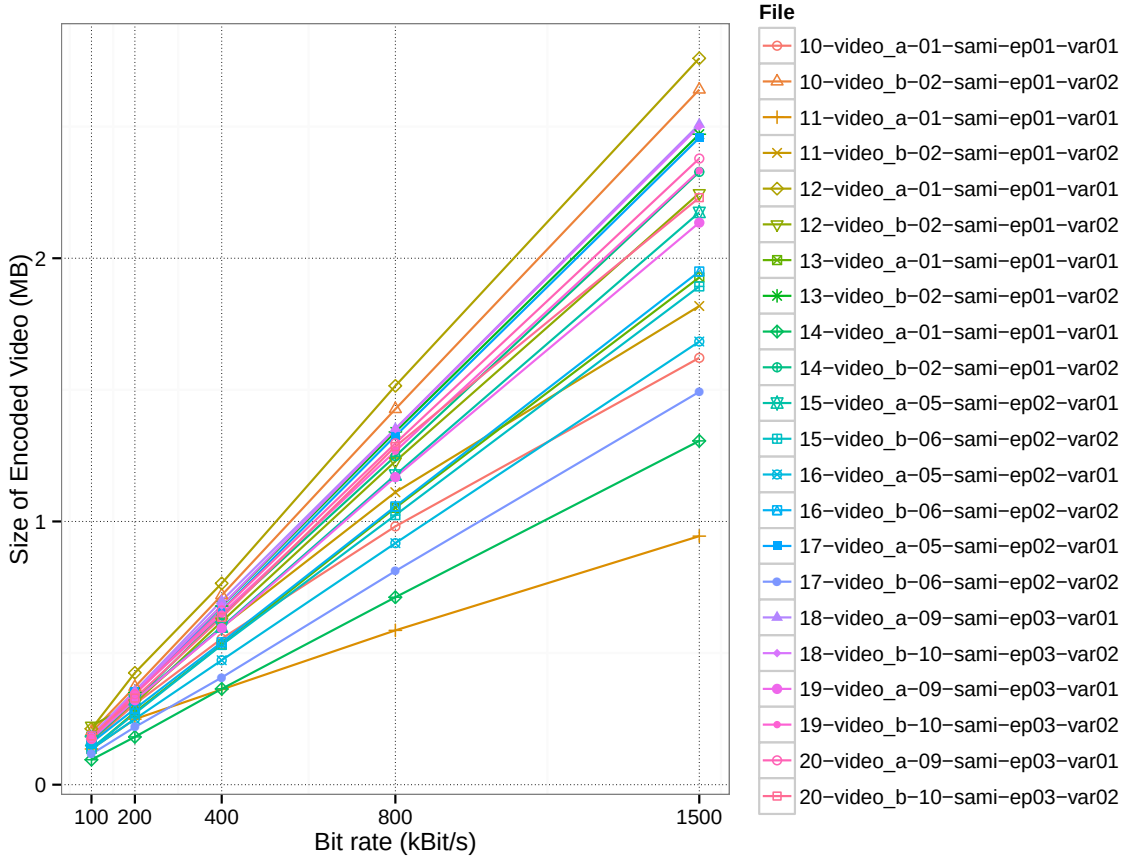


Figure 5.4.: Impact of bit rate reduction on the file size

While the reduction of the bit rate does impact the resulting file size as shown in figure 5.4, the reduction of the resolution does not seem to influence the size of the end result - a plot of the data showing a sample at bit rate 400 kBit/s, a GOP length of 25 frames and a buffer size of 220 kBit/s is presented in figure 5.5. It is remarkable that according general consideration reducing the resolution does not lead to significantly smaller file sizes of the encoded files - the size nearly stays the same if no other factors, such as a reduction of the bit rate, apply. Figure 5.5 also shows two results that do not fit to this assumption. The clips $12 - video_a - 01 - sami - ep01 - var01$ and $12 - video_b - 02 - sami - ep01 - var02$ do have a significantly higher file size than all other clips encoded with the same settings.

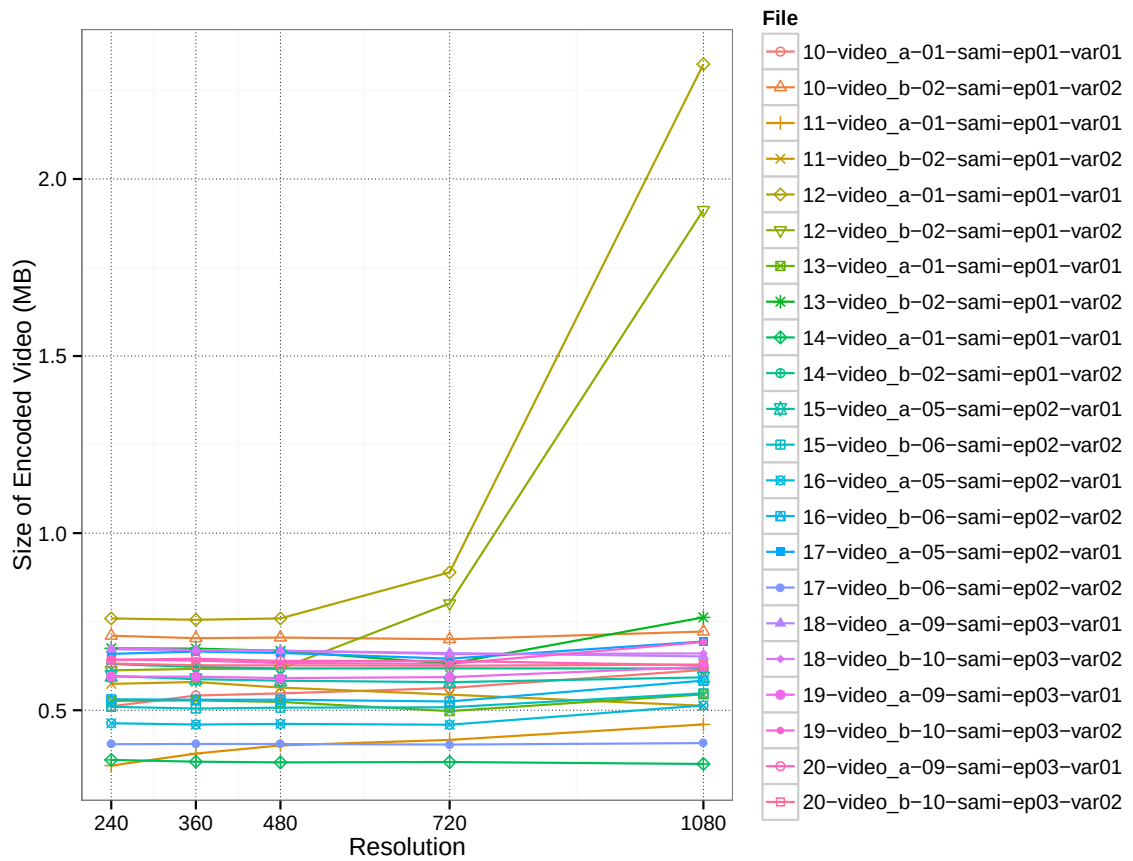


Figure 5.5.: Impact of resolution reduction on the file size

According to the description of the base footage in table 4.3 this pair of clips contains high motion for the full length of the video. The type of the content might be an explanation for why the file size of these two videos does not decrease in the same way as all other clips while reducing the resolution. Reducing the GOP length does not lead to a

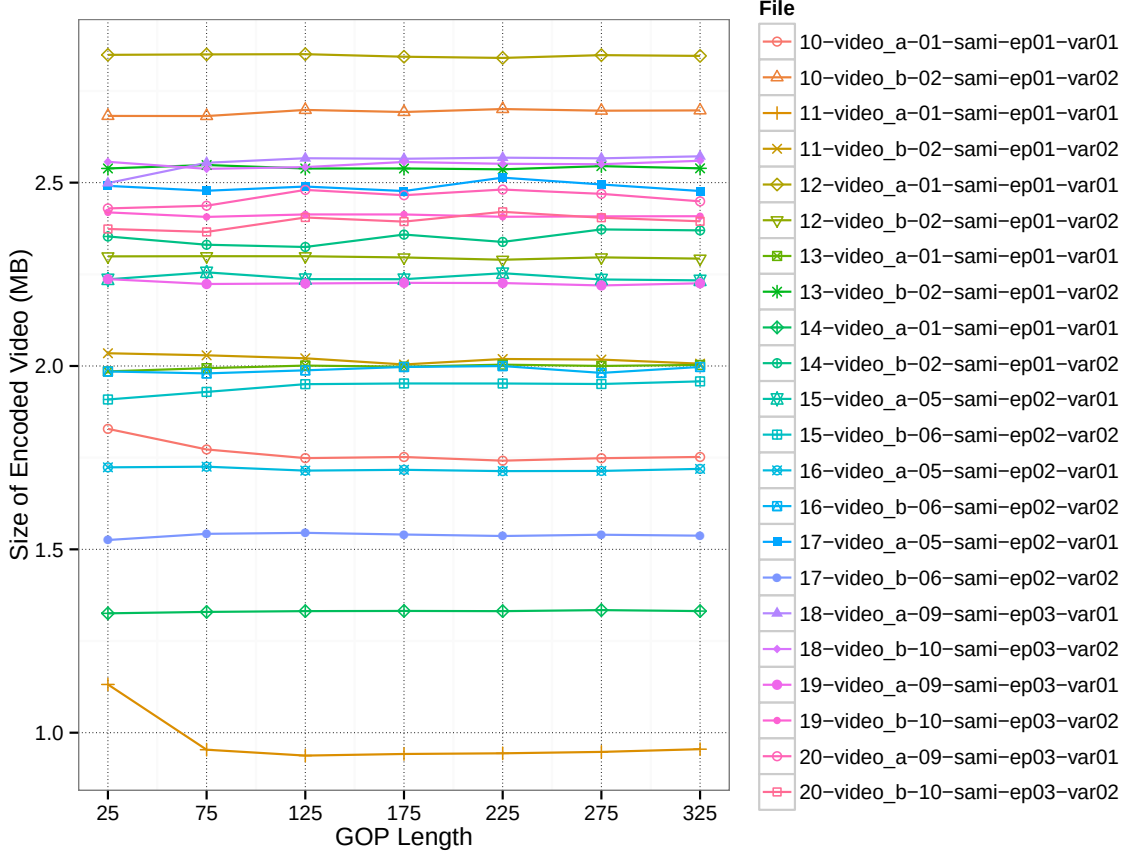


Figure 5.6.: Impact of variation in the GOP length on the file size

significant reduction of the file size as shown in figure 5.6 either. The result is interesting because normally I-frames, which are encoded without reference to another frame, are much bigger than P-frames since they contain more information and should also require more space. Therefore, an increase of the GOP length should normally go along with a reduction of the file size. A description of how GOP length and frame encoding work is given in chapter 3 in section 3.4.4

Figure 5.7 shows that reducing the buffer size also has a minor impact on the resulting file size.

According to the plots seen in this section only a reduction of the bit rate has a significant impact on the resulting file size, every other parameter does not lead to smaller files as the file sizes remain the same even when using different HRC combinations.

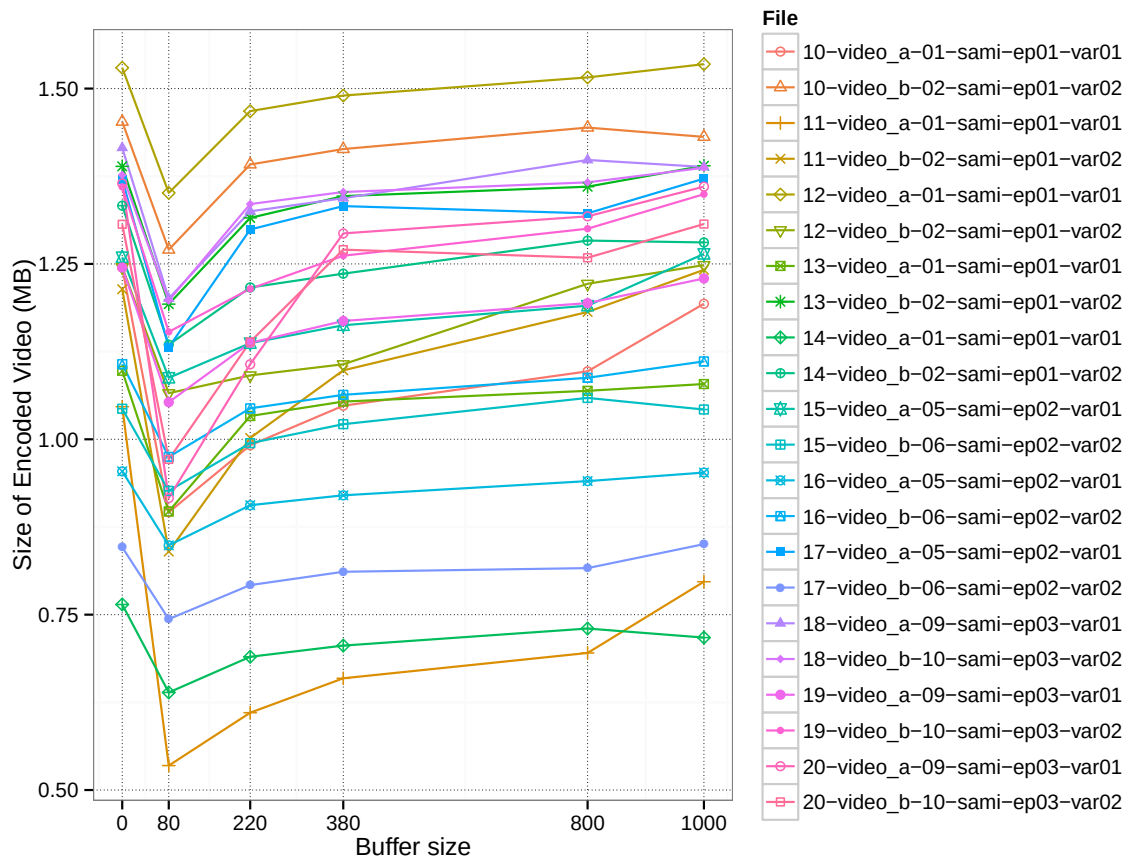


Figure 5.7.: Impact of different buffer sizes on the file size

5.3. Analysis of Objective Evaluation Results

The following section summarizes the results of the data gathered by means of objective quality assessment methods. I used the metrics PSNR, SSIM and VQM, which are described in chapter 3. The calculation of PSNR and SSIM was performed by the x264 encoder while VQM was calculated with the help of the MSU VQM Tool, all tools and sample statements are mentioned in chapter 4.

The outcome of the PSNR calculation are values between 0 and 100 dB indicating the mean error of the reference and the output video. The SSIM values might range from -1 to 1, representing the structural similarity between two signals while the calculation of the general Video Quality Metric results in values greater than 0, where 0 represents equal frames.

For categorizing the results there are several mapping methods available to translate the objective results, which are normally only values within a different range and might be confusing, to subjective scores. These scores are also used in subjective evaluations, which are in my opinion much easier to understand and much better to compare than a single value without a reference. According to [26] subjective quality is usually expressed on a five point scale 1-Bad, 2-Poor, 3-Fair, 4-Good, 5-Excellent, table 5.1 shows an overview of how it is possible to map the objective results to the subjective scores. The scheme for mapping the PSNR and SSIM was proposed in [13], [12] published the mapping of the VQM.

MOS	PSNR	SSIM	VQM
5 (Excellent)	≥ 45.0	≥ 0.99	< 2
4 (Good)	$\in [33, 45)$	$\in [0.95, 0.99)$	$\in [2, 4)$
3 (Fair)	$\in [27.4, 33)$	$\in [0.88, 0.95)$	$\in [4, 6)$
2 (Poor)	$\in [18.7, 27.4)$	$\in [0.50, 0.80)$	$\in [6, 8)$
1 (Bad)	< 18.7	< 0.5	> 8

Table 5.1.: Mapping PSNR, SSIM and VQM to subjective MOS values

5.3.1. Impact of Bit Rate

The following figures show the impact of reducing the bit rate on the perceived video quality. Figure 5.8 represents the results of the PSNR calculation in relation to different bit rate values. A first look on the medians in the figure reveals that reducing the bit rate also leads to a loss in the perceived video quality measured through PSNR. According to the interquartile range the quality never seems to be brilliant but always in the range of acceptable quality. The number of outliers towards better and worse video quality in higher bit rates is remarkable. The quality seems to vary and is dependent on the

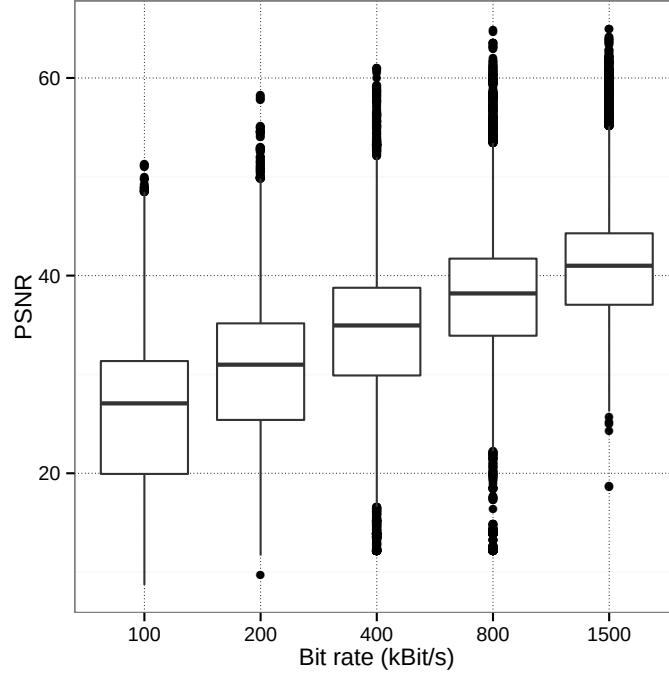


Figure 5.8.: PSNR for different bit rate values

content. Previous studies in this field [10] showed that the PSNR can be used as a quality estimation method when the content is fixed for the entire experiment.

Figure 5.9 shows the PSNR quality evaluation results of each clip at a resolution of 854×480 pixels. The visualized results show that the quality of files with the highest bit rate can all be classified between “Good” and “Excellent”, while the quality of files with a bit rate of about 100 kBit/s are only rated between “Fair” and “Poor”. There seem to be two videos that do have excellent quality even if the bit rate is reduced to a minimum. The pair of video clip 12 obviously performs very bad in low bit rate environments, increasing the bit rate leads to acceptable quality. According to table 4.3 pair 12 contains content with very high motion. While pair 12 performs bad in low bit rate environments there is also a pair of clips that works very well either with low or high bit rates. 11 – *video_a* – 01 – *sami* – *ep01* – *var01* as well as the corresponding classic version have significantly higher PSNR values at a bit rate of 100 kBit/s than all other clips. According to table 4.3 this pair of clips only contains very little motion and most of the time some text inserts are shown.

While the PSNR values in figure 5.8 confirmed a quite acceptable quality of the tested files upon reducing the bit rate, the results of the SSIM calculation in figure 5.10 show a slightly different picture. According to the mapping in table 5.1 the quality assessed through SSIM calculation is much better rated especially in higher bit rates. Bit rates higher than 800 kBit/s result in excellent quality while a bit rate of 100 kBit/s results

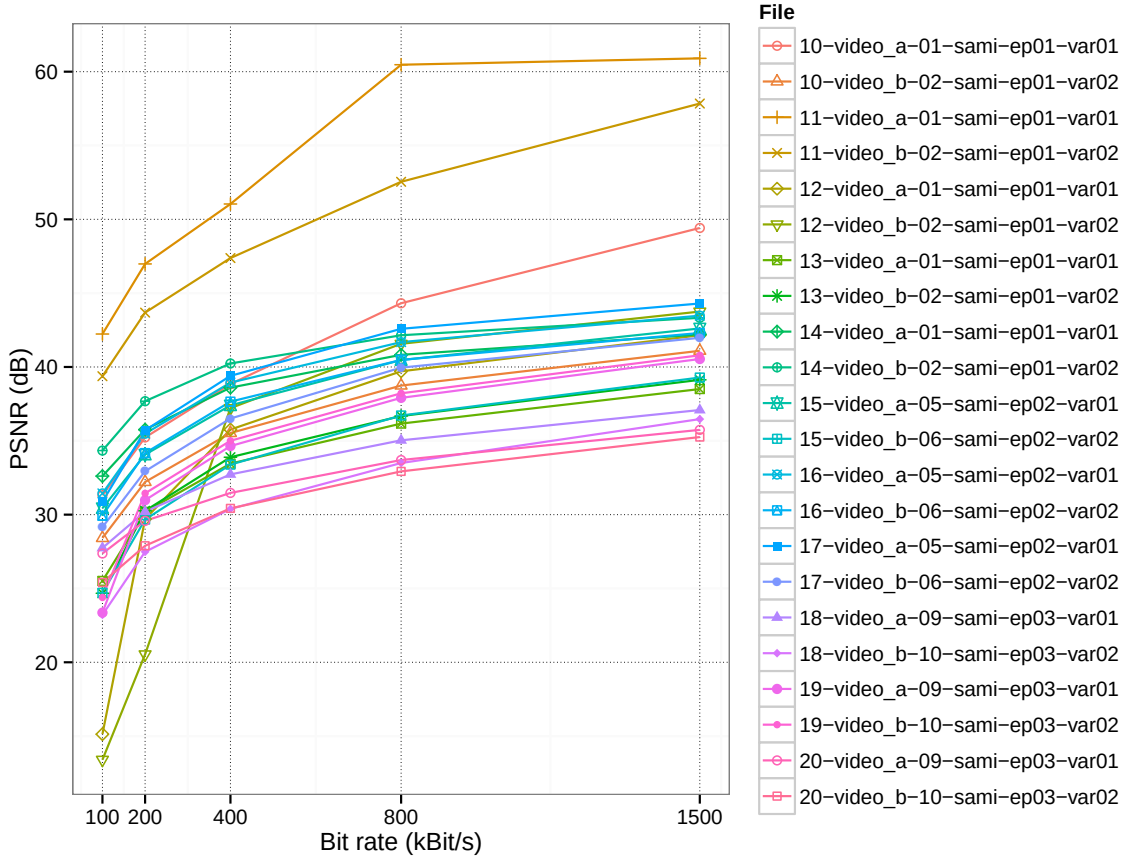


Figure 5.9.: PSNR for different bit rate values grouped by the input files

in poor video quality. The SSIM values validate the statement that reducing the bit rate results in lower perceived video quality. In general, the interquartile ranges of the higher bit rate results are smaller than in relation to lower bit rates which indicates a high variation in the values. The number of outliers is equally distributed among all bit rates.

Figure 5.11 shows the calculation results of the SSIM for the same subset as presented in figure 5.9. For the clips of video pair 11, which also performed very well in PSNR evaluation, the SSIM metric also confirmed the quality and reviewed these files with the best quality. While the PSNR calculation verified a very bad performance of the clips in pair 12, SSIM results did not rate this pair of clips with bad quality. SSIM confirms the clip 18 – *video_b* – 10 – *sami* – *ep03* – *var02* of having bad quality especially at the lower bit rates, with a significant difference in quality related to the clips with higher bit rates.

Figure 5.12 presents the results of VQM calculations and shows a very similar picture as figure 5.8 and 5.10. VQM values represent a better quality once the values are as close as possible to 0. The interquartile ranges are very small for bit rates over 800 kBit/s, which

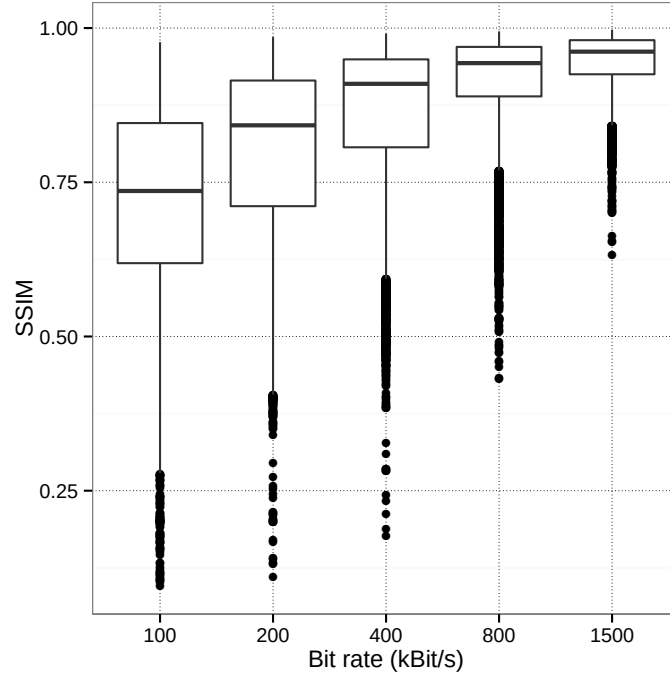


Figure 5.10.: SSIM for different bit rate values

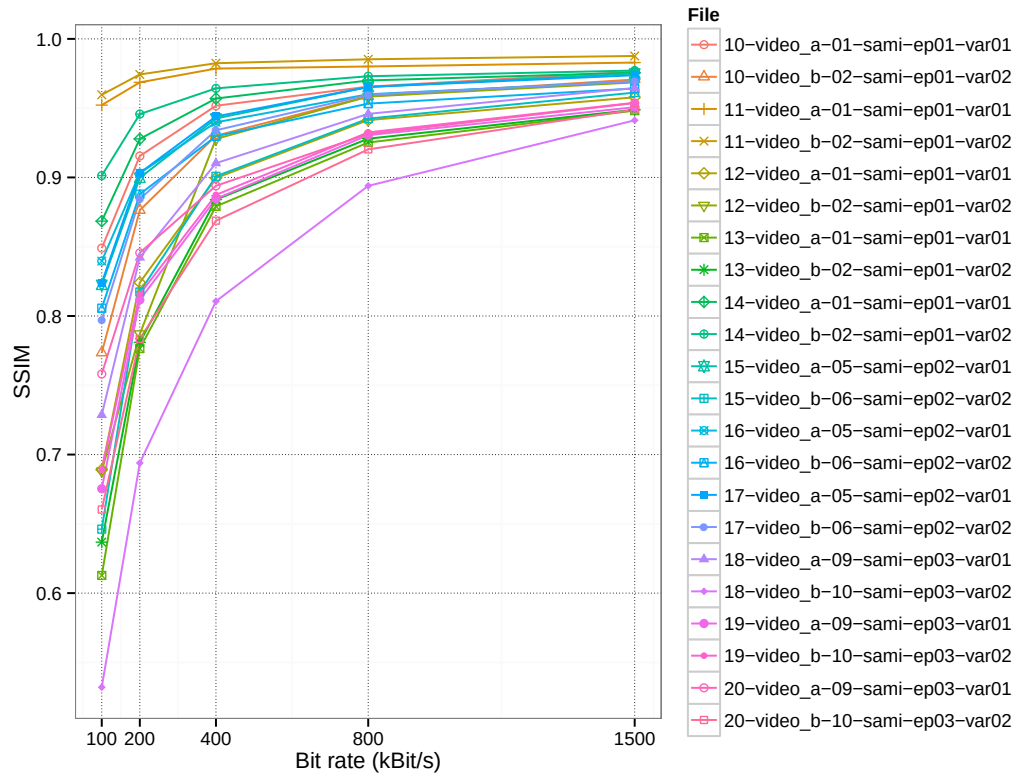


Figure 5.11.: SSIM for different bit rate values grouped by the input files

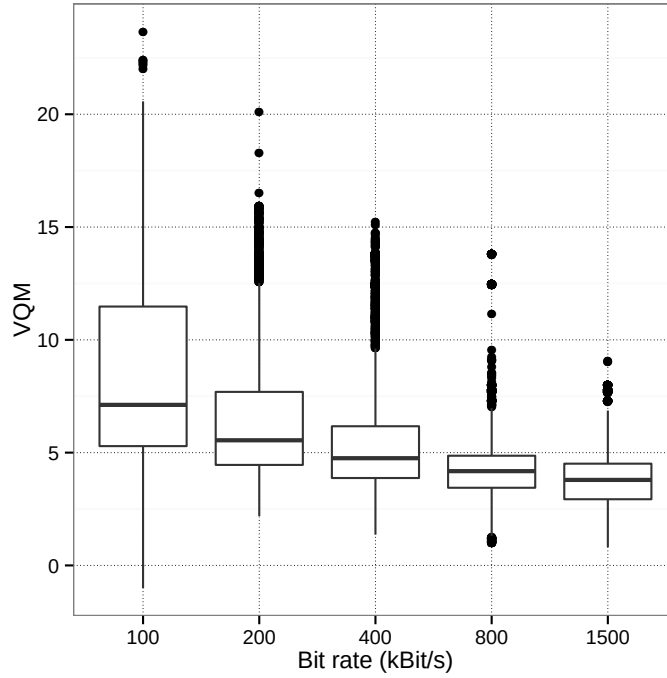


Figure 5.12.: VQM for different bit rate values

means that the results are quite consistent without a lot of variations. The number of outliers for example is quite high for bit rates of 200 - 400 kBit/s.

5.3.2. Impact of GOP Length

The impact of variations in the GOP length on the perceived video quality is described in this section.

While the reduction of the bit rate resulted in differences in the results of the objective calculations as shown in section 5.3.1, variations in the GOP length of a video file does not seem to impact the perceived video quality. The results for PSNR calculations are presented in figure 5.13, showing no difference throughout all GOP lengths. A smaller GOP length results in PSNR values that indicate steady good quality without significant changes in any other GOP lengths. The representation for each chosen GOP length has nearly the same spread and the same number of outliers, so unfortunately, no variation in terms of quality can be found. Since the GOP length that indicates the distance between consecutive I-frames was chosen to be very high there might be differences in the overall quality results when performing the same experiment with smaller GOP lengths, seeing as more I-frames, which are the base for each B- and P-frame, should lead to an increase in video quality.

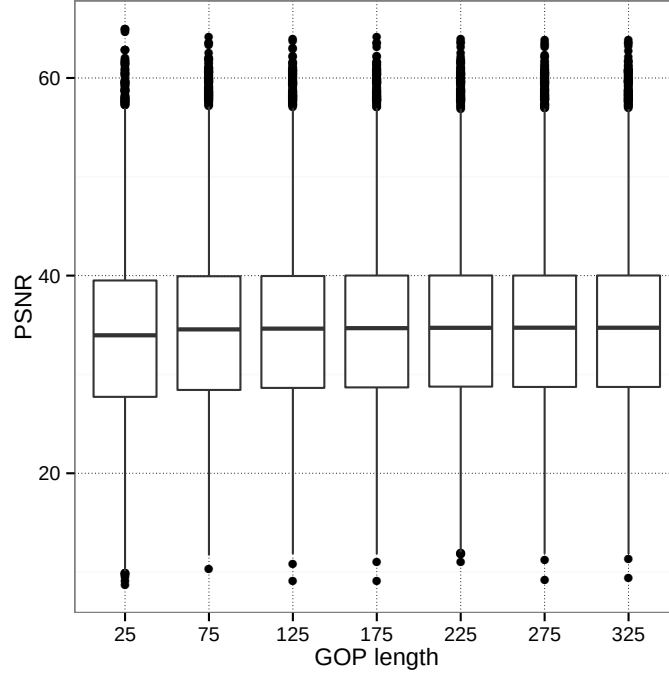


Figure 5.13.: PSNR for different GOP lengths

Figure 5.14 shows very well that there is hardly any difference in the perceived video quality at different GOP lengths. The subset presents the data of the clip 19 – *video_a* – 09 – *sami* – *ep03* – *var01* at a resolution of 720p. The influence of the bit rate on the perceived video quality, discussed in section 5.3.1, can be seen very clearly. The figure shows that there are slight differences in the GOP lengths in the video quality at lower bit rates, where the variations might be caused by the generally lower quality while there is simply no variation at higher bit rates.

The figures 5.15 - 5.17 are showing a more detailed plot of figure 5.13 with an insight in bit rates smaller or equal 400 kBit/s. As already seen in figure 5.14 the overall quality for clips with a bit rate of 100 kBit/s is between “Bad” and “Fair”. Figure 5.15 shows that the interquartile range is quite high for clips with a bit rate that low which means the results of PSNR calculations are varying and a general rating for that category is not possible.

As seen in figure 5.16 the overall quality of clips with a bit rate of 200 kBit/s is rated as “Fair” with slight differences in the interquartile range at GOP lengths of 25 frames. All other GOP lengths does not lead to different ratings regarding quality, noticeable is that the number of outliers towards excellent quality is higher than in the figure shown above.

Figure 5.17 shows that the quality for clips encoded with a bit rate of 400 kBit/s is rated as “Good” with a high number of outliers towards excellent and also bad quality.

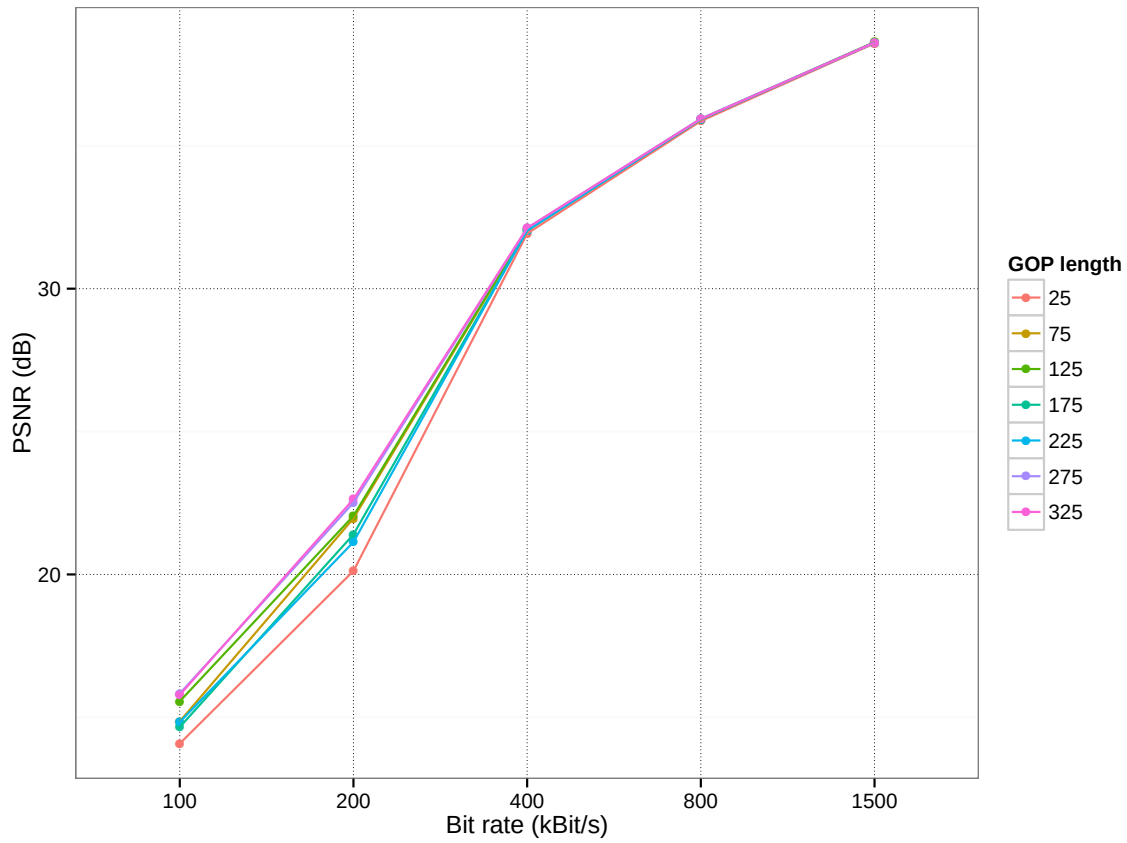


Figure 5.14.: PSNR showing the influence of the GOP length

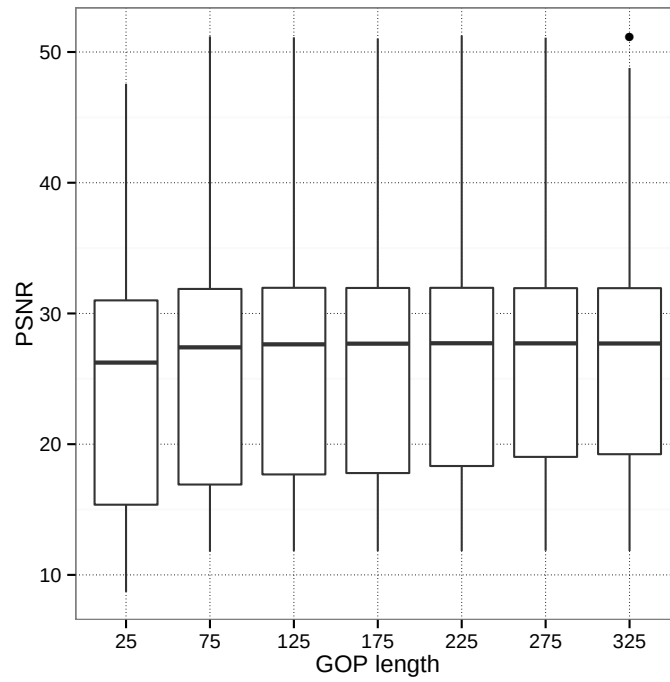


Figure 5.15.: PSNR for different GOP lengths, Bitrate: 100 kBit/s

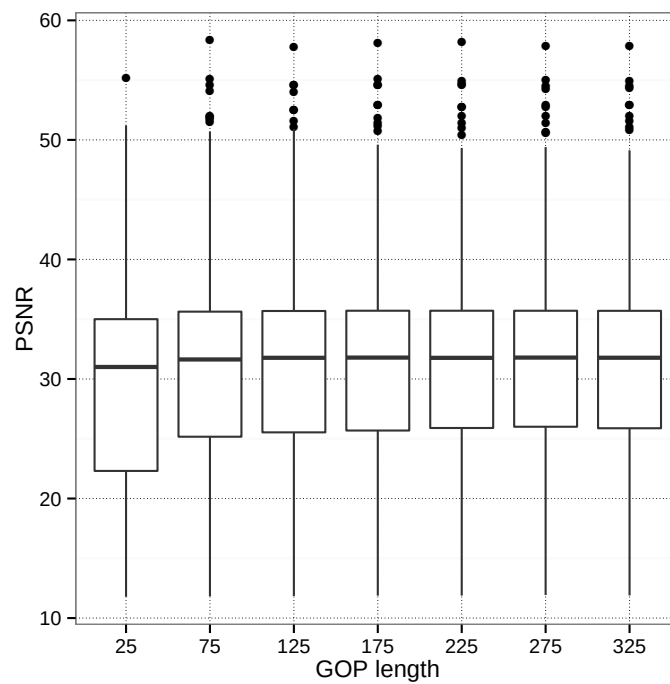


Figure 5.16.: PSNR for different GOP lengths, Bitrate: 200 kBit/s

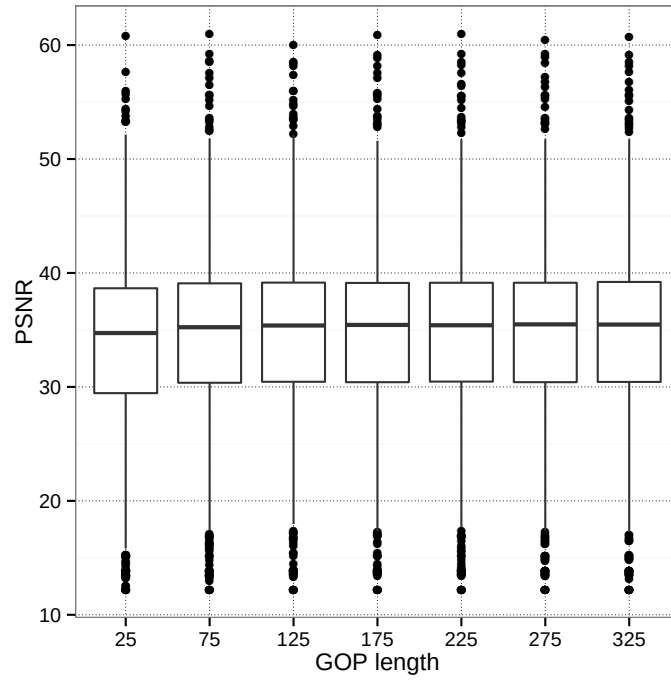


Figure 5.17.: PSNR for different GOP lengths, Bitrate: 400 kBit/s

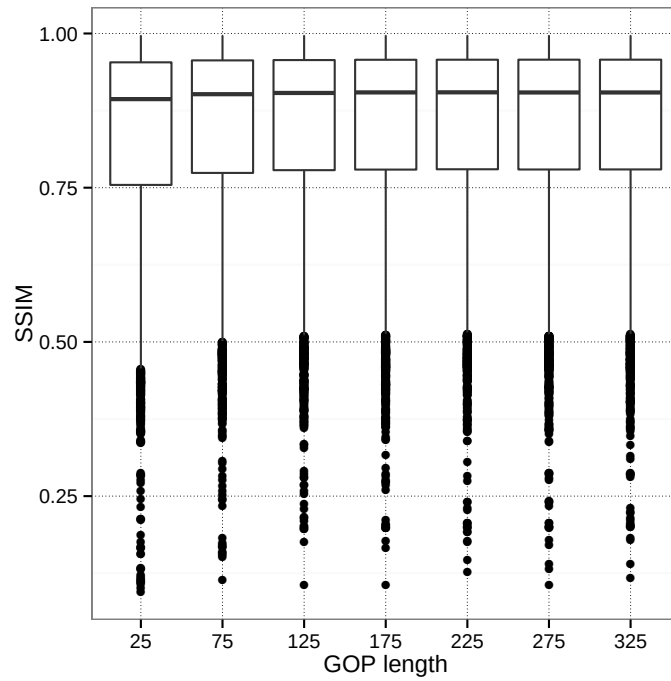


Figure 5.18.: SSIM for different GOP lengths

The results of the VQM and SSIM calculation shown in figure 5.19 and 5.18 nearly present the same picture - the results show that the overall quality can be determined as “fair” to “good” according to each algorithm without any differences in the variation of GOP lengths.

Considering the results of this section the length of a group of pictures does not have a significant influence on the video quality.

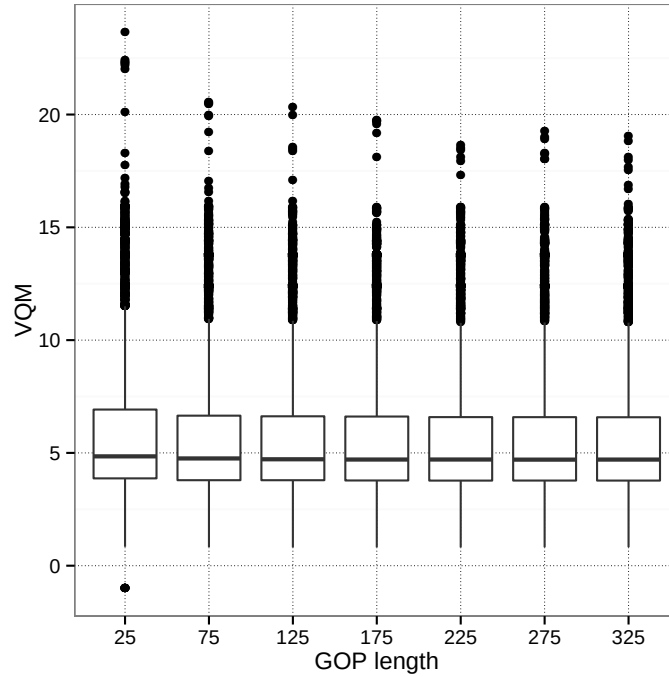


Figure 5.19.: VQM for different GOP lengths

5.3.3. Impact of Buffer size

The use of the VBV (Video Buffer Verifier) of the x264 encoder helps to constrain the output bit rate to make the video files suitable for streaming. The buffer works mainly on the decoder side, as data that is downloaded will first be saved in the VBV buffer and must be available there before the decoding process can start. Using a VBV buffer always involves two settings, first the size of each buffer which is specified in kBit and the other one is the rate at which the buffer is filled, specified in kBit/s. The values presented in the following figures on the x axis, always called “Buffer size”, were used for the parameter vbv-maxrate while the size of the vbv-buffer was chosen as half of the vbv-maxrate.

As shown in figure 5.20, 5.21 and 5.22 the buffer size does not have a major impact on the video quality. The results of each algorithm remain the same for each HRC without

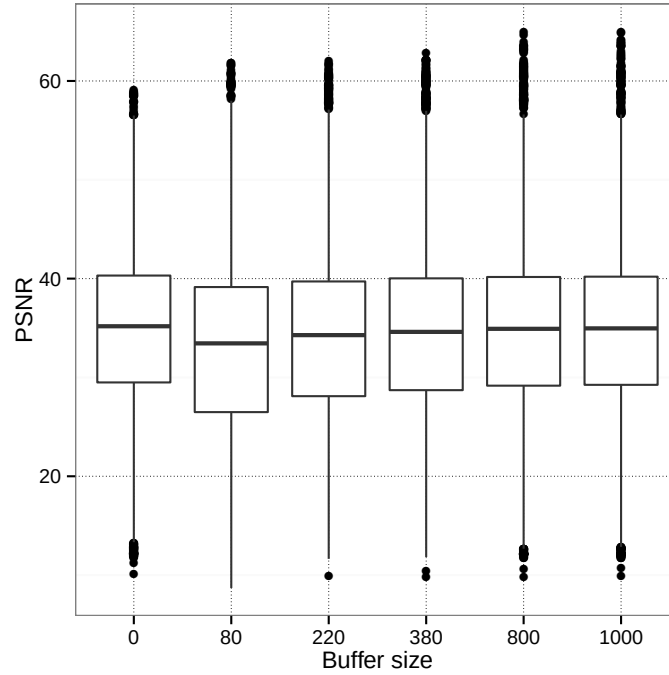


Figure 5.20.: PSNR for different buffer sizes

significant changes on the video quality. As described above the buffer size is mainly a description of how the decoder should work if the file is streamed over a network to reduce the distortions that can occur during the transfer of a bitstream. For video clips that are played only on the local file system the buffer size does not matter. The impact on the quality can be figured out in experiments with network simulation methods that assume diverse bandwidth environments with packet loss and other limitations to get an overview of how the different buffer sizes react and work under these circumstances and will influence the video quality.

Figure 5.23 shows in detail the results of the SSIM calculation on different buffer sizes of the video clip 19 – *video_a* – 09 – *sami* – *ep03* – *var01* at a resolution of 480p. The lines show that variations in the buffer sizes do not have any effect, which is discussed at the graphics above. To show the impact of the bit rate on the resulting quality I added a line for each of the tested bit rates on the graphic which simply shows the same behavior.

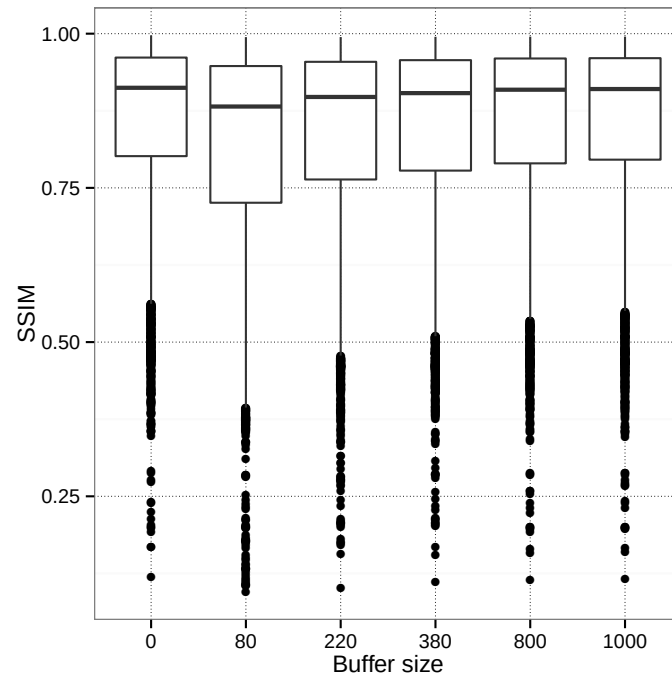


Figure 5.21.: SSIM for different buffer sizes

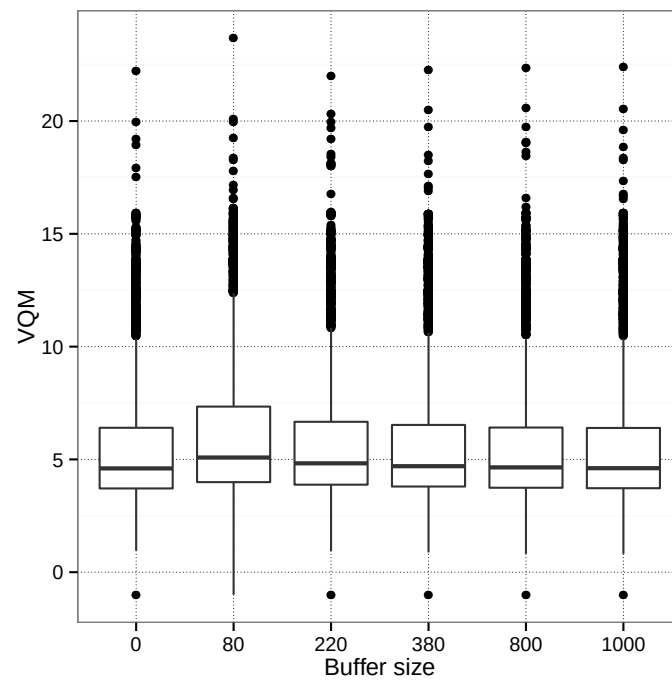


Figure 5.22.: VQM for different buffer sizes

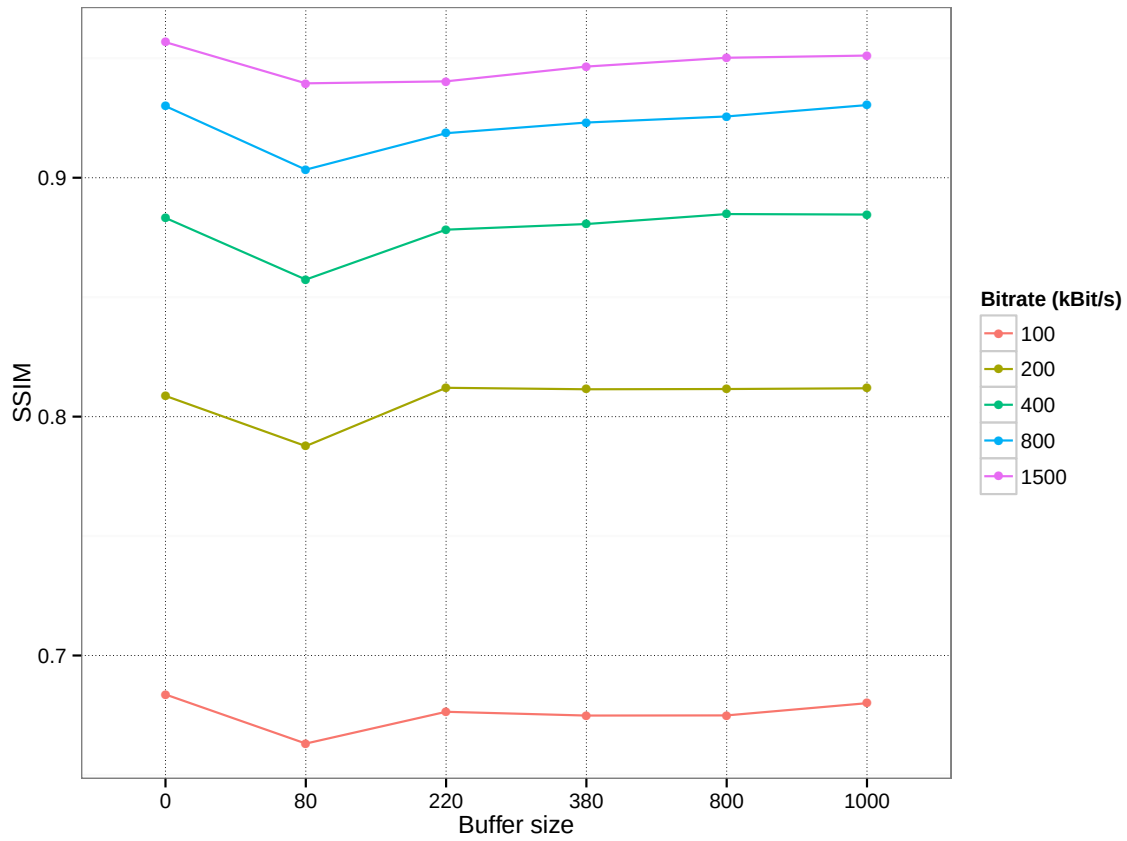


Figure 5.23.: SSIM for different buffer sizes grouped by the bit rate

5.3.4. Impact of Resolution

The figures presented in this section show the overall quality of the clips for each resolution tested. Figure 5.24 is very interesting as it shows that the overall quality of a small

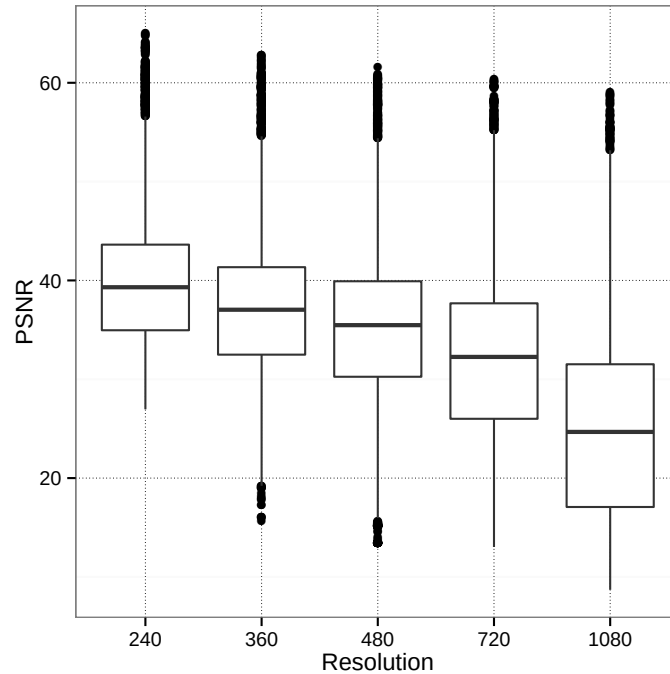


Figure 5.24.: PSNR for resolution

resolution video seems to be better than that of the corresponding HD clips. The interquartile ranges are all equally the same and the quality lies between good and fair except for the HD files, where the quality is significantly lower than in all other resolutions but with very high whiskers that reach up to excellent quality. The number of outliers is high, indicating that many video clips in each resolution seem to have brilliant quality according to the PSNR calculation algorithm. It seems that comparing the resolution is not possible, seeing as the results vary and do not show a clear answer.

While the PSNR calculation shown in figure 5.24 resulted in fair to good quality, the SSIM shown in figure 5.25 attests the SD content an excellent quality. Files with the resolution of 1080p also differ in terms of quality. The results for SSIM only show a poor quality for the HD clips, while the number of outliers as well as the whiskers do not indicate towards better quality values but in the direction of worse quality. The results of the VQM calculation shown in figure 5.26 do not differ from the results already shown. In general, results show that an extremely low bit rate is not suitable for high definition content. As shown in figure 5.27 a sharp decrease in video quality can be noted for files with a resolution of 1080×720 pixels as well as 1920×1080 pixels. To reach fair video

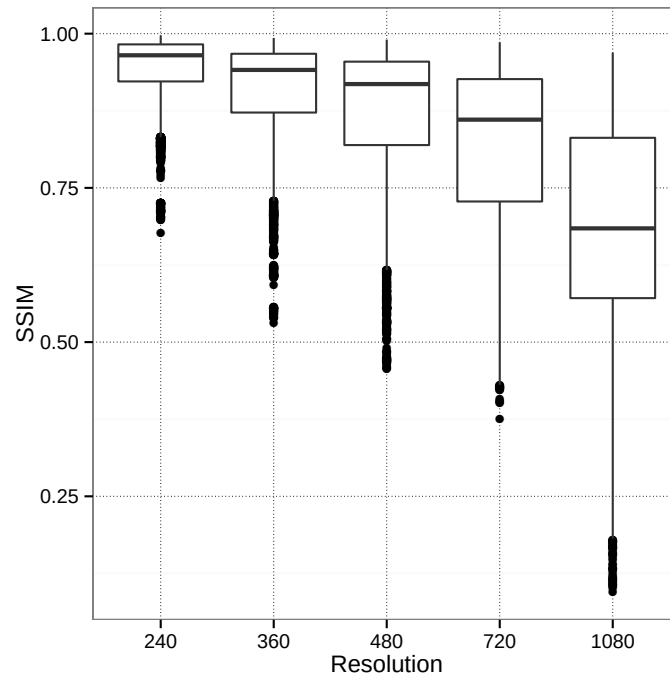


Figure 5.25.: SSIM for resolution

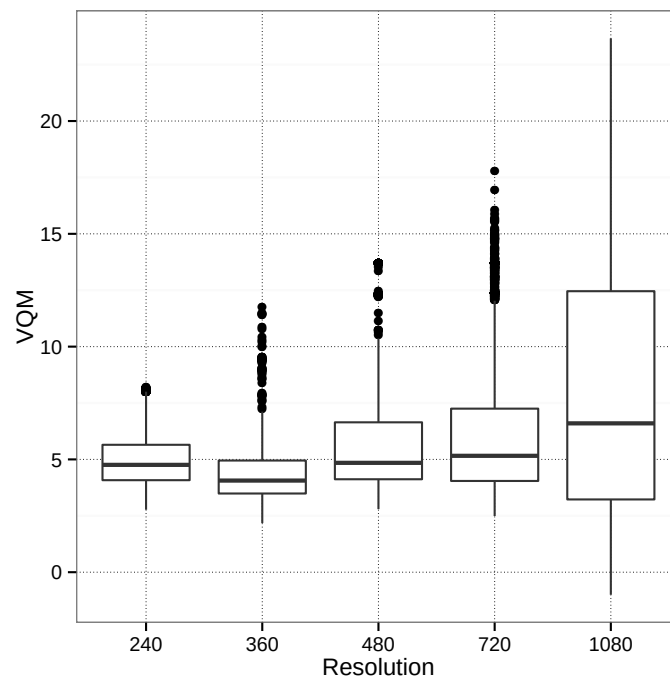


Figure 5.26.: VQM for resolution

quality in high definition resolutions a minimum bit rate of 800 kBit/s must be used. On reviewing the encoder settings the setup of the experiment is perhaps the reason for the bad quality files with higher resolutions. The use of an extended encoding profile instead of a low complex profile might help to improve the quality.

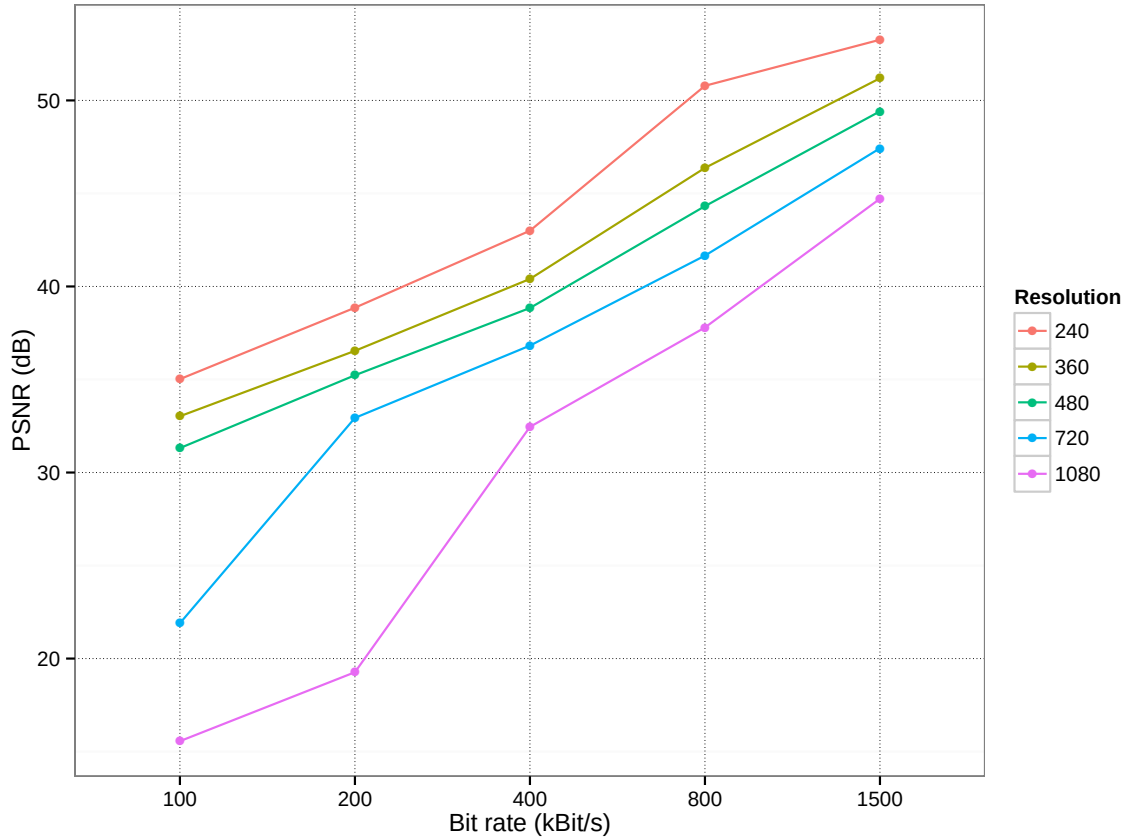


Figure 5.27.: PSNR grouped by resolution

5.3.5. Impact of Content Type

As described in chapter 4 the video clips can be classified into two types. The **mobile** ones were generated with respect to the limitations that can occur when viewing the video clips on mobile devices, while the **classic** clips do not follow any guidelines regarding content. The classic clips contain other cuts and different backgrounds, especially if text like credits or other information is shown, and a different camera motion. Figure 5.28 shows the PSNR values for each video clip and should be helpful for answering the question whether the content type has an impact on the perceived video quality. It is noteworthy that the quality of clip pair 11, which does only contain an intro scene with nearly no motion of objects, as examined earlier, is outstanding. The pair 12 (also

examined earlier) cannot be rated properly as there is so much variation in the values which might be owing to the content, which is of high motion while the results for other pairs are much more consistent. In reviewing figure 5.28 there is no evidence that the content type might impact the perceived video quality. The results for each type, mobile and classic, are always close together.

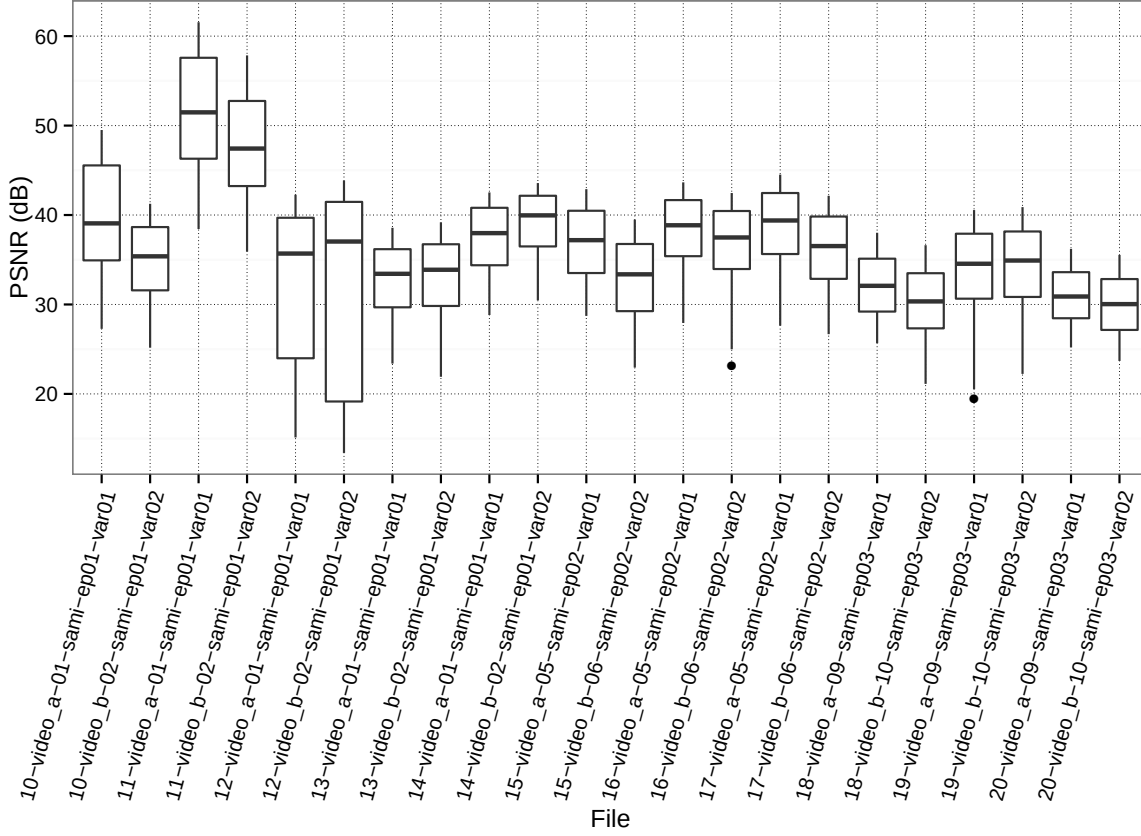


Figure 5.28.: PSNR for each file

5.3.6. ANOVA Validation

In order to study the influence of the different parameters on the perceived quality I have performed an ANOVA (Analysis of Variance) on the PSNR, SSIM and VQM data set. Table 5.2, 5.3 and 5.4 show the results where the first column indicates the degrees of freedom, which is defined as the number of used values for the corresponding setting minus 1. The second column is the Sum of Squares between the different parameter settings, the third column is the number of mean squares indicating the ratio of Sum of Squares to degrees of freedom. The F statistic is presented in the fourth column while the p-value is shown in the last column. All p-values are very close to zero, which

implies that the perceived video quality is affected by the different encoder settings as the variation in the calculated metrics in PSNR, SSIM and VQM is very high. Based on the F statistic values it can be said that in general the bit rate impacts the results of objective metric calculations the most, followed by resolution, content type and buffer size while the GOP length has the least influence.

	Degrees of Freedom	Sum of squares	Mean Squares	F statistic	p-value
bit rate	4	1356253	339063	10378.47	$< 2e - 16$ ***
buffer size	5	20500	4100	125.50	$< 2e - 16$ ***
GOP length	6	4433	739	22.62	$< 2e - 16$ ***
resolution	4	1195633	298908	9149.36	$< 2e - 16$ ***
content type	1	30351	30351	948.06	$< 2e - 16$ ***
Residuals	46177	1478279	32		

Table 5.2.: ANOVA on PSNR with bit rate, buffer size, GOP length, resolution and content type

	Degrees of Freedom	Sum of squares	Mean Squares	F statistic	p-value
bit rate	4	296.1	74.03	12021.2	$< 2e - 16$ ***
buffer size	5	8.4	1.67	271.2	$< 2e - 16$ ***
GOP length	6	0.9	0.16	25.5	$< 2e - 16$ ***
resolution	4	373.7	93.42	15169.0	$< 2e - 16$ ***
content type	1	4.8	4.82	795.58	$< 2e - 16$ ***
Residuals	46177	279.6	0.01		

Table 5.3.: ANOVA on SSIM with bit rate, buffer size, GOP length, resolution and content type

	Degrees of Freedom	Sum of squares	Mean Squares	F statistic	p-value
bit rate	4	111567	27892	5140.15	$< 2e - 16$ ***
buffer size	5	2588	518	95.40	$< 2e - 16$ ***
GOP length	6	940	157	28.87	$< 2e - 16$ ***
resolution	4	64245	16061	2959.91	$< 2e - 16$ ***
content type	1	1102	1102	204.06	$< 2e - 16$ ***
Residuals	46177	249472	5		

Table 5.4.: ANOVA on VQM with bit rate, buffer size, GOP length, resolution and content type

5.4. Summary

The findings of this chapter are summarized below.

High resolution files need more than twice the time for encoding than SD files. While the average encoding duration of a 720p file was about 12 seconds, the duration of the 1080p clips lies at 31 seconds. In general, the performance depends on different factors - first of all the content, the reference duration, as well as the file size of the input footage.

The average file size of all encoded clips is about 0.88 MB. The only significant decrease in the file size can be noted when reducing the bit rate. A reduction of the bit rate by about 50% leads to a decrease of the file size of about the half. Variations in the resolution and GOP lengths do not lead changes regarding file size. As already mentioned above the resulting file size is also dependent on the content and the file size before encoding. Some videos cannot be encoded with the same efficiency as others.

In general it can be said that a reduction of the bit rate reduces the overall perceived video quality as well as the file size. According to the analysis of the objective video quality measures the bit rate seems to be the only parameter in this experiment that does have a significant impact on the video quality. Neither variations of the GOP length nor changing the buffer sizes affects the quality. Lower bit rates result in higher variations, as the interquartile ranges are bigger and the quality varies depending on the content.

The results indicating that lower resolution files are of better quality than high resolution files indicating be handled with care. As already mentioned above the design of the experiment might influence the results in this case, seeing as extremely low bit rates will not capture enough details for a presentation in such resolutions.

The content type, which is classified in mobile and classic in this experiment is not crucial for the video quality, all of the tested pairs resulted in almost the same quality.

Regarding the performance of the metrics it can be said that PSNR evaluation seems to be very variable, especially at lower bit rates as there are many outliers visible on the plots, which might be due to a limitation of the metric but can also be attributed the video content. PSNR predicts a slightly poorer quality for the same videos than for example SSIM does. It bears mentioning, that the metrics differ regarding which clip has the worst quality. Comparing the metrics with each other is not possible, but in general they show the same results with marginal differences, they all rate the quality as quite similar.

Chapter 6.

Conclusion

The aim of this thesis was to measure the influence of different encoder settings on the perceived video quality with different available objective methods and to generate a model of recommendable encoder settings.

I first gave an introduction on the progress in the field of video streaming and the increasing amount of traffic generated via mobile phones and video consumption in the past years in chapter 1.

An overview of relevant literature and past studies was given in chapter 2. Focus was laid on studies presenting objective full reference methods as well as different performance studies of existing metrics, especially PSNR, SSIM and VQM. I have mentioned the most important databases for measuring the performance of video quality metrics, which all provide a large set of video clips with subjective evaluation results.

In chapter 3 the technical background of this thesis was described. An overview of video streaming with the technical background of the streaming paradigm, requirements of a streaming service and available transport protocols were discussed. As every network provider has also challenging tasks to fulfill regarding video streaming and transmission of multimedia content to meet the requirements of the customers, there is one term which is relevant for monitoring the performance of a networks service - Quality of Service. QoS has to be distinguished from Quality of Experience, which was also part of this chapter. The methods for measurement of the perceived video quality by means of subjective and objective quality assessment methods were discussed with the focus on objective full reference methods, as well as problems and limitations of the introduced algorithms and methods. I have described the main purposes for video compression and how compression works in detail. I also gave an overview of common compression standards and their progress. I focused on the H.264/AVC standard and highlighted the differences and improvements of H.264/AVC on previous standards. Adapting the content to the need of the customer is mandatory in today's environments, so I introduced the field of video transcoding, which must be distinguished from video encoding and scalable video coding. As compression always results in a loss of information an overview of the most common

video artifacts was given. For a better understanding of how compression works a short overview of the human visual system was included.

The experimental setup of this study was described in chapter 4. I gave information on which hardware and software tools I used during the study as well as a short description of the chosen encoder settings and the reference files. An insight into the whole encoding process with sample statements for encoding, as well as the comparison process of reference files and encoded videos were given. To summarize the experiment I have encoded each of the 22 reference videos with a combination of the settings listed in table 6.1. In total this resulted in 1050 possible HRC combinations which lead to 23100 encoded videos. For analyzing the quality of the produced video clips I used the objective full reference metrics PSNR, SSIM and VQM, which were either gained through statistic outputs of the encoder console or by calculations through the MSU VQM Tool.

Setting	Used Value
Resolution (pixel)	426×240 , 640×360 , 854×480 , 1280×720 , 1920×1080
Bit rate (kBit/s)	100, 100, 400, 800, 1500
Group of Pixel Length	25, 75, 125, 175, 225, 275, 325
Buffer Size	0, 80, 220, 380, 800, 1000

Table 6.1.: HRC of the experiment

The findings of the experiment, which are described in detail in chapter 5, are summarized below:

Encoding high resolution video files needs twice the time of a normal SD file will need for encoding while, there is no significant difference in the duration between first and second pass encoding. The duration depends on content as well as file size and duration of the reference files.

Reducing the resolution does not lead to smaller file sizes, to decrease the file size by about 50% a reduction of the bit rate has to be undertaken.

The overall perceived video quality is influenced by the bit rate. A reduction of the bit rate leads to significantly lower video quality, while the variation of the GOP length, buffer size and resolution does not have an impact on the video quality. Bit rates of about 400 kBit/s generate video clips with good quality and can be used in any environment for SD content. For encoding HD clips a bit rate of 800 kBit/s or higher is recommended.

As the type of the clips was classified to reach two different areas of application - videos seen on a mobile device and video files that will be watched on a normal screen or TV - there was no difference in the performance between these two types of clips.

Quality evaluation of the three metrics PSNR, SSIM and VQM show nearly the same results.

6.1. Future Work

As the design of the experiment resulted in over 20000 video files it was not possible to verify the results gathered through objective methods with subjective tests. Several studies mentioned in chapter 2 propose to always use subjective as well as objective methods for measuring the perceptual video quality because objective methods do not always describe in a satisfying way what users feel about the video quality. Reviewing all files would be impossible for humans, so a subset of the data which perhaps show significant quality loss or outstanding good quality might be part of this experiment. Subjective experiments are very time consuming and expensive and might not be possible in every environment. Therefore, for better comparability a test of the specified HRC combinations on the freely available video quality databases mentioned in chapter 2, which all provide results of a subjective evaluation might be helpful, but would need to be done in a completely different study.

During the last year, further progress in video encoding standards was achieved. The High Efficiency Video Coding Standard also called H.265 was developed - see [59] for more details. An interesting research work would be comparing the outcomes as well as the performance of the new codec with the data gathered in this experiment to see if and how the new codec will improve the quality of experience.

Creating a test model with much more objective quality metrics, which all are more complex and will therefore require more computing capacity, is also an interesting task for the future. Either to corroborate the results of the study shown in this thesis or to evaluate the new metrics.

Appendix A.

Curriculum Vitae

Personal Information

Name	Doris Isabella Braunoeder, BsC
Nationality	Austria
Email	doris.braunoeder@gmail.com

Education

1993 - 1997	Volksschule Hannersdorf (Elementary School)
1997 - 2001	Bundesrealgymnasium, Oberschuetzen (Secondary School)
2001 - 2006	Handelsakademie fuer Informationstechnologie, Oberwart
2006 - 2009	Bachelor Studies in Informatics at University of Vienna
since 2009	Master studies for Media Informatics at University of Vienna

Professional experience

2003 - 2007	Summer Job at SKA RZ Bad Tatzmannsdorf
2007 - 2010	different marginal employments besides studies
2010 - 2013	Software Developer REBEAT Digital www.rebeat.com
since 2013	Software Developer The Documentary Network www.documentary.net

Bibliography

- [1] Cisco White Paper. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013-2018. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf, February 2014. [Online; accessed 31-July-2014].
- [2] Youtube Press. Youtube Statistics. <https://www.youtube.com/yt/press/en/statistics.html>, 2014. [Online; accessed 31-July-2014].
- [3] Sandvine. GLOBAL INTERNET PHENOMENA REPORT 1H 2014. <https://www.sandvine.com/trends/global-internet-phenomena>, 2014. [Online; accessed 10-September-2014].
- [4] International Telecommunication Union. ITU-R Recommendation J.144: Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference, 2004.
- [5] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.
- [6] Zhou Wang and A.C. Bovik. Mean Squared Error: Love it or leave it? A new look at Signal Fidelity Measures. *IEEE Signal Processing Magazine*, 26(1):98–117, January 2009.
- [7] M.H. Pinson and S. Wolf. A new standardized method for objectively measuring video quality. *IEEE Transactions on Broadcasting*, 50(3):312–322, September 2004.
- [8] Video Quality Experts Group (VQEG). Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment, Phase II. <http://www.its.bldrdoc.gov/vqeg/projects/frtv-phase-ii/frtv-phase-ii.aspx>, 2003. [Online; accessed 10-September-2014].
- [9] K. Seshadrinathan, R. Soundararajan, A.C. Bovik, and L.K. Cormack. Study of Subjective and Objective Quality Assessment of Video. *IEEE Transactions on Image Processing*, 19(6):1427–1441, June 2010.

- [10] Quan Huynh-Thu and Mohammed Ghanbari. The accuracy of PSNR in predicting video quality for different video scenes and frame rates. *Telecommunication Systems*, 49(1):35–48, 2012.
- [11] F.N. Rahayu, U. Reiter, M.T.M. Nielsen, T. Ebrahimi, P. Svensson, and A. Perkis. Analysis of SSIM performance for Digital Cinema applications. In *International Workshop on Quality of Multimedia Experience. QoMEX 2009*, pages 23–28, July 2009.
- [12] T. Zinner, O. Hohlfeld, O. Abboud, and T. Hossfeld. Impact of frame rate and resolution on objective QoE metrics. In *Second International Workshop on Quality of Multimedia Experience (QoMEX), 2010*, pages 29–34, June 2010.
- [13] Thomas Zinner, Osama Abboud, Oliver Hohlfeld, Tobias Hossfeld, and Phuoc Tran-Gia. Towards QoE Management for Scalable Video Streaming. In *21th ITC Specialist Seminar on Multimedia Applications - Traffic, Performance and QoE*, Miyazaki, Japan, March 2010.
- [14] J. Brandt and L. Wolf. Impact of adaptation dimensions on video quality. In *IEEE 14th International Symposium on Consumer Electronics (ISCE), 2010*, pages 1–6, June 2010.
- [15] Mario Vranje, Snjeana Rimac-Drlje, and Kreimir Grgi. Review of objective video quality metrics and performance comparison using different databases. *Signal Processing: Image Communication*, 28(1):1 – 19, 2013.
- [16] M.K. Thakur, V. Saxena, and J. P. Gupta. A performance analysis of objective video quality metrics for digital video watermarking. In *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010*, volume 4, pages 12–17, July 2010.
- [17] S. Chikkerur, V. Sundaram, M. Reisslein, and L.J. Karam. Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison. *IEEE Transactions on Broadcasting*, 57(2):165–182, June 2011.
- [18] P.I. Hosur and P. Costa. Objective video quality assessment for multi bit rate adaptive streaming. In *IEEE 16th International Symposium on Consumer Electronics (ISCE), 2012*, pages 1–6, June 2012.
- [19] G. Van Wallendael, S. Van Leuven, J. De Cock, P. Lambert, R. Van de Walle, N. Staelens, and P. Demeester. Evaluation of full-reference objective video quality metrics on high efficiency video coding. In *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pages 1294–1299, May 2013.
- [20] Tsung-Jung Liu, Yu-Chieh Lin, Weisi Lin, and C-C Jay Kuo. Visual quality assessment: recent developments, coding applications and future trends. *APSIPA Transactions on Signal and Information Processing*, 2:e4, 2013.

- [21] Alberto Testolin, Marco Zanforlin, Michele De Filippo De Grazia, Daniele Munaretto, Andrea Zanella, Marco Zorzi, and Michele Zorzi. A machine learning approach to QoE-based video admission control and resource allocation in wireless systems. In *13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2014, pages 31–38, June 2014.
- [22] M. Vranjes, S. Rimac-Drlje, and D. Zagar. Subjective and objective quality evaluation of the H.264/AVC coded video. In *15th International Conference on Systems, Signals and Image Processing. IWSSIP 2008*, pages 287–290, June 2008.
- [23] S. Wulf and U. Zolzer. Full-reference video quality assessment on high-definition video content. In *6th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2012, pages 1–10, December 2012.
- [24] S. Paulikas. Estimation of video quality of H.264/AVC video streaming. In *IEEE EUROCON*, 2013, pages 694–700, July 2013.
- [25] C. Keimel, T. Oelbaum, and K. Diepold. Improving the verification process of video quality metrics. In *International Workshop on Quality of Multimedia Experience. QoMEx 2009*, pages 121–126, July 2009.
- [26] International Telecommunication Union. ITU-R Recommendation BT.500: Methodology for the subjective assessment of the quality of television pictures, 2012.
- [27] C. Greco, I.D. Nemoianu, M. Cagnazzo, J. Le Feuvre, F. Dufaux, and B. Pesquet-Popescu. Chapter 11 - Multimedia Streaming . In Sergios Theodoridis and Rama Chellappa, editors, *Academic Press Library in Signal Processing Image and Video Compression and Multimedia*, volume 5 of *Academic Press Library in Signal Processing*, pages 327 – 381. Elsevier, 2014.
- [28] Dapeng Wu, Y.T. Hou, Wenwu Zhu, Ya-Qin Zhang, and J.M. Peha. Streaming video over the internet: approaches and directions. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3):282–300, March 2001.
- [29] John G. Apostolopoulos, Wai tian Tan, and Susie J. Wee. Video Streaming: Concepts, Algorithms, and Systems. Technical report, HP Laboratories, 2002.
- [30] International Telecommunication Union. ITU-R Recommendation E.800 : Definitions of terms related to quality of service, 2008.
- [31] S. Moller, K.-P. Engelbrecht, C. Kuhnel, I Wechsung, and B. Weiss. A taxonomy of quality of service and Quality of Experience of multimodal human-machine interaction. In *International Workshop on Quality of Multimedia Experience. QoMEx 2009*, pages 7–12, July 2009.

- [32] International Telecommunication Union. ITU-R Recommendation ITU-R P.10/G.100: Amendment 1: New Appendix I Definition of Quality of Experience (QoE), 2006.
- [33] S. Winkler and P. Mohandas. The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics. *IEEE Transactions on Broadcasting*, 54(3):660–668, September 2008.
- [34] Zhou Wang, Hamid R. Sheikh, and Alan C. Bovik. Objective video quality assessment. In *IN THE HANDBOOK OF VIDEO DATABASES: DESIGN AND APPLICATIONS*, pages 1041–1078. CRC Press, 2003.
- [35] Pinar Sarisaray-Boluk. Performance comparisons of the image quality evaluation techniques in Wireless Multimedia Sensor Networks. *Wireless Networks*, 19(4):443–460, 2013.
- [36] Z. Wang, E.P. Simoncelli, and AC. Bovik. Multiscale structural similarity for image quality assessment. In *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, 2004*, volume 2, pages 1398–1402, November 2003.
- [37] Alexander Tanchenko. Visual-PSNR measure of image quality . *Journal of Visual Communication and Image Representation*, 25(5):874 – 878, 2014.
- [38] H.R. Sheikh and AC. Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, February 2006.
- [39] Kalpana Seshadrinathan and Alan C. Bovik. Motion-based perceptual quality assessment of video. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *Human Vision and Electronic Imaging*, volume 7240 of *SPIE Proceedings*, page 72400. SPIE, 2009.
- [40] D.M. Chandler and S.S. Hemami. VSNR: A Wavelet-Based Visual Signal-to-Noise Ratio for Natural Images. *IEEE Transactions on Image Processing*, 16(9):2284–2298, September 2007.
- [41] Zhou Wang and Eero P. Simoncelli. Reduced-reference image quality assessment using a wavelet-domain natural image statistic model. In *Proceedings of SPIE Human Vision and Electronic Imaging*, pages 149–159, 2005.
- [42] Qiang Li and Zhou Wang. Reduced-Reference Image Quality Assessment Using Divisive Normalization-Based Image Representation. *IEEE Journal of Selected Topics in Signal Processing*, 3(2):202–211, April 2009.
- [43] Y. Fu-zheng, W. Xin-dai, C. Yi-lin, and W. Shuai. A no-reference video quality assessment method based on digital watermark. In *14th IEEE Proceedings on*

Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003., volume 3, pages 2707–2710 vol.3, Sept 2003.

- [44] Fuzheng Yang, Shuai Wan, Yilin Chang, and Hong Ren Wu. A novel objective no-reference metric for digital video quality assessment. *IEEE Signal Processing Letters*, 12(10):685–688, October 2005.
- [45] Zhou Wang, AC. Bovik, and B.L. Evan. Blind measurement of blocking artifacts in images. In *International Conference on Image Processing, 2000*, volume 3, pages 981–984, 2000.
- [46] Iain E. G. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley & Sons, Inc., New York, USA, 2003.
- [47] David R. Bull. Chapter 1 - An Introduction to Video Coding . In Sergios Theodoridis and Rama Chellappa, editors, *Academic Press Library in signal Processing Image and Video Compression and Multimedia*, volume 5 of *Academic Press Library in Signal Processing*, pages 3 – 26. Elsevier, 2014.
- [48] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. Video coding with H.264/AVC: tools, performance, and complexity. *IEEE Circuits and Systems Magazine*, 4(1):7–28, January 2004.
- [49] Stefan Winkler. *Digital Video Quality – Vision Models and Metrics*. John Wiley & Sons, January 2005.
- [50] David R. Bull. Chapter 12 - Video Coding Standards. In David R. Bull, editor, *Communicating Pictures*, pages 411 – 449. Academic Press, Oxford, 2014.
- [51] Jun Xin, Chia-Wen Lin, and Ming-Ting Sun. Digital Video Transcoding. *Proceedings of the IEEE*, 93(1):84–97, January 2005.
- [52] I Ahmad, Xiaohui Wei, Yu Sun, and Ya-Qin Zhang. Video transcoding: an overview of various techniques and research issues. *IEEE Transactions on Multimedia*, 7(5):793–804, October 2005.
- [53] L. Goldmann, F. De Simone, F. Dufaux, T. Ebrahimi, R. Tanner, and M. Lattuada. Impact of video transcoding artifacts on the subjective quality. In *Second International Workshop on Quality of Multimedia Experience (QoMEX), 2010*, pages 52–57, June 2010.
- [54] H. Schwarz, D. Marpe, and T. Wiegand. Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1103–1120, September 2007.

- [55] Nidhi and N. Aggarwal. A review on Video Quality Assessment. In *Recent Advances in Engineering and Computational Sciences (RAECS)*, pages 1–6, March 2014.
- [56] MeWiki. X264 Settings. http://mewiki.project357.com/wiki/X264_Settings, 2014. [Online; accessed 05-August-2014].
- [57] Werner Robitza, Yohann Pitrey Helpiquet, Matej Nezveda, Shelley Buchinger, and Helmut Hlavacs. Made for Mobile: a video database designed for Mobile Television. In *Sixth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM 2012)*, Scottsdale, Arizona, January 2012.
- [58] Shelley Buchinger, J. Wippersberg, K. Lojka, K. Macher, Werner Robitza, Matej Nezveda, Patrik Hummelbrunner, and Helmut Hlavacs. Mobile TV Content Design Rules. In Y. Kompatsiaris, B. Merialdo, and S. Lian, editors, *TV Content Analysis: Techniques and Applications*. CRC Press, Taylor Francis LLC, March 2012.
- [59] Yanxiang Wang, Charith Abhayaratne, and Marta Mrak. Chapter 3 - High Efficiency Video Coding (HEVC) for Next Generation Video Applications. In Sergios Theodoridis and Rama Chellappa, editors, *Academic Press Library in signal Processing Image and Video Compression and Multimedia*, volume 5 of *Academic Press Library in Signal Processing*, pages 93 – 117. Elsevier, 2014.