

# Computational inference and control of quality in multimedia services

***Citation for published version (APA):***

Menkovski, V. (2013). *Computational inference and control of quality in multimedia services*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven.  
<https://doi.org/10.6100/IR751521>

***DOI:***

[10.6100/IR751521](https://doi.org/10.6100/IR751521)

***Document status and date:***

Published: 01/01/2013

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# Computational inference and control of quality in multimedia services

Vlado Menkovski





# Computational inference and control of quality in multimedia services

## PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de  
Technische Universiteit Eindhoven, op gezag van de  
rector magnificus, prof.dr.ir. C.J. van Duijn, voor een  
commissie aangewezen door het College voor  
Promoties in het openbaar te verdedigen  
op dinsdag 5 maart 2013 om 16.00 uur

door

Vlado Menkovski

geboren te Skopje, Macedonië

Dit proefschrift is goedgekeurd door de promotor:

prof.dr. A. Liotta

This Ph.D. thesis has been approved by a committee with the following members:

prof.dr.ir. F.de Turck, Universiteit Gent, Belgium

prof.dr. A. Kondozi, University of Surrey, UK

prof.dr.ir. P.H.N.de With, Eindhoven University of Technology, The Netherlands

prof.dr.ir. E. Fledderus, Eindhoven University of Technology, The Netherlands

A catalogue record is available from the Eindhoven University of Technology Library

Computational inference and control of quality in multimedia services

Author: Vlado Menkovski

Eindhoven University of Technology, 2013.

ISBN: 978-90-386-3355-8

NUR 980

Copyright © 2013 by Vlado Menkovski

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the prior written consent of the author.

Typeset using L<sup>A</sup>T<sub>E</sub>X, printed in The Netherlands



*To my parents,  
Angel who dedicated his life to his sons and  
Snežana whose strength inspires me deeply.*





# Summary

Quality is the degree of excellence we expect of a service or a product. It is also one of the key factors that determine its value. For multimedia services, understanding the experienced quality means understanding how the delivered fidelity, precision and reliability correspond to the users' expectations. Yet the quality of multimedia services is inextricably linked to the underlying technology. It is developments in video recording, compression and transport as well as display technologies that enables high quality multimedia services to become ubiquitous. The constant evolution of these technologies delivers a steady increase in performance, but also a growing level of complexity. As new technologies stack on top of each other the interactions between them and their components become more intricate and obscure. In this environment optimizing the delivered quality of multimedia services becomes increasingly challenging. The factors that affect the experienced quality, or Quality of Experience (QoE), tend to have complex non-linear relationships. The subjectively perceived QoE is hard to measure directly and continuously evolves with the user's expectations. Faced with the difficulty of designing an expert system for QoE management that relies on painstaking measurements and intricate heuristics, we turn to an approach based on learning or inference. The set of solutions presented in this work rely on computational intelligence techniques that do inference over the large set of signals coming from the system to deliver QoE models based on user feedback. We furthermore present solutions for inference of optimized control in systems with no guarantees for resource availability. This approach offers the opportunity to be more accurate in assessing the perceived quality, to incorporate more factors and to adapt as technology and user expectations evolve. In a similar fashion, the inferred control strategies can uncover more intricate patterns coming from the sensors and therefore implement farther-reaching decisions. Similarly to natural systems, this continuous adaptation and learning makes these systems more robust to perturbations in the environment, longer lasting accuracy and higher efficiency in dealing with increased complexity. Overcoming this increasing complexity and diversity is crucial for addressing the challenges of future multimedia system. Through experiments and simulations this work

demonstrates that adopting an approach of learning can improve the subjective and objective QoE estimation, enable the implementation of efficient and scalable QoE management as well as efficient control mechanisms.

# Contents

<b>Summary</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 QoE Definition . . . . .	3
1.2 Factors that affect QoE . . . . .	5
1.3 Resources vs. Quality . . . . .	8
1.4 Handling the complexity of QoE modelling . . . . .	11
1.5 Learning vs. Deterministic Design . . . . .	12
1.6 Main contributions . . . . .	13
<b>2 Objective QoE Models</b>	<b>15</b>
2.1 State of the art . . . . .	16
2.1.1 PSNR . . . . .	16
2.1.2 SSIM . . . . .	17
2.1.3 VQM . . . . .	19
2.1.4 MOVIE . . . . .	21
2.1.5 Reduced and no reference methods . . . . .	22
2.2 Models for JPEG2000 and MPEG4/AVC . . . . .	23
2.2.1 Characterizing the video content . . . . .	27
2.3 Experimental analysis of objective QoE . . . . .	29
2.3.1 PSNR Results . . . . .	31
2.3.2 SSIM Results . . . . .	31
2.3.3 VQM Results . . . . .	31
2.3.4 MOVIE Results . . . . .	33
2.3.5 SI, TI, C and M Results . . . . .	33
2.4 Discussion and conclusions . . . . .	33
<b>3 Subjective QoE Models</b>	<b>37</b>
3.1 State of the art . . . . .	38
3.1.1 Rating the quality . . . . .	38
3.1.2 Limits and noticeable differences . . . . .	41
3.2 Maximum likelihood difference scaling . . . . .	43
3.2.1 The video subjective study . . . . .	47

3.2.2	MLDS subjective results . . . . .	48
3.3	Adaptive MLDS . . . . .	52
3.3.1	Adaptive MLDS method . . . . .	52
3.3.2	Learning convergence . . . . .	55
3.3.3	Experimental setup and results . . . . .	56
3.4	Conclusions . . . . .	62
<b>4</b>	<b>QoE Management Framework</b>	<b>63</b>
4.1	Existing approaches . . . . .	63
4.2	QoE management for video streaming . . . . .	65
4.2.1	Architecture of a video streaming systems . . . . .	65
4.2.2	A hybrid QoE management framework . . . . .	68
4.3	QoE management for a Mobile IPTV service . . . . .	70
4.3.1	Objective Measurements . . . . .	70
4.3.2	Subjective measurements . . . . .	71
4.4	Computational inference of QoE models . . . . .	71
4.4.1	Supervised learning background . . . . .	72
	Decision Trees . . . . .	73
	Support Vector Machine . . . . .	75
	Ensemble methods . . . . .	75
4.4.2	QoE models for mobile IPTV . . . . .	76
	Subjective QoE models developed in the lab . . . . .	76
	Subjective QoE models for mobile IPTV . . . . .	82
4.5	On-line inference of QoE models . . . . .	85
4.5.1	Online supervised learning background . . . . .	85
	Hoeffding trees . . . . .	86
	Online ensemble methods . . . . .	88
4.5.2	Subjective QoE models with continuous learning . . . . .	89
4.6	QoE Remedies . . . . .	94
4.6.1	Estimating the QoE remedies . . . . .	94
4.6.2	QoE remedies for mobile IPTV . . . . .	99
4.7	Conclusions . . . . .	99
<b>5</b>	<b>QoE Active Control</b>	<b>103</b>
5.1	QoE active control framework . . . . .	103
5.2	HTTP adaptive streaming client . . . . .	105
5.2.1	DASH standard . . . . .	106
5.2.2	Existing heuristic strategies . . . . .	107
5.3	The intelligent streaming agent . . . . .	111

<b>CONTENTS</b>	<b>xi</b>
5.3.1 Reinforcement learning background . . . . .	112
5.3.2 Intelligent streaming agent architecture . . . . .	118
5.3.3 Reward/penalty function . . . . .	120
5.3.4 Estimating network throughput trends . . . . .	123
5.3.5 Simulating the background traffic . . . . .	126
5.3.6 Agent performance . . . . .	127
5.4 Conclusions . . . . .	128
<b>6 Conclusions</b>	<b>131</b>
<b>List of Figures</b>	<b>135</b>
<b>References</b>	<b>153</b>
<b>List of Publications</b>	<b>155</b>
<b>Awards</b>	<b>159</b>
<b>Curriculum Vitæ</b>	<b>161</b>



# 1

## Introduction

We perceive the environment through our senses. Between the different senses, the visual dominates the sensory input accounting for 80% of the received information. Accordingly, the neural system has evolved to efficiently process this incoming data. Approximately 50% of the posterior cerebral cortex is dedicated to visual information processing [1]. We perceive the details in the visual information directly and rapidly [2]. Comparably, people retain 10% of what they hear, 30% of what they read and 80% of what they see and do [3]. Video utilizes these mechanisms to deliver high density of information. It is therefore a very attractive medium for a growing number of digital services. Whether it is a simple animation that illustrates the distance to the next turn on our navigation device or a video enabled group teleconferencing that delivers visual contact with our peers, video delivers more information faster than other media. As video enabled services become more present in our lives, our expectations about their performance and reliability is being set. In order to meet customer's expectations, service providers need to be able to deliver increasingly more demanding services with higher quality standards. This development delivers a high toll on maintenance costs and requires frequent upgrades of available resources. Moreover, the upgrade of some wired and wireless transmission technologies is becoming more challenging as technologies are reaching some physical limits. In this situation the need for smarter management strategies is evident as traditional management approaches such as over-provisioning offer little to improve the utilization of the resources.

Efficient management of networked services requires understanding of the relationship between different available resources, i.e. computational, storage, network throughput and the delivered quality. However, video-enabled services are operating on a vast diversity of terminal devices, encoding and transmission systems. Video is practically being watched on devices of all shapes and sizes, and over many different wired and wireless network systems. Each of them delivers slightly different experience to the



user, as their performance and characteristics varies.

Another dimension of complexity is added by the interactions between different parts of the system. As stacks of technologies work together to deliver the services, the complex interdependencies between them and the effect on the perceived quality is often not fully understood. In addition, service providers rarely have control over all the components of the video delivery system, and have to rely on the quality of supporting services.

On top of all, there is no established method for estimating the quality of video-enabled services. The perceived quality is highly subjective and depends on many external factors, such as the environment, user expectations and the type of content. Existing methods for subjective quality estimation are complex and costly to implement and present high variance in the results.

Motivated by these challenges, this thesis proposes an approach for efficient management of multimedia services. It presents a QoE aware framework for network management that incorporates computational intelligence methods to deal with the evolving complexities in the multimedia systems, and introduces a novel psychometric method that deals with the difficulties of subjective measurements. The framework is designed as a control loop over a general-purpose multimedia system.

As illustrated in Figure 1.1, a negative feedback control loop consists of three units: the controller, the sensor and the system under control. The sensor measures the system output and compares it with the desired one. The difference is fed into the controller that inputs a control strategy to the system in order to minimize the measured difference. Similarly, the multimedia system controller issues different management strategies based on the measured performance of the system (Figure 1.2). The measured value is the subjective QoE perceived by the user and the objectively measured performance by the system components. This value is compared with the desired level of performance and the difference is sent to the controller. The controller can then manage the different system components, allocate necessary resources, execute admission control, or implement other control strategies to achieve the desired level of performance.

This approach offers a viable way to incorporate the large number of factors that affect the quality into the decision process of the controller. It provides for a way to continuously learn and improve the management process based on measurements of the performance and subjective user feedback. In this manner the system maintains a high level of performance

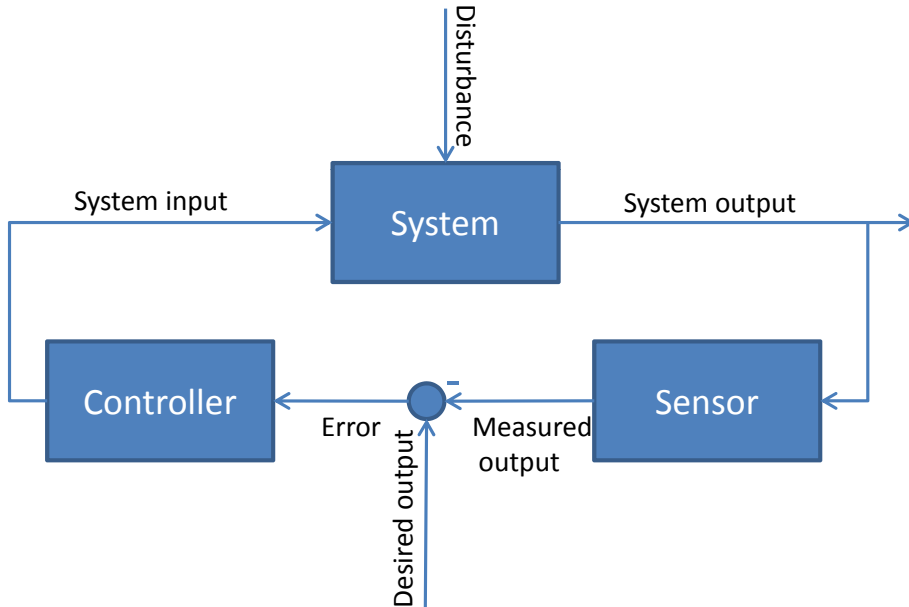


Figure 1.1: A feedback loop in a control system

with minimum cost in the changing environment. This results in a better utilization of the available resources and a user-centric based management.

## 1.1 QoE Definition

In the framework presented here, the system output (or the main performance metric) is the subjectively perceived quality. But how do we understand the term 'quality' in regards to multimedia services?

According to the dictionary, the meaning of quality is: "The degree of excellence" [4] (of a product, service or activity). Quality as a phenomenon has been examined in many disciplines such as philosophy, business and engineering. More specifically quality involves perception, but also expectations. Some definitions refer to it as a subjective phenomenon: "The feeling of high quality occurs when perception exceeds expectation; the feeling of low quality occurs when perception does not meet expectation." [5]. Other definitions focus on more objective, measurable factors: "Degree to which a set of inherent characteristics fulfils requirements." where requirement

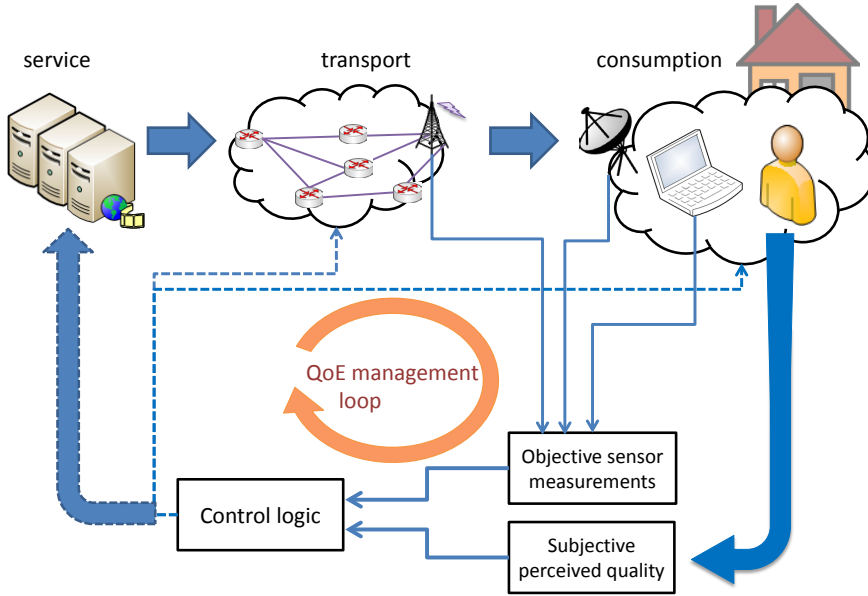


Figure 1.2: Control loop in a multimedia system

is defined as need or expectation [6]. In any case, quality is connected with either objective or subjective expectations. More precisely quality is evaluated in regards to the objectively measured or subjective perceived performance.

For services such as telephony, computer networks, and including voice services built on top of them a commonly used metric for quality is the Quality of Service (QoS). QoS defines a set of requirements that need to be met in order for the service to be considered of high quality. These requirements are objectively measurable values and consider performance factors, such as latency and errors in the network. The possibility to set these well defined QoS requirements is enabled by the good understanding of the compression and transmission factors and the subjective perception of speech. However, with the introduction of new types services and varied content, all delivered on a plethora of different devices, understanding 'subjective' perception becomes challenging.

The need to better communicate the service quality has created a need for a precise quality metric. However, instead of continuously expanding and adapting the QoS requirements, the choice was made to introduce a new metric: the Quality of Experience (QoE) [7]. This metric is better suited for the task, because it is a subjective metric, which captures the effect of all the factors that contribute to the subjective experience.

QoE appears in the literature by different definitions, but generally it is agreed that "QoE measures the quality experienced while using a service" [8]. However, other definitions, such as the one from the ITU-T Focus Group on IPTV (FG IPTV) [9] avoid using the term quality. The FG IPTV defines QoE as the overall acceptability of an application or service, as perceived subjectively by the end-user. The definition of the European Network on QoE in multimedia systems and services is '*QoE is the degree of delight or annoyance of the user of an application or service. It results from the fulfilment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state*' [7]. The relation to the subjective perception of the user and its expectations is clearly evident as the defining characteristic of the metric.

Defining QoE is the initial step. In order to successfully build an efficient management system we need to first understand which factors affect it and how. Next we need to understand the relationship between the available resources in the system and those factors. Finally we need to know how to develop control strategies that utilize the available resources in such a manner as to maximize the delivered QoE.

## 1.2 Factors that affect QoE

QoE is a metric that captures the degree to which our expectations about the service have been met. But, how do we form expectations of a video service and which are the factors that affect it?

We perceive multimedia stimuli first with our senses, and ultimately through the cognitive processes in the brain. Naturally, the characteristics of the human visual and auditory systems are intrinsically linked to the expectations from multimedia services. The auditory system can detect sound in a specific range of frequencies. The typical hearing range for a human is between 20Hz and 20KHz [10]. Therefore, reproduction of sound outside of this range will add no additional quality or value to the

service; it will only spend more resources. The visual system has elevated sensitivity to contrast in the range of 0.5 to 16 cycles per degree of visual angle and drops abruptly on higher frequency [11]. This means that a 1080p HDTV reproduction viewed at a distance between 3 and 4 screen heights can generate patterns on our retina with up to 28 and 37 cycles per degree respectively. This is significantly above the threshold limit for most viewers [12], so increasing the resolution more at this distance will be of little utility.

These examples demonstrate that even though we intuitively consider fidelity of the video and audio as the most significant factor in quality, this has to be taken into the context of the characteristics of the HVS. Improving certain aspects of fidelity could be without any benefit. On the other hand, limited loss of fidelity can be just as imperceptible as a flawless reproduction, while delivering significant benefits in terms of resource utilization.

The Human visual and auditory system is not only characterized by a hearing range and a contrast sensitivity range. It is actually very complex and not fully understood. As more characteristics are being discovered the more we can use this knowledge to optimize multimedia services. Some of these limitations are commonly used to improve coding efficiency. For example high spatial frequencies are perceived achromatic [13]. So the amount of data that conveys the colour of the image can be safely reduced in respect to the amount of data that carries the luminance of the image. Other characteristics, such as the masking effects can be used to cover noise in images. If the noise is superimposed over a region of patterns with high contrast, it is significantly less perceptible than over a uniformly coloured region.

Video compression techniques benefit from the varying sensitivity to different ranges of spatial frequency. The video is first transformed into the frequency domain using a discrete cosine transform (DCT) [14]. Then different parts of the image can be encoded with different precision. The last step is known as quantization, whereby high-frequency coefficients are more coarsely quantized than low-frequency coefficients. This is referred to as a lossy compression method. Even though lossy compression can deliver significant benefits in reducing the size of the video, it also causes loss in quality. Quantization causes artefacts such as blockiness, particularly in heavily compressed video, which degrade the QoE.

Further artefacts appear in a video due to modern compression tech-

niques, such as blur, colour bleeding, ringing, false edges and jagged motion [12]. Some of them are present due to spatial compression techniques, which compress individual images. Others are present due to temporal compression method, which reduces the redundancy over multiple images.

Another factor that introduces artefacts is transmission errors [15, 16]. When a packet of video data is lost, has errors or does not arrive on time the video decoder can freeze the video playback or compensate by using some concealment method. Usually this means interpolating neighbouring pixels in space and in time [17]. However, this often results in very noticeable artefacts.

When transmission protocols are used to guarantee delivery via retransmission, the lack of network resources leads to delays and freezes in playback. This is a very important factor in the overall experience of the service and can have the most significant impact [18]. It is also established that impairments such as freezes and errors have higher impact on the QoE as their amplitude and frequency increase [19].

Adaptive video streaming technologies allow for reducing the fidelity of the signal in order to avoid freezes. This technique attempts to improve the delivered QoE in cases of restricted resources by downloading the video at lower bit-rates. In this case predicting the right bit-rate level is important as the changes of bit-rate levels during playback have shown to be an impairment on its own. The size of the impairment is proportional to the frequency and amplitude of the change [20].

Naturally, the audio quality is a key factor in the QoE of multimedia services. In fact, audio quality is even more important factor than video quality [12]. Audio compression also benefits from the characteristics of the auditory system. Lossy audio compression methods cause audio artefacts in a similar fashion to the video compression methods. However, audio compression and encoding requires significantly less resources than video and, because of its importance, its resources are rarely restricted. This often shifts the management focus on the video aspect.

Nevertheless, other more general aspects are also important factors. One such example is the audio and video synchronization. The investigation of media synchronization in [21] concludes that the effect of unsynchronized audio on the QoE depends on the type of content. For some types of content as head and shoulders news broadcast, it has a massive effect. However, for other content the viewers can demonstrate more tolerance. In another investigation of audio and video correlation and lip synchronization Mued

et al. conclude that the effects on the perceived quality from audio-video misalignment are different when the content is of a passive or an active communication [22] .

Depending on the type of service, there could be other types of impairments such as start-up delays or loss in responsiveness. Overall the factors that affect the QoE are noticeable and are not expected by the viewer. We use our eyes and ears to collect the information from the outside world, but it is the brain that forms our perceptions [23]. The cognition process in the human brain is not understood well, however we know that we do not need all the details to recognize a pattern. Our sensors are designed in this manner, working in restricted ranges. The rest of the details are conceptualized by the cognition process. Nevertheless, with fewer details, the brain needs to work harder to compensate. Sometimes we are willing to do that, because we are watching an old family movie on an outdated technology. But, other times, when we are watching a video on our new and costly mobile device our expectations are high, so the delivered QoE needs to reflect that. As this might be an insurmountable task, measuring QoE in relative terms can be a better solution than attempting to make an inaccurate absolute metric. Pursuing this approach we have developed a method for measuring subjective video quality, which estimates the utility of the system resources in terms of the delivered QoE [24, 25].

### 1.3 Resources vs. Quality

We have seen that a significant amount of factors contribute to the QoE. However, there is also a clear relationship between these factors and the available resources. Most types of artefacts can be efficiently masked if we can encode the video with enough bits, which also need to be transported by the network accurately and on time. Unfortunately, in any real engineered system the available resources are limited. We can only serve a limited number of users, the network can only transport a limited number of bits per second, and finally storage space and computational power are equally limited. So in order to efficiently manage the system we need to understand the relationship between the allocated resources and the resulting QoE.

The audio and video fidelity have a clear relationship with the available resources. The more bits are used in the encoding process the more accurate the decoded audio and video will be. Uncompressed video contains no encoding degradation, but it requires very large storage space and

is not suitable for transmission over a restricted network channel. Compression can be 'lossless' where the signal can be exactly reproduced or 'lossy' that introduces loss in the fidelity of the signal. Since high definition multimedia content requires significant compression rates (50:1 [26]), lossy encoding techniques present an attractive choice. This is further supported by the leniency of the human visual and auditory system to certain type of distortions. So, the optimal quality is usually achieved in a balanced combination of the parameter values that results in minimal use of resources and a satisfactory level of accuracy.

Each digital video encoding process produces video streams with specific bit-rates. The bit-rate is directly linked to the quality of the video stream and most encoders accept a bit-rate setting as input. It can be either set as a soft (indication) or as a hard (constraint) limit on the encoder in constant bit-rate encoding. In variable bit-rate encoding (VBR), the indicator is usually a quality setting. Therefore, the stream is with constant quality rather than having a constant bit-rate. Based on this setting and the complexity of the video, the encoder compresses the video with a certain average bit-rate. Therefore the bit-rate required to encode the video depends on the type of encoding algorithm, the complexity of the video and the desired quality. Since transport throughput is also a limited resource, the video bit-rate needs to be adjusted accordingly in order to meet the transport network constraints. This is commonly achieved by compressing the video with constant bit-rate encoding (CBR). Typically, MPEG-like algorithms will introduce increasingly larger amounts of artifacts (such as blockiness and blurriness) as the bit-rate is reduced [12]. In other words the video data will be more coarsely quantized in the frequency domain, which will lead to blockiness in the decoded video. The encoder attempts to limit the blockiness effect on low spatial frequencies of the video, which are less perceptible to the viewers. However, very constrained compressions result in highly visible artefacts. Another type of artifact due to encoding is blurriness. This one arises from inadequate temporal fidelity of the encoded video [27].

The loss of fidelity can originate from the pre-encoding process as well. The spatial resolution of the digital video is one of the key factors for the size of the video after encoding. The recording equipment usually has much higher resolution than what can be practically used in video streaming applications. Particularly for mobile devices the resolution needs to be adjusted to the limitations of the devices in screen resolution, computational



power or network throughput. It is common to downscale the resolution of the original video before encoding, because it reduces the encoding computation time as well as the size of the resulting video. In cases where the target screen resolution is lower than the input video this pre-encoding process is only beneficial. On the other hand, restricting only the bit-rate can degrade the video more, add more computational on the encoder and on the decoder for downscaling.

In addition to the spatial resolution decrease, there is temporal resolution decrease, or decrease in frame-rate. Frame-rate is usually kept to less than 30 frames per second due to the characteristics of the human visual systems. However frame-rate acceptability depends on the type of content [28]. Certain types of content that have low mobility and small spatial resolution, frame-rates as low as 10 frames per second can be acceptable. This is particularly useful for very low bit-rate channels in mobile environments where lower frame-rates help achieve the required low throughput. Similar pre-encoding can be implemented in the audio, when sampling rate and sampling frequency are downscaled.

Making the right decisions during the encoding process is key to minimizing the amount of delivered artifacts. Rate distortion theory provides a theoretical foundation for this problem. The theory is a branch of information theory and deals with lossy data-compression [29]. Based on this theory many rate distortion optimization (RDO) methods have been developed and are commonly incorporated in the decision process of multimedia encoders [30].

Another reason for occurrence of artefacts is errors incurred during transmission. The amount of degradation in quality due to such errors is not easy to estimate due to the very nature of video compression [31]. The removal of temporal redundancy in the video leads to propagation of the errors in multiple frames. This can be constrained by adding more I-frames or reference frames that do not require frame from other data to be compressed. However, increasing the frequency of the I-frames decreases the efficiency of the compression. Another approach to protect the data stream is to use forward error correction techniques [32]. These techniques add redundant data to the stream, which allows recovery of a limited number of bits in case of errors or loss during transmission. Selecting the appropriate amount of redundant data can be achieved by applying RDO to this problem as well [33].

When transport protocols ensure delivery via packet retransmission the

same mechanisms can cause delays and hence lower throughput. The network throughput can also fluctuate, causing jitter in the arrival time of the packets. This can result in difficulties for streaming of video as the decoder cannot wait for late video packets. Buffering is used to compensate for the effects of delay and jitter. However, in order to compensate for high variation in packet arrival time significantly large buffer is necessary. This imposes high memory requirements on the client but it also has a direct effect on QoE, because it increases the start-up time of the playback. This effect can be sometimes more damaging to QoE than lowering the bit-rate.

In this thesis we demonstrate a number of objective and subjective QoE measurement techniques applied on a limited range of factors. We have developed implementations on existing techniques and certain extensions where the state-of-the art did not produce desired results. The results from these measurements deliver valuable information on the effects of these factors on QoE.

Nevertheless, in the effort to capture a fuller picture of the QoE of an operating multimedia system, it becomes evident that there are a large number of factors in play and there are an equally large number of parameters in the system that contribute to this factors. The relationship between the parameters, resources and technology in the system can be complex and highly non-linear. So, there is a clear need to deal with this complexity and uncover the relationship between the system intricacies and the QoE.

## 1.4 Handling the complexity of QoE modelling

Our ambition is to create a framework for QoE-aware management of multimedia services. In order to do that we need to be able to understand how our management decisions are affecting the QoE. Yet, as QoE is multifaceted and has complex relationships with the available resources in the system, its modelling presents a challenge.

Many subjective studies of certain aspects of QoE (including our own) have been executed. Most of them are focused on the efficiency of the encoding [34], while others on the effects of specific errors on the quality. However, considering all the factors that affect quality in a subjective study would be an insurmountable task.

A more tractable way to deal with the multitude of factors that affect QoE is to use computational techniques. Computational intelligence and, more concretely, Machine Learning techniques offer the ability to correlate

a vast amount of parameters with each output metric. They can discover complex interdependencies and detect non-obvious patterns. QoE models developed by combining objectively and subjectively measurable factors can deliver much better understanding of the delivered QoE to the viewer than by just looking at individual parameters such as bit-rate or video resolution.

In this thesis we present a system that collects a multitude of measurements from a multimedia system and correlates this information with subjective feedback from the users. The models delivered from the correlation can be further used to estimate the performance of the system over a longer period of time.

On-line learning techniques can be used to continuously adapt these models and deliver an accurate estimation of QoE, even in a continuously changing environment. The understanding of QoE can deliver an efficient longer term management cycle of monitoring, evaluation and provisioning. Despite that, when faced with active control or short-term management decisions we need to understand the effect of each decision on the QoE. For this type of management instead of inference and modelling we need to move on to optimal control strategies.

In the following chapters we present description of a QoE management framework that addresses the challenge of complexity and adapts in an on-line fashion. We also present an approach of QoE-aware active control of multimedia systems, where short term decisions are made in correspondence with the fluctuations in the available resource.

## 1.5 Learning vs. Deterministic Design

Management of networked services typically means provisioning enough resources and allocating them appropriately. However, as certain resources are shared over the systems their availability varies over time. For many applications, making real-time decisions on the available resources makes the difference between delivering high quality service and failing to do so.

The usual approach in developing an efficient controller for real-time management is to design a suitable heuristic (or a rule based system) that will take appropriate decisions based on the state of the system. This approach requires a thorough understanding of the effects of the decisions on the performance of the system. As complexity in the system grows the design of efficient heuristics becomes more challenging and more expensive.

As the system evolves rules become outdated.

In some areas, such as video encoding and video streaming RDO methods have been implemented to optimize the trade-off between quality and resources. Even though these methods have sufficient theoretical basis, practically the models that they rely on to calculate the rate and distortion do not fully capture the complexity of different video sources [26]. Furthermore, with the growth in the complexity of the systems, the interdependencies between the decision are not fully taken into account [35].

In contrast to this methodology, in this thesis we present an approach of 'learning' optimal strategies rather than 'designing' them. A computational intelligence technique based on reinforcement learning is used to discover the longer term utility of the decision, given the state of the system, and develop an optimal strategy.

This technique relies on previous techniques for modelling QoE and on well-established methods for reinforcement learning, offering an approach for designing system control that is scalable and adaptable to the changes in the environment.

## 1.6 Main contributions

The focus of this work is developing methods for efficient management and control of delivered quality in multimedia services.

The main challenges facing this goal are understanding the different factors that affect the quality, the growing complexity in the interaction of those factors and the effect of the management decision on the quality.

In order to understand the delivered quality, we have defined QoE as its metric, discussed the factors that affected it and the resources that relate to these factors.

In Chapter 2 we continue to present a discussion on objective QoE methods and our implementations and developments of supporting techniques. Objective QoE methods are a cost effective way to measure the factors the contribute to the delivered quality. Their use is widespread, and they correlate in varying degree with the subjective QoE.

To understand the delivered QoE thoroughly, we turn to the subjective QoE methods in Chapter 3. This chapter presents a discussion on the existing subjective QoE methods and their drawbacks. Furthermore, it introduce a novel video subjective method based on psychometric evaluation that addresses many of these challenges.

Multimedia delivery systems are typically complex and their successful management requires a broader approach. In Chapter 4 we present our framework for QoE monitoring and provisioning that learns how to model all available measurements into a QoE value. Moreover, we provide a solution for calculating the remedies in systems where the measured values are not satisfactory.

In following chapter (Chapter 5) we present our approach for real-time management or control of a multimedia system that infers the control logic based on the measured QoE.

The work on objective and subjective QoE models builds a basis for the QoE management and the QoE active control framework. These two frameworks offer a method for efficient management and control of the quality in multimedia services faced with growing complexity and continuous evolution of both the user expectations and the underlying technologies.

# 2

## Objective QoE Models

We have seen how QoE is a metric that relates to our subjective expectations. Even though these expectations are not objectively measurable, many factors that contribute to them are. For example we can measure the loss of IP packets in the network and make an estimation on the effect that this will have on QoE. Similarly, we can measure the amount of signal degradation that a lossy compression process inflicts on the content. These measurements do not convey the exact difference between the expected and the delivered quality in a general case, but for more specific uses they can provide a good indication. The models that contain objectively collected measurements of factors that affect QoE are referred to as objective QoE models.

The main motivation for the use of the objective methods is that the objective factors can be measured precisely and at a lower cost than subjective assessment. Furthermore, many such methods can be deployed on a wide range of systems and their operation can be efficiently automated. Due to this, objective methods are frequently used for modelling the system quality and in-turn optimizing multimedia services.

Since video quality has such a significant importance on the overall QoE management of multimedia streaming services [36], this section is dedicated to a review of a range of methods for objective video quality assessment (VQA). VQA has been of significant importance since the early days of digital video, so many methods have been devised. The objective quality methods are further divided into three groups based on the level of involvement of the original reference signal in the estimation. The Full-reference (FR) methods require the original material in its entirety. They operate by comparing the original with the impaired material to calculate the degradation. The calculation ranges from simple algorithms, such as signal error estimates, to very complex ones that incorporate many HVS characteristics in the estimation. The Reduced-reference (RR) methods use parts or digests of the original material for the comparison calculation.

They are better suited for situation where the original content is difficult to store or transport to the place of estimation or computational power is limited. Finally, No-reference (NR) methods do not use any part of the original content. They do not rely on comparisons but on measurements of external factors to model the QoE. The NR methods often are significantly restricted for specific applications and are not applicable for general use, but require the least resources and are useful for cases where the original content is not available.

As the goal of this thesis is to develop efficient methods for managing QoE of multimedia services, first we need to understand how to measure it. In light of this, we present an overview on the most commonly used objective QoE metrics as well as our experimental analysis of them in the rest of this chapter.

## 2.1 State of the art

There are a vast number of objective video quality assessment methods [37, 38]. Some have evolved from image quality assessment, others have been particularly designed for video. They range in complexity from simple and easy to implement to very complex and computationally expensive. They also vary in performance, some are very restricted with limited correlation with the subjective QoE, and others are much better correlated. The rest of this section presents a set of representative objective metrics, from very simple with low correlation to very complex with high correlation with subjective QoE.

### 2.1.1 PSNR

Peak signal to noise ratio (PSNR) is one of the most commonly used FR VQA. The method is designed for a more general use, as it computes errors in any type of signal, and is also intensely used for image quality assessment (IQA) and VQA due to its simplicity.

PSNR estimates the difference between the original image and the distorted one by calculating the mean squared error (MSE) (Equation 2.1) between the two signals and giving the ratio between the maximum of the signal and the MSE (Equation 2.2), where  $x_{ij}$  is the value of the pixel in the original image at coordinates  $(i, j)$ ;  $y_{ij}$  is the value of the pixel at the same coordinates in the impaired image; and where  $MAX_I$  is the maximum

amplitude of the pixel values in the image.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (x_{ij} - y_{ij})^2 \quad (2.1)$$

$$PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right)^2 \quad (2.2)$$

Regardless of its significant drawbacks (mainly its low correlation with subjective estimations) [39–43], PSNR is still very present in video quality analysis. It is easy to compute and provides a first impression on the quality achieved.

Different studies in VQA have shown that PSNR shows certain level of correlation to subjective quality when small number of factors is considered [44]. Typical example is the quantization level effect on quality during the compression of video. When all other factors are constant the effect of the quantization in the encoder tends to correlate well with PSNR. In fact, PSNR is also used for quality decisions in the encoder [45].

### 2.1.2 SSIM

The structural similarity index (SSIM) is a method that was originally developed for IQA [46], but is widely used for VQA as well. SSIM does not purely focus on bit-errors, but more on the changes in the structure of the image. In this way it addresses some of the drawbacks of PSNR, such as susceptibility to changes in brightness and contrast. The HVS demonstrates luminance and contrast masking, which SSIM takes into account while PSNR does not. The HVS demonstrates light adaptation characteristics and as a consequence of that it is sensitive to relative changes in brightness. This effect is referred to as luminance masking. On the other hand, changes in contrast are less noticeable when the base contrast is high than when it is low. This effect is referred to as contrast masking.

The SSIM index executes three comparisons, in terms of luminance, contrast and structure. The output value is a combination of these three comparisons as given in Equation 2.3, where  $x$  and  $y$  are vectors containing the pixel values of the original and the impaired image, respectively.

$$SSIM(x, y) = f(l(x, y), c(x, y), s(x, y)) \quad (2.3)$$



For the luminance comparison, first the mean luminance is calculated (Equation 2.4).

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i \quad (2.4)$$

Then a luminance comparison is executed as in Equation 2.5.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (2.5)$$

The value of the parameters  $C_1$  is set to  $(K_1L)^2$ , where  $K_1 \ll 1$  is a small constant and  $L$  is the dynamic range of the pixel values. The average luminance is removed from the signal amplitude and a contrast comparison is computed. The base contrast of each signal is computed using its standard deviation (Equation 2.6).

$$\sigma_x = \left( \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}} \quad (2.6)$$

The contrast comparison is computed as given in Equation 2.7.

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (2.7)$$

For the structure comparison, the average luminance is subtracted and is divided by its base contrast to normalize it. A Pearson correlation coefficient is calculated as a measure of structural similarity (Equation 2.8 & 2.9).

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (2.8)$$

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y) \quad (2.9)$$

The SSIM output is a combination of all three components (Equation 2.10).

$$SSIM(x, y) = [l(x, y)]^\alpha [c(x, y)]^\beta [s(x, y)]^\gamma \quad (2.10)$$

This SSIM model is parameterized by  $\alpha, \beta, \gamma$  where typically the parameter values are  $\alpha, \beta$  and  $\gamma = 1$ . In order to simplify the expression the parameter  $C_3$  is set to  $C_3 = C_2/2$  (Equation 2.11).

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (2.11)$$

Since the luminance calculation 2.6 and the contrast calculation 2.7 are consistent with the luminance and contrast masking effects respectively, the SSIM metric performance is better correlated with subjective QoE [43].

### 2.1.3 VQM

Methods developed for IQA can deliver some level of indication as to the degradation of video quality. However, these methods are not designed for video and omit the temporal factors' effects on the quality. One such example is the motion masking effects, where high motion in the video decreases the effect of loss of structure on the quality [47]. A model that addresses the quality of video, taking into account the structural and temporal aspects, is the video quality model (VQM) [48].

Because of its good correlation with subjective values, VQM is commonly used as a FR method. However, VQM does not compare the original to the impaired video directly. It extracts features from the original and the impaired video separately, and then compares those features to calculate quality. This makes the method applicable for RR use as well. The features that are extracted from the original video account for 9.3% of the uncompressed size of the video. Additionally another 4.5% of data needs to be transmitted for the initial pre-processing step of VQM where both videos are calibrated in space and in time.

VQM is implemented by first applying a perceptual filter to the video stream. This enhances some properties of perceived video quality, such as the edge information. After this perceptual filtering, the video is segmented in space and time into spatial-temporal (S-T) subregions. Next the features are extracted from these S-T subregions. Finally, a perceptibility threshold is applied to the extracted features, so that only impairments above this threshold are considered.

The masking effects in the HVS imply that impairment perception is inversely proportional to the amount of localized spatial or temporal activity that is present. In other words, spatial impairments become less visible as the spatial activity increases, and temporal impairments become

less visible as the temporal activity increases. Furthermore, these masking effects interact with each other, so spatial masking has effect on temporal quality perception and vice versa. To account for these effects, in VQM the perceptual impairment at each S-T region is calculated using comparison functions. Some features use a comparison function that performs a simple Euclidean distance between two original and two processed feature streams. But more commonly, features use either the ratio comparison function or the log comparison function.

After different impairment parameters have been calculated in different spatial and temporal regions, these values need to be collapsed into a single value for the quality index. Optimal spatial collapsing function often involves some form of worst case processing, such as taking the average of the worst 5% of the distortions observed at a particular point in time. Because localized impairments tend to draw the focus of the viewer, making the worst part of the picture the predominant factor in the subjective quality decision is a good strategy. Finally spatial and temporal collapsing functions are used to produce a single objective quality value for the video sequence.

The parameters included in the VQM model are the following:

- `si_loss`: detects a decrease or loss of spatial information (e.g., blurring);
- `hv_loss`: detects a shift of edges from horizontal & vertical orientation to diagonal orientation; this might be the case if horizontal and vertical edges suffer more from blurring than diagonal edges;
- `hv_gain`: detects a shift of edges from diagonal to horizontal & vertical; this might be the case if the processed video contains tiling or blocking artefacts;
- `chroma_spread`: detects changes in the spread of the distribution of two-dimensional colour samples;
- `si_gain`: measures improvements to quality that result from edge sharpening enhancements;
- `ct_ati_gain`: accounts for the interactions between the amount of spatial detail and motion on the perceived of spatial and temporal impairments (spatial and temporal masking);

- *chroma\_extreme*: detects severe localized colour impairments, such as those produced by digital transmission errors.

The General Model is a weighted linear combination of these parameters (equation 2.12). The weights given in equation 2.12 are selected to achieve maximum objective to subjective correlation for a wide range of video quality and bit rates .

$$\begin{aligned}
 VQM &= -0.2097 * si\_loss \\
 &+ 0.5969 * hv\_loss \\
 &+ 0.2483 * hv\_gain \\
 &+ 0.0192 * chroma\_spread \\
 &- 2.4316 * si\_gain \\
 &+ 0.0431 * ct\_ati\_gain \\
 &+ 0.0076 * chroma\_extreme
 \end{aligned} \tag{2.12}$$

#### 2.1.4 MOVIE

MOTION-based Video Integrity Evaluation (MOVIE) is another VQA index that integrates both spatial and temporal aspects [49]. It implements a spatio-temporally localized, multi-scale decomposition of the reference and test videos using a set of spatio-temporal Gabor filters [50]. The MOVIE index is composed of two components. The first one is the spatial MOVIE index, which uses the output of the multi-scale decomposition of the reference and test videos to measure spatial distortions in the video. The second one is the temporal MOVIE index, which captures temporal degradations. The Temporal MOVIE index first computes the motion information from the reference video to generate motion trajectories. Then it evaluates the temporal quality of the test video along the computed motion trajectories of the reference video. In this way MOVIE attempts to account for the motion processing of the HVS and capture the intensity of the temporal distortions as would be perceived by the viewer.

Both MOVIE components work together. The spatial quality map generated by the spatial MOVIE, responds to the blur in the test video. The temporal quality generated by the temporal MOVIE, maps motion compensation mismatches on the edges.

The maps are then collapsed into two indexes. This is done by calculating the ratio of the standard deviation to the mean of the values in the

map. This statistics is known as a coefficient of variation, and is a good predictor of the subjective quality of a video.

Even though, MOVIE is an objective method that correlates well with subjective feedback from viewers, its high cost in computational power and memory limits its implementation in real-time systems.

### 2.1.5 Reduced and no reference methods

A more suitable alternative for real-time quality estimation in content delivery systems is given by the RR and NR methods. These methods are also applicable when the original content is not available, e.g. when different video filtering is applied (denoising, deinterlacing, resolution upscaling or due to storage restrictions). These methods, or models, are usually much more restricted than the FR methods. Particularly the NR methods, only deal with specific types of impairments and are not very accurate for general use.

Gunawan and Ghanbari have developed a RR method that uses local harmonic strength features from the original video to calculate the amount of impairments or quality of the affected video [51]. Harmonic gains and loss correlate well with two very common types of impairment present in MPEG encoded video, i.e. blockiness and blurriness. The features (harmonic data) in this RR method have a very low overhead of only 160 to 400 bits per second, which is a negligible amount compared to the size of the video.

Ma et al, present a RR method that generates both spatial and temporal features [52]. From the spatial perspective they define an energy variant descriptor (EVD) to measure the energy change in each individual encoded frame, which results from the quantization process. The EVD is calculated as the proportion of the medium plus high frequencies in the image to low frequencies. The EVD of the original video is then compared to the EVD of the compressed video. Due to the fact that different frequencies are quantized with different fidelity in a lossy compression process, the EVD difference will indicate a level of impairment. The temporal features are collected from the difference between two adjacent frames. On these computed difference frames, a generalized Gaussian density function is employed to extract these features. Then a city block distance is used to calculate the distance between the features of the original video and the impaired one. Finally all of the distances are combined to produce the quality index. The authors claim that even though this is a RR method it outperforms simpler FR methods such as PSNR and SSIM.

NR methods are even more flexible than RR because they are applicable to any video environment, even ones that do not have any information on the original source of the video. Naturally their accuracy and generality is highly constrained. NR models are frequently used to calculate the impact of transport errors on the delivered video. In [16] authors present a model for estimating MSE caused by packet loss, by examining the video bit-stream. A single packet loss does not affect only the pixels of the video frame that lost information in that packet but, due to the temporal compression mechanisms of MPEG videos, these errors are propagated by the motion vectors in the subsequent frames. The location of the packet is of significant importance, because different types of frames carry information of different nature. A similar approach for MPEG2 video is presented in [15]. This method uses different machine learning (ML) algorithms to predict the visibility of the lost packet on the presented video. The additional complexity that these NR methods face is that they are not aware of the decoder's approach to conceal the error. A typical concealment approach is zero-motion concealment, in which a lost macro-block is concealed by a macro-block in the same location from a previous frame. However, the visibility of this concealment depends on the content at this position, size of the screen and many other factors that are generally related to the overall QoE.

## 2.2 Models for JPEG2000 and MPEG4/AVC

During the encoding process of multimedia content the encoder continually makes decision that the delivered bit-rate as well as the fidelity of the compressed signal. Many optimization techniques are developed to help make the most efficient decisions. These optimization techniques commonly rely on objective quality models to evaluate the effect of different decisions. In this section we present a discussion on objective models used in the encoding process of images in JPEG2000 [53] and for video in MPEG4/AVC [54].

Commonly JPEG2000 implementations adopt the rate-based distortion minimization encoding approach that requires the user to specify the desired bit-rate or the desired quality level. The encoder then needs to match the desired bit-rate, while minimizing the loss of quality in the image or provide a standardized level of quality for the minimum bit-rate. For both of these modes the encoders need to understand the relationship between

the bit-rate and the quality.

The simplest models for distortion are based on the MSE calculation (e.g. PSNR). Even though these objective metrics do not correlate well with the perception of the HVS their simplicity makes them an attractive option for many encoder implementations. However, in other implementations some characteristics of the HVS are used to improve the accuracy of the MSE based models. The basic idea is to remove perceptually irrelevant information, that is, information when removed introduce artifacts that are imperceptible or below the threshold of detection by the receiver. In the bit-rate mode that would mean eliminating all artifacts up to the point that produces the desired size of image, while ordering the artifacts based on their how sensitive the viewers is to them. And in the constant quality mode, eliminating all the artifacts that create a distortion above the desired level.

One of the better understood aspects of HVS is the contrast sensitivity [55]. The contrast sensitivity function (CSF) describes the sensitivity of the human eye to different spatial frequencies. Instead of using MSE a CSF weighted MSE function is introduced, which more efficiently reports the quality [55]. Artifacts that are superimposed on a non uniform background with similar spatial frequency are much less noticeable. This type of masking effect is exploited to improve the efficiency of RDO. Furthermore, an improvement in encoding efficiency in JPEG2000 is introduced by adjusting the quantization step for each spatial frequency band individually and in accordance with the CSF. In other words, the degradation will be calculated higher per bit for spatial frequencies, for which the human eye is more sensitive to.

The perceptual distortion coding presented in [56] attempts to discriminate between signal components which are detected and are not detected by the HVS. The models approach is to 'hide' the coding distortion beneath the detection threshold and to remove perceptually irrelevant signal information. In this method three visual phenomenon are used to calculate the thresholds: contrast sensitivity, luminance masking and contrast masking. The thresholds are defined by the smallest contrast that yields a visible signal over background of uniform intensity. Constant a key factor because the HVS perception depends much less on the absolute luminance perceived, but rather on the variation in the signal relative to its surrounding background. This phenomena is known as Weber-Fechner's law [57]. After calculating the localized thresholds the method uses a probability spectral

and spatial summation model to develop an overall perceptual distortion metric. The method is suitable for generating consistent quality images at lower bit-rates.

A particular type of encoding, referred to as Embedded coding [58], is designed for sending images over a network. The images encoded with Embedded encoding if truncated are decoded into a visible image. This type of images can also be decoded progressively, improving the overall experience of the user. The image is encoded from the most significant bit-plane to the least significant bit-plane. A RDO strategy adapted to the embedded approach is the visual progressive weighting approach. In this approach much more aggressive weighting strategy is implemented in the beginning of the bit-stream with the more significant bit-planes and a less aggressive as decoding proceeds and quality improves [59].

In video coding the RDO is further complicated by the temporal component. The temporal aspect introduces additional masking effects that can be leveraged for reduction of bit-rate. However, in video coding the complexity of maintaining constant level of quality over the different frames is much higher [54].

One of the most commonly used video coding standards now is H.264 (MPEG-4 Part 10). H.264 as other standards before (H.263 and MPEG-2) uses translational block-based motion compensation and transform based residual coding [54]. The output bit-rate can be controlled by several coding parameters the quantization scale and the coding mode. Large quantization scale reduces the bit-rate, but also the fidelity of the compressed video. The RDO problem is typically divided into three subproblems: Group of pictures (GOP) bit allocation; frame bit allocation; and macroblock quantization parameter ( $Q$ ) selection. For CBR allocation, first the GOP is allocated the selected amount of bits, than this amount is distributed to all the frames in the GOP. The distribution is made in such manner that the quality of the frames is kept as constant as possible. Finally, a similar approach is taken to redistribute the allotted bits within the frame to the macroblocks [60]. Measuring the quality or the distortion is commonly done using PSNR of the compressed against the original signal.

The specification of H.264 does not define the way that the encoder is implementing the RDO, this is left to the developers. The reference H.264 software uses RDO in which the Lagrangian multiplier  $\lambda$  of the cost function  $J = D + \lambda R$  is selected taking into account the quantization value. An optimal solution for this optimization would require perfect understanding



of the characteristics of the video content and the effects on the encoder parameters on it. Since this understanding is not available, the parameters are either selected by executing several encoding passes and observing the results [61] or using models for the rate-distortion effects based on different parameter values [62]. The later approach is more favorable in many approaches where the encoding process is time sensitive.

In the transform based coding approaches the macroblock data is transformed into a set of coefficients using DCT (Discrete cosine transform). The coefficients are then quantized and encoded with a variable-length coding [62]. Due to this number of bits and the distortion for a given macroblock depend on the quantization parameter and the entropy of the coefficients.

$$R(Q) \approx H(Q) \quad (2.13)$$

where  $H(Q)$  is the entropy of the DCT coefficients. The entropy of the coefficients is typically described by a Laplacian distribution [62] and the rate function is modeled as given in 2.14.

$$R(Q) = \begin{cases} \frac{1}{2} \log_2(2e^2 \frac{\sigma^2}{Q^2}), & \frac{\sigma^2}{Q^2} > \frac{1}{2e} \\ \frac{e}{\ln 2} \frac{\sigma^2}{Q^2}, & \frac{\sigma^2}{Q^2} \leq \frac{1}{2e} \end{cases} \quad (2.14)$$

The distortion in the  $i_{th}$  macroblock is introduced by uniformly quantizing the DCT coefficients with a step size  $Q_i$ . This model defines the distortion model as given in 2.15.

$$D = \frac{1}{N} \sum_{i=1}^N \frac{Q_i^2}{12} \quad (2.15)$$

In another model He et al, present the R-D model based on the fractions of zeros among the quantized DCT coefficients  $\rho$  [63]. This model also make the assumption that the coefficients are distributed with a Laplacian distribution. The model represents the rate with a linear dependency from  $\rho$  (Equation 2.16).

$$R = \theta(1 - \rho) \quad (2.16)$$

and the distortion as given in Equation 2.17.

$$D = \sigma^2 e^{-\alpha(1-\rho)} \quad (2.17)$$

where  $\theta$  and  $\alpha$  are model parameters. The authors claim that this model improves the delivered quality by an average 0.3dB [64].

Even though, most commonly generalized Gaussian or Laplacian distribution is assumed for the DCT coefficients, methods exist that use other distributions have also been proposed (Cauchy [60], however more complex distribution lead to increased complexity in the models.

RDO optimization offers a key powerful mechanisms for optimizing the trade-off between the bit-rate and quality. However, these sophisticated the optimization models still rely on the simple objective quality (distortion) measures due to the strict restrictions in computational complexity in the domain of multimedia encoding.

### 2.2.1 Characterizing the video content

The perceived QoE depends on the type of video content as well [24]. In order to build a more comprehensive QoE model we need to incorporate information about the video content. Defining features that can be automatically extracted from the video and that will carry information about the type of content is not straight forward.

Two features that are expected to correlate well with the difficulty in compressing videos are the Spatial Information (SI) and Temporal information (TI) indexes. These can also to a certain extent correlate with the type of the content [25].

The SI index carries the amount of spatial information in each frame of the video. The more structural details or edges are present in the video the higher the SI index will be. The SI is calculated as the standard deviation over both the x and y directions of the frame (spatial standard deviation) after the frame data has been put through a Sobel filter [65].

$$SI[F_n] = STD_{space}[Sobel(F_n)] \quad (2.18)$$

The *Sobel* function given in equation 2.18 implements the Sobel filter, which is used to extract the edge structure information from the image [66]. The  $F_n$  variable refers to the  $n^{th}$  frame of the video.

The TI index carries information about the amount of temporal information in the video, or the intensity of changes in the video over time. This is proportional to the amount of movement in the video.

The TI is calculated as the difference between two frames and a spatial standard deviation on that difference (Equations 2.19 & 2.20).

$$\Delta F_n = F_n - F_{n-1} \quad (2.19)$$

$$TI[F_n] = STD_{space}[\Delta F_n] \quad (2.20)$$

Videos with similar SI, TI values tend to contain similar type of content or have similar characteristics. For example 'head and shoulders' videos, which are common for news broadcast, tend to have low SI and TI. On the other, hand videos with complex scenes and high amount of movement such as action movies will have high SI and TI. In a similar fashion, different types of content such as football matches, documentaries or music videos tend to make separate clusters of SI and TI combinations. For this reason SI and TI can be useful features to convey the type of content for the QoE modelling.

Qualitatively similar types of features can be collected as a by product from the encoding process itself [67]. In this case no additional calculation is necessary to obtain these features. Hu and Wildfeuer define two indexes, one for scene complexity C and one for level of motion M that can be computed based on the amount of data in the I frames and the P frames of the encoded video.

With a typical Group of Pictures type encoding (such as the H.264) the amount of data in the I-frame corresponds to the level of complexity in the image [67] (equation 2.21). This is due to the fact that the more low frequency components there are in the image the more bits the encoder needs to use in the I-frame to encode the frame if the encoding is set to a constant quality mode. Similarly the P-frame corresponds to the amount of changes that happened since the last I-frame, so it correlates well with the amount of movement in the video (equation 2.22).

$$C = \frac{B_I}{2 \cdot 10^6 \cdot 0.91^{QP_I}} \quad (2.21)$$

$$M = \frac{B_P}{2 \cdot 10^6 \cdot 0.87^{QP_P}} \quad (2.22)$$

In equations 2.21 & 2.22,  $B_I$  and  $B_P$  correspond to the number of bytes in the I-frames and P-frames respectively and  $QP_I$  and  $QP_P$  is the quantization parameter for the I-frames and the P-frames (measured across the whole frame-set of the coded sequences).



Figure 2.1: Snapshots from 4 of the used videos. Starting from top left image in clockwise order: river bed, park run, sunflower, mobile and calendar.

## 2.3 Experimental analysis of objective QoE

VQA indexes vary in complexity and how accurately they correlate with subjective estimations. However, in specifically constrained conditions their evaluations can bring valuable information for the delivered quality. Understanding these constraints can deliver a useful tool for a more general QoE model.

To explore this, we implement an experiment where the content is impaired only with lossy compression. The level of impairment is controlled with a constant bit-rate (CBR) level of encoding. In this setting we observe how the output of different objective VQA metrics changes with the type of content.

The raw video samples we used for this assessment are part of the Live Video Quality Database [34, 68]. The description of each video is given in table 2.1. Snapshots of four videos from the database are given in Figure 2.1.

The ten different videos were compressed with H.264 compression [69]. The PSNR index was calculated against the original uncompressed sequences. The videos native resolution is 768x432 at 25 frames per second. The videos were compressed with bit-rate settings ranging from 64kb/s

bs	Blue Sky	Circular camera motion showing a blue sky and some trees
rb	River Bed	Still camera, shows a river bed containing some pebbles in the water
pa	Pedestrian Area	Still camera, shows some people walking about in a street intersection
tr	Tractor	Camera pan shows a tractor moving across some fields
sf	Sunflower	Still camera, shows a bee moving over a sun-flower in close-up
rh	Rush hour	Still camera, shows rush hour traffic on a street
st	Station	Still camera, shows railway track, a train and some people walking across the track
sh	Shields	Camera pans at first, then becomes still and zooms in; shows a person walking across a display pointing at it
mc	Mobile & Calendar	Camera pan, for train moving horizontally with a calendar moving vertically in the background
pr	Park run	Camera pan, a person running across a park

Table 2.1: Description of the Live videos

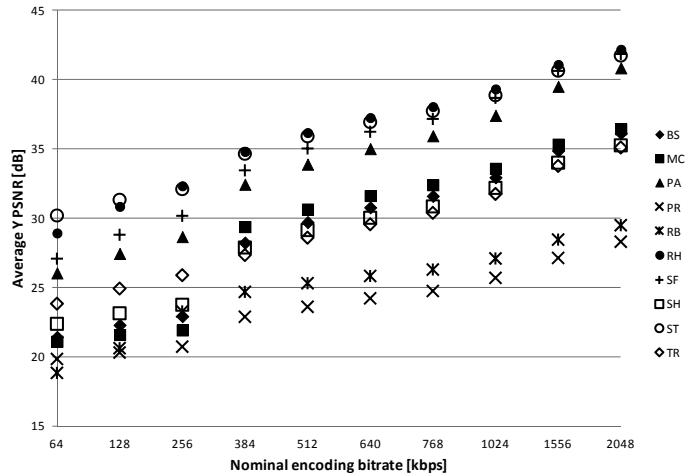


Figure 2.2: PSNR calculated quality degradation from CBR compression over different bit-rates

to 2Mb/s. To achieve the very low bit-rate of 64kb/s, the video had to be spatially and temporally sub-sampled to 384x216, at 12.5 frames per second.

### 2.3.1 PSNR Results

The PSNR calculations were executed (frame-by-frame) as defined in equation 2.2, on the luminance (Y) component of the raw original sequence and the sequence impaired with compression. The mean values for the calculations over all the frames in the figure are given in Figure 2.2

### 2.3.2 SSIM Results

The SSIM calculations were executed in a similar manner as the PSNR. Videos are impaired with different level of compression and on each pair of frames (original and impaired) SSIM is calculated. The mean values for each sequence are given in Figure 2.3

### 2.3.3 VQM Results

The VQM calculations were implemented using the VQM reference software [70]. The VQM was executed again on pairs of original and impaired videos,

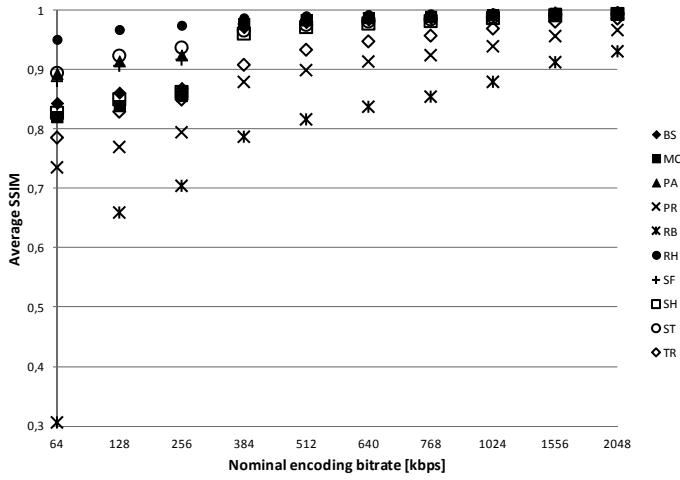


Figure 2.3: SSIM calculated quality degradation from CBR compression over different bit-rates.

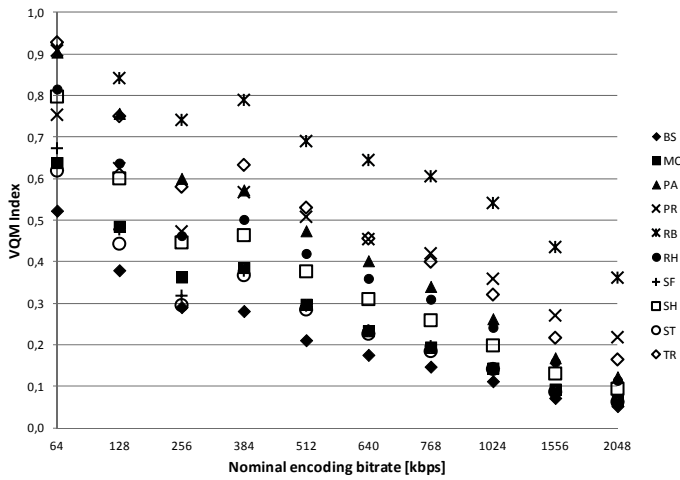


Figure 2.4: VQM calculated quality degradation from CBR compression over different bit-rates

for each level of impairment. The results of the VQM calculations are given in Figure 2.4.

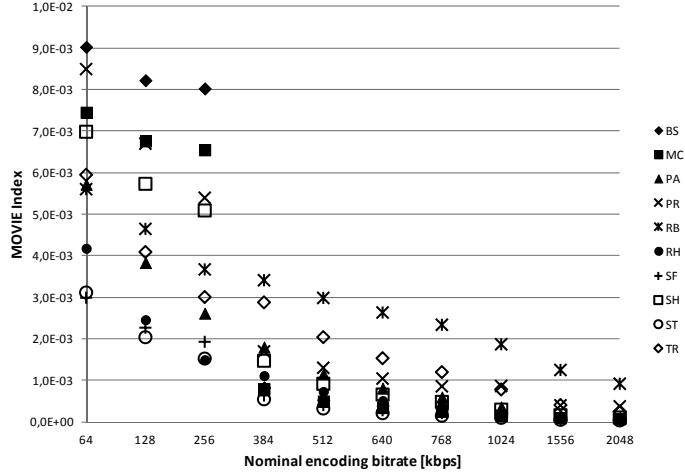


Figure 2.5: MOVIE calculated quality degradation from CBR compression over different bit-rates

### 2.3.4 MOVIE Results

Finally we executed the MOVIE index on the database of videos we developed for this experiment. The MOVIE reference software was received courtesy of the authors [71]. The results are shown in Figure 2.5.

### 2.3.5 SI, TI, C and M Results

The description of each type of video can be obtained from the Live database. However, in order to objectively generalize on the types of video we set out to compute the spatial and temporal features of the videos. In this way we can analyze if there is any correlation between the spatial and temporal features of the video and the VQA indexes output. The results of the SI, TI, C and M estimation are given in Figure 2.6 and Figure 2.7 respectively.

## 2.4 Discussion and conclusions

The results from the objective VQA show that the PSNR index seems linearly proportional with the selected rate of bit-rate increase (Figure 2.2). In contrast, SSIM shows more nonlinear drop in estimations (Figure 2.3). This is more akin to what would be expected from a subjective evaluation.



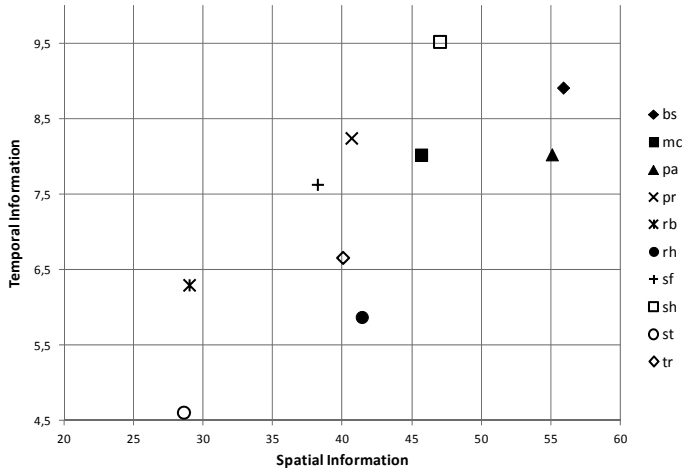


Figure 2.6: Spatial and temporal information of the videos part of the objective VQA

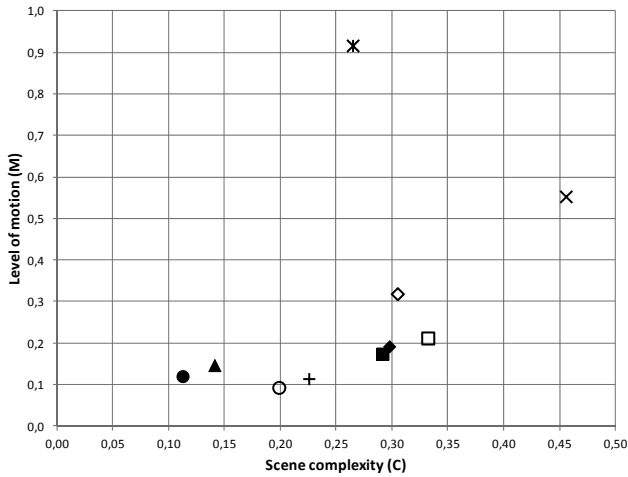


Figure 2.7: Scene complexity and level of motion in the videos part of the objective VQA

It is perceptually evident that the videos in the lower bit-rate range are significantly more degraded than in the higher range. The VQM results present an inversely proportional quality index to the level of bit-rate. But the values do not demonstrate the level of nonlinearity that one would expect for this set of bit-rate levels (Figure 2.4). Finally, the MOVIE index demonstrates a clearly emphasized non-linear response (Figure 2.5). Above 256kb/s the quality index drops to much lower values (lower is better). This indicates that MOVIE finds the improvement in quality with much higher gradient in the lower bit-rate region than in the higher region. These results are better correlated with the measured subjective perception of quality both by the authors [34] and our own measurements discussed in the following chapter [72].

The evaluation of the SI and TI does not show a clear representation why certain videos are compressed with less quality than others using the same bit-rate. However, there are some indications. The videos that have both low SI and high TI demonstrate worse VQA indexes than the others (Figure 2.6). On the other hand from the results of the 'scene complexity' (C) and 'level of motion' (M) indexes (Figure 2.7) the relationship to the performances is much more evident. Clearly the worse performing videos 'river-bed' (rb) and 'park run' (pr) stand out with exaggeratedly higher C and M values. This high correlation is to be expected since the C and M values are derived from the encoder directly. Nevertheless, this presents a good indication of the quality of these features for characterizing the type of content in QoE models.



# 3

## Subjective QoE Models

The subjective QoE methods are concerned with quantifying the experienced quality of the users. Because these methods measure the subjective quality in an unmediated manner, their measurements are commonly used as 'ground truth' for evaluation of other methods [36].

There are different approaches to subjective evaluation. The most direct technique is the rating approach, where the participants are asked to rate the quality of the content on different scales. The motivation here is that the participants can directly report the level to which their expectations have been met. However, research in psychophysics on measurements of subjective values demonstrates that the rating approach has significant drawbacks [73], mainly due to the high bias and variability in the results.

Other subjective evaluation methods such as the Just Noticeable Differences (JND) and the method of limits focus on estimating the parameter value for which the impairment becomes perceptible. The quality can be considered as acceptable as long as the change in parameters results in no perceivable difference using the method of limits [74]. In JND the smallest change in a parameter that results in detection is defined as 1 JND. Then the amplitude of the subjective value is measured using this unit, the JND. In this manner JND quantifies the amount the degradation of quality.

On the other hand, more recent research in psychophysical methods indicate that difference scaling methods show the best performance [75]. These methods can deliver the relative differences in quality between different video samples and quantify the quality in a relative way [73].

The reason for the many different approaches in subjective quality measurements is mainly due to the difficulties associated with accurate measurement of subjective values. In the following section we will examine the commonly used methods and present a more detailed discussion on the advancements we have made in this area.

### 3.1 State of the art

In this section we present a discussion on existing methods for video quality estimation. The second is divided in two sub-sections, the first one focuses on methods that use rating, and the second focuses on the estimations with limits and noticeable differences.

#### 3.1.1 Rating the quality

The method most commonly encountered in the literature for subjective video quality evaluation is the rating method. This method has been standardized by the International Telecommunication Union (ITU) in [76]. In the recommended setup for a subjective study the viewing conditions are strictly controlled, 15 or more non-expert participants are selected and trained for the exercise. The grading scale for absolute category rating (ACR) is defined as Mean Opinion Score (MOS) and it has five values: 1 - 'Bad', 2 - 'Poor', 3 - 'Fair', 4 - 'Good', 5 - 'Excellent'. Similarly for impairments: 'Very annoying', 'Annoying', 'Slightly annoying', 'Perceptible' and 'Imperceptible'. There is also a comparisons scale for Differential MOS (DMOS) going from 'Much worse' to 'Much better' with an additional value in the middle 'The same'.

A typical example of a subjective assessment using ACR has been executed in [77]. The assessment is on quality degradation of error-prone network transmission. The subjective study was executed on 40 subjects in order to provide for a database for further evaluation of FR, RR and NR objective VQA.

The effect of interactions between bit-rate levels and temporal freezes in video playback on the quality have been evaluated in [78] also by means of an ACR subjective study. They concluded that the video quality is affected by both types of impairments, however the temporal impairment have a more intensive effect. Furthermore, introducing the second impairment affects the effect on the perceived quality from the first. They developed a non-linear model integrating both impairments to predict the overall quality.

The Live video database [68] contains a set of test sequences impaired with two types of compression methods and two types of transports errors. In this study the subjective test was implemented using the DMOS [79] scale. The study was executed to evaluate the performance of many objective VQA methods, using subjective data as benchmark. Thus, the

objective methods (PSRN, SSIM, MS-SSIM, ... , MOVIE) were evaluated by their correlation with the subjective data. The results indicate that the MOVIE VQA index delivers the best performance overall.

The Video Quality Experts Group (VQEG) reports the results of their recent subjective study of High Definition Television (HDTV) in [80]. They explored the effects that compression quantization level, bit-rate, transmission errors and different concealment strategies have on the perceived quality. They produced a linear model of the video quality and the impairment factors using two or ten parameters.

Evidently the rating method is widely used. Even though, the method gives a sense of direct measurement of the human perceived quality, it is based on a flawed concept. This type of 'direct' psychophysical measurements dates back to the work of Stevens (1957) [81]. Stevens claims that there are internal psychological scales that can be empirically measured by directly inquiring the 'how much' question. However, later work in psychophysics uncovered that such direct methods are inheritably biased due to the qualitative nature of the scale ('Poor', 'Bad', 'Fair') and present a high level of variance [82] [83] [84] [85]. In fact, Shepard continues states that: "Although the (human) subject himself can tell us that one such inner magnitude is greater than another, the psychophysical operations that we have considered are powerless to tell us anything further about 'how much' greater one is than the other." [86].

In much later work, similar conclusions have been reached by video quality experts. Watson concurs that that the brain perceptual system is more accurate at grasping 'differences' rather than giving absolute rating values [87]. In his analysis of the properties of subjective rating, Winkler discusses the MOS variability and standard deviation in a set of subjective databases [88]. The analysis shows that the standard deviation of MOS for the midrange is between 15 – 20% of the scale, although it decreases at the edges. This is confirmed also by a recent study by the video quality experts group [72, 89]. The diagrams depict the standard deviation for Mean Opinion Score (MOS) rating tests and Differential Mean Opinion Score (DMOS) tests (Figure 3.1). The variance is presented on the vertical axis as a percentage of the rating scale.

It is evident that the variance of the results is in the range of 15 – 20% of the scale. Particularly the middle range of the scale has variance reaching 20% and above. Thus, if we aimed to map the MOS scale to quality labels such as 'Poor' (lower end of the x scale), 'Fair' (on the middle) and 'Good'

(high end of the scale); the results would hardly be reliable. These diagrams give a general feel as to the actual perceived quality but cannot be used to directly map quality labels onto a continuous axis or to draw any conclusive results [90]. Furthermore, the question remains as to whether the distance between the 'Poor' and 'Fair' value on the x-axis is the same as the distance between the 'Fair' and 'Good' perception of the participant.

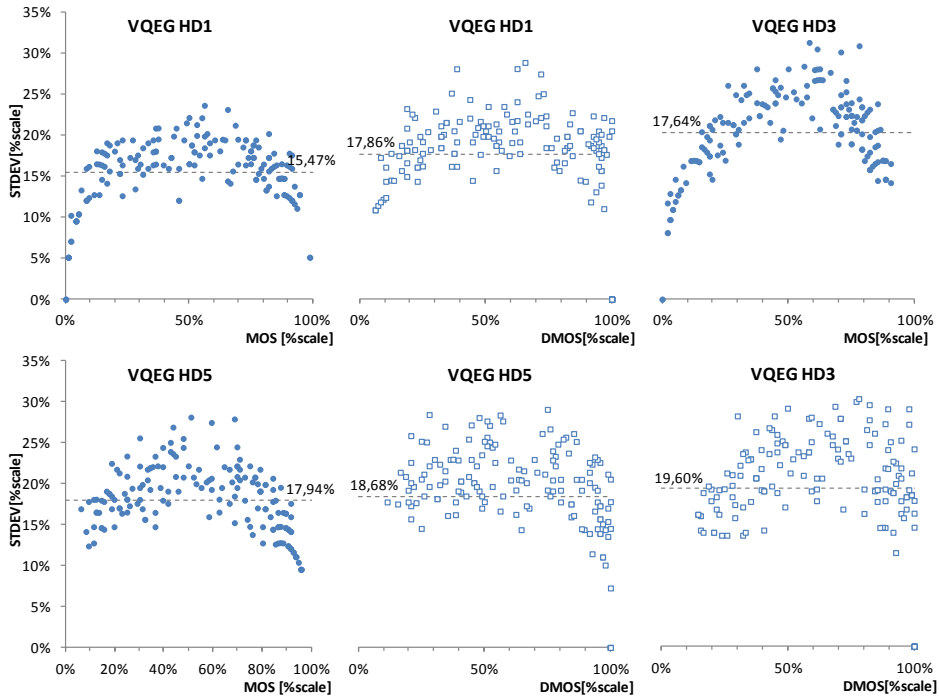


Figure 3.1: Subjective Variability [72]

An analysis of the subjective scales in [91] presents more interesting perspectives on the rating method. A striking one is that because the MOS scale is an ordinal qualitative scale it should not be used as a quantitative scale. The authors present an argument that the commonly used mapping in the literature from qualitative to a quantitative scale in rating is not justified. Therefore analysis of MOS in decimal values would be invalid as well as its variability calculated to less than one single ordinal steps in the scale. In other words, numbers associated with the 5 labels do not represent actual distance between them. To avoid these issues with the scales, the authors propose a label-free scale or labels only at the end of the scale.

This approach, however convenient for the analysis of the results, opens the question as to how people would map their internal representation of 'Good' and 'Bad' on this label-free scale. After all the main goal of the rating is to deliver a qualitatively measurable result such as 'Good' or 'Bad'.

### 3.1.2 Limits and noticeable differences

Due to the limitations of rating, other approaches have been developed for measuring the quality or degradation in video services.

The method of limits is a psychometric approach proposed by Fechner [92]. The method is designed to determine threshold level of stimuli by introducing a gradual increase until the stimulus becomes detectable. The procedure can also be run in reverse (until the stimulus becomes undetectable). The participant gives a 'Yes' or 'No' response at each level of intensity according to whether the change has been detected.

The method has been also used for determining the acceptability of video quality. In [93], McCarthy et al. use the method of limits to explore the effects of encoding quantization and changes in frame-rate on the acceptability of video quality. They implemented the study on two devices, a desktop computer and a hand held palmtop computer. The study collected data consisting of acceptability ratings for the different test conditions. These acceptability ratings were transformed into ratio measures by calculating the proportion of time during each 30-second period that the quality was rated as acceptable. The study was implemented on sports coverage videos with 41 participants on the first type of device and 37 on the second. The results of this study indicate that the participants are more sensitive to reduction in frame quality (quantization) than to changes in frame-rate. The authors claim that the results challenge the conventional wisdom that for sport events with high amount of movement, high frame-rate is necessary for high level of quality.

In a subjective study addressing the user expectations in mobile content delivery [74] the authors examined the acceptability of QoE, for different types of content and on three different devices, using the method of limits. Content types included: news broadcast, sports, video game animations, music videos and movies. They repeated the experiment in an ascending and descending order. In this study they evaluated the quality based on the video encoding bit-rate and the frame-rate of the video. The results collected from 96 participants varied significantly over different types of



content. On the mobile phone terminal, the mean acceptability thresholds for football were found to be 128kb/s with 15 frames/s, while for the Romance movie is 32kb/s with 10 frames/s. Even though, this finding about the big difference in quality vs. resources for different types of content is most interesting, the results bring light to one of the main pitfalls of this method. The method of limits presents a significant hysteresis in the results, the finding on the ascending order vary significantly from the finding on the descending order evaluation. Because of this effect, accurate estimation of the acceptability is not clear for the values within the hysteresis. This is the main drawback of this method.

The JND as introduced by Weber [94] is defined as the smallest detectable difference between two intensities of a sensory stimulus. It is a statistical value usually defined as the level detectable in 50% of the tested cases. The concept of JND was re-introduced for scaling video quality by Watson in a proposal for a new quality scaling method [87]. In [95], Watson and Kreslake execute a subjective study by asking the participants which of two presented videos is more impaired. This is called 'pair comparison'. From the responses to that simple question, they measure the observer's internal 'perceptual scale' for visual impairments. This method relies on Thurstone's 'Law of comparative judgment' [96]. Thurstone proposes that physical stimuli give rise to perceived magnitudes on a one-dimensional internal psychological scale. He continues to include an inevitable variability in the neural system. In 'Case five' of the law, the stimuli are perceived with a normal distribution with a standard deviation of 1. So if two stimuli are presented and the participant is asked to discriminate between the two, the probability of giving the right question is a function of the distance between the mean values of both probability distributions on the internal scale. In this manner by acquiring enough data the most likely values for different stimuli can be inferred statistically.

However, when participants are presented with a two different levels of quality in video, they will discriminate between two points on the internal scale. If the two points are not close enough, discriminating becomes too easy and leads to results that tend to sort this points on the intensity scale, but not scale them. In other words results from pair-wise comparison do not yield information about the distance between the points, only their order on the scale. Relying on the probability of incorrectly responding to the levels of quality is challenging and requires the ability to very finely tune the parameters of the video. This is not trivial, as encoding param-

eters usually change in discrete steps. Furthermore, subjective methods are commonly used to estimate the quality of a predetermined set allowed (or favorable) parameter values. For example 10 different levels between 64kb/s and 2Mb/s of constant bit-rate encoding, cannot be directly scaled by the proposed method without some kind of interpolation between the samples. However, the effects of the interpolation on the perceived quality in light of the complex masking effects of the HVS are not analyzed.

On the other hand if the participants are presented with two ranges of quality (such as in two pairs of videos) and asked to discriminate between the two ranges, then the intensity of quality is actually scaled. This approach is discussed in more detail in the following section.

## 3.2 Maximum likelihood difference scaling

The Maximum Likelihood Difference Scaling method (MLDS) is a psychophysical method that scales the relative differences perceived psychologically between physical stimuli [97]. In other words by executing a 2-alternative forced choice (2AFC) subjective study and statistical analysis, the method will output a relative scale of the difference between stimuli with increasing or decreasing intensities. The MLDS method has been used by Maloney and Yang as a tool for subjective analysis of image quality [97]. Furthermore, Charrier et al. show how to use MLDS to achieve difference scaling of compressed images with lossy image compression techniques using MLDS [98]. They implement a comparison of image compression in two different colour spaces, and conclude that in the CIE 1976  $L^*a^*b^*$  colour space the images can be compressed by 32% more, without additional loss in perceived quality. Their results and discussion clearly show the applicability of MLDS and the ease of collecting data with it.

The image quality study clearly presents the advantages of using difference scaling methods for applications where quality is the target measurement. Motivated by the advantages of this approach, we have developed the appropriate tools for implementation of difference scaling for estimation of quality in video.

We carry out a subjective study to estimate the quality scale for a range of videos with different spatial and temporal characteristics. The results presented demonstrate that MLDS can be used for estimating quality of video with higher accuracy and significantly lower testing costs than subjective rating.

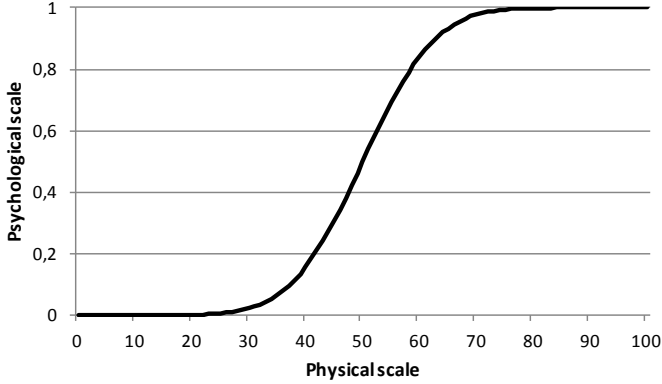


Figure 3.2: Psychometric function

As discussed above, the goal of the MLDS method is to map the objectively measurable scale of video quality to the internal psychological scale of the viewers. The output is a quantitative model for this relationship based on a psychometric function [75] as depicted in Figure 3.2.

The horizontal axis of the Figure 3.2 represents the physical intensity of the stimuli - in our study this will be the video bit-rate. The vertical axis represents the psychological scale of perceived difference in signal strength - for our purpose the difference in video quality. The perceptual intensity of the first (or reference) sample  $\psi_1$  is 0 and the last sample perceptual difference  $\psi_{10}$  is fixed to 1 without the loss in generality [99]. The MLDS produced model is an estimate of the rest of the amplitudes of the stimuli on viewers' internal quality scale.

The 2AFC test is designed in the following manner. Two pairs of videos are presented to the viewers ( $\psi_i, \psi_j$  and  $\psi_k, \psi_l$ ). The intensity of the physical stimuli is always in the following manner  $i < j$  and  $k < l$ . The method needs to compare sizes of distances between the qualities of videos so that the results can directly let us build a model of the quality distance between all of the presented videos.

The viewer needs to select the pair of videos that have bigger difference in quality between them. In other words if the expression  $|\psi_j - \psi_i| - |\psi_l - \psi_k| > 0$  is true the viewer selects the first pair, otherwise he or she will choose the second.

Because the stimuli are ordered as  $i < j$  and  $k < l$  we can safely assume

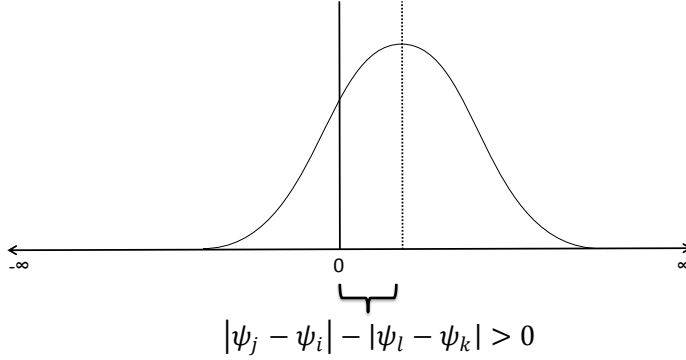


Figure 3.3: Signal of 1 unit superimposed over noise with 0 mean and standard deviation of 1

due to the monotonicity of the psychometric curve that in the psychological domain also  $\psi_j \geq \psi_i$  and  $\psi_l \geq \psi_k$  and we drop the absolute values. The decision variable used by the observer is the following:

$$\Delta(i, j, k, l) = \psi_j - \psi_i - \psi_l + \psi_k + \epsilon \quad (3.1)$$

where  $\epsilon$  is the error or noise produced by the viewers visual and cognitive processing. As defined in 3.1 the observer will select the first pair when  $\Delta(i, j, k, l) > 0$  or the second one when  $\Delta(i, j, k, l) < 0$ .

In order to use the maximum likelihood method to determine the  $\Psi = (\psi_1, \dots, \psi_{10})$  parameters, we need to define the likelihood (probability given the parameters) that the viewer will find the first pair with larger difference than the second pair. For this the method models the perceived distances using signal detection theory (SDT) [100].

The equal variance Gaussian model defined in the SDT is used to model the process of selection that the user is executing for each presented pair. This model assumes that the signal is contaminated with  $\epsilon$ , a Gaussian noise with zero mean and standard deviation of  $\sigma$  (Figure 3.3). Each time the observer is presented with a pair of videos, the perceived difference is a value of the random variable  $X$  drawn from the distribution given in Figure 3.3. The distribution in Figure 3.3 is with arbitrary signal strength of 1.

The probability that  $\Delta(i, j, k, l) > 0$  is given by the surface under the Gaussian from zero to plus infinity (Figure 3.4). For reasons of mathematical simplicity it is better to represent the surface under the curve with a cumulative Gaussian function. The inverse portion of the surface (Figure

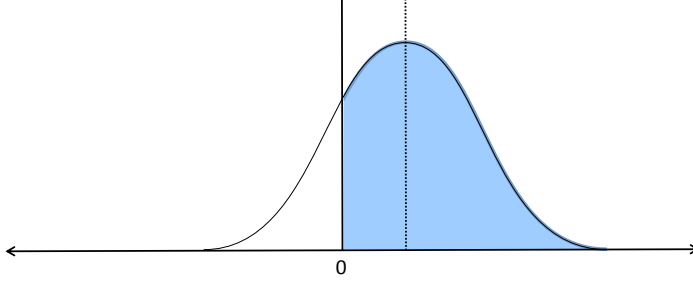


Figure 3.4: The shaded area corresponds to the probability that the signal is positive

3.5) is as in equation 3.2.

$$F(x; \mu, \sigma^2) = \Phi\left(\frac{x - \mu}{\sigma}\right) = \frac{1}{\sqrt{2\pi}} \int_x^{-\infty} e^{-\frac{(t-\mu)^2}{2}} dt \quad (3.2)$$

Looking at the inverse part of the surface under the Gaussian the probability of detecting the signal would be:

$$P(R = 1; \mu_s, \sigma^2) = 1 - \Phi\left(\frac{0 - \mu_s}{\sigma}\right) = \Phi\left(\frac{\mu_s}{\sigma}\right) \quad (3.3)$$

and

$$P(R = 0; \mu_s, \sigma^2) = 1 - P(R = 1; \mu_s, \sigma^2) = 1 - \Phi\left(\frac{\mu_s}{\sigma}\right) \quad (3.4)$$

where  $\mu_s$  is the mean or the intensity of the signal,  $\sigma$  is the standard deviation of the noise and  $R$  is 1 when the first pair is selected and 0 when the second pair is selected. The likelihood for the whole set of responses in a test is the product of all of the individual probabilities where  $\delta(i, j, k, l)_n = \psi_{j_n} - \psi_{i_n} - \psi_{l_n} + \psi_{k_n}$ .

The Maximum Likelihood method estimates the parameters, such that the given likelihood is maximized. For example, if we have  $x = \{x^t\} (t = 1..N)$  instances drawn from some probability density family  $p(x|\theta)$  defined up to parameters  $\theta$  (Equation 3.5).

$$x^t \sim p(x|\theta) \quad (3.5)$$

If the  $x^t$  samples are independent, the likelihood parameter  $\theta$  given a sample set  $x$  is the product of the likelihood of individual points (Equation 3.6).

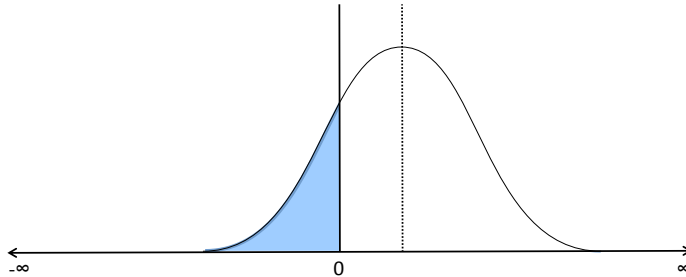


Figure 3.5: The shaded area corresponds to the probability that the signal is negative

$$L(\theta|x) \equiv p(x|\theta) = \prod_{N=1}^{t=1} p(x^t|\theta) \quad (3.6)$$

There is no closed form for such a solution, so a direct numerical optimization method needs to be used to compute the estimates (Equation 3.7).

$$\hat{\theta} = \operatorname{argmax}_{\theta} l(\theta|x) \quad (3.7)$$

### 3.2.1 The video subjective study

The experimental setup consists of a web application that displays two pairs of videos to the viewer as shown in Figure 3.6. The user response is recorded in the application database. The web application is developed using the java server pages technology [101]. The videos are displayed using the JW player [102], which is a Flash 5 web player capable of displaying H.264 encoded videos. The videos are encoded using the X264 library [103] and saved in mp4 file format.

The raw videos are the unimpaired samples of the Live video database used in the objective VQA.

Ten different videos (Table 2.1, Figure 2.1) are encoded at constant bitrate, each one at different values ranging from 2Mbps to 64kbps. The frame-rate is 25 and the spatial resolution is 768 by 432 pixels. The video player is configured to pre-buffer the full content before playing, so additional impairments such as freezes during the playback are avoided.

The results are collected in a database in the format:



Figure 3.6: Video display layout in MLDS Application

bit-rate 1	bit-rate 2	bit-rate 3	bit-rate 4	R (index bigger pair)
------------	------------	------------	------------	-----------------------

For computing the difference scales ( $\Psi = (\psi_1, \dots, \psi_{10})$ ), we used the MLDS implementation [99] in R programming language [104]. The output  $\psi$  values are fitted to a psychometric curve using a probit regression fit with variable upper/lower asymptotes using the 'psyphy' package in R [105].

The results of the subjective study are the parameters  $\mu$  and  $\sigma$  of a cumulative Gaussian (psychometric) curve that describes dependency between the QoE and bit-rate. This curve is calculated for each type of video assessed during the study.

### 3.2.2 MLDS subjective results

The MLDS experiment with 10 levels of stimuli requires 210 responses to cover all possible combinations for a single video. We have done three rounds per video sample or 630 tests for each video; in total we have collected 6300 responses. The videos are displayed one at the time or in pairs. They are 10 seconds long, so to view a single test up to 40 seconds are needed, but in most cases the larger difference is evident much sooner to most observers.

To calculate the standard error we executed a bootstrap [106] fitting

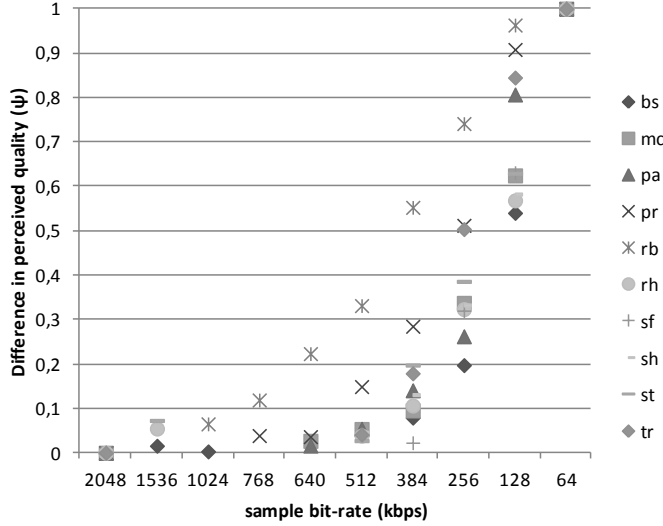


Figure 3.7: Results of the MLDS experiment by video type

procedure with 10,000 rounds. The mean values are given in Figure 3.7 and the standard deviation for each point in Figure 3.8.

The results in Figure 3.7 show that most of the videos follow a similar trajectory of the difference in quality. There is little perceived difference down to 512kbps and then a rapid rise appears. The difference is not zero in the high range, as we can also see from the standard error on the points from 1536kbps to 512kbps, but it is very low relative to the lower bit-rate samples. This means it is safe to say that there is little benefit from increasing the bit-rate above 512kbps. The exception is the 'rb' video and somewhat the 'pr' video. The 'rb' video displays a surface of water, which shows significantly different compression characteristics than the rest of the videos.

To quantitatively analyze the characteristics of each model we fitted a cumulative Gaussian curve to each difference model as demonstrated in Figure 3.9, which represents the psychometric curve [107]. There is high goodness of fit to the curve with small residuals. This further demonstrates the success of this subjective study to model the quality difference perception with a psychometric curve.

For each video the  $\mu$  and  $\sigma$  of the fitted curve are given in Figure 3.1. A



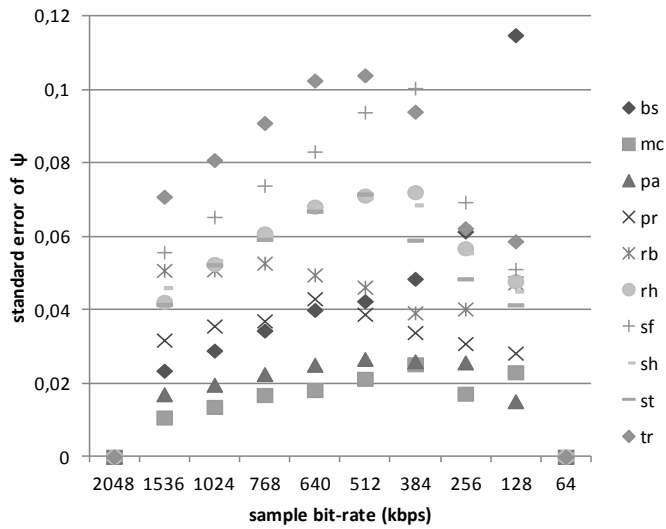


Figure 3.8: Standard error of the MLDS results by video type

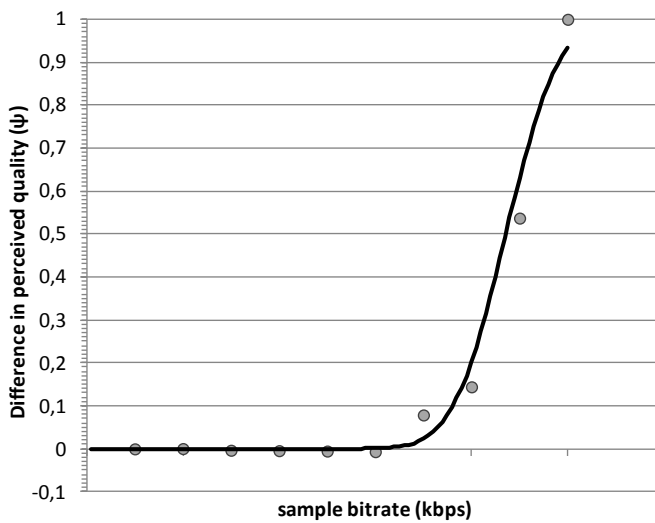


Figure 3.9: Fitting a cumulative Gaussian on the 'bs' MLDS model

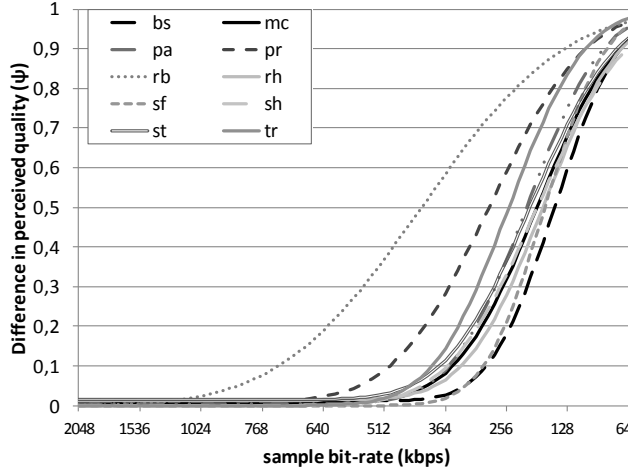


Figure 3.10: Fitted psychometric curves from the MLDS results

plot of each of the fitted models is given in Figure 3.10. The plotted curves model a smooth quality distance for different bit-rates from the reference 2Mbps video.

	bs	mc	pa	pr	rb	rh	sf	sh	st	tr
$\mu$	-5.43	-5.07	-5.08	-4.57	-4.09	-5.22	-5.54	-5.13	-5.00	-4.94
$\sigma$	0.24	0.20	0.20	0.15	0.11	0.22	0.25	0.21	0.20	0.19

Table 3.1: The  $\mu$  and  $\sigma$  of the cumulative Gaussian

Observing the parameter values in Table 3.1 we can draw the same conclusions as above, in a quantitative form. The mean value of the psychometric curve of the 'rb' video is noticeably lower than the rest of the videos, so its curve gradient increases earlier than the other video types. The remaining psychometric curves cluster together and confirm that most of these videos difference in quality is negligible to the reference, down to 512kbps, while the bit-rate between 256 and 128kbps is half way to the perceived distance between the reference and the 64kbps video. The results accurately capture the nonlinearity in the perceived quality by the viewers.

### 3.3 Adaptive MLDS

The MLDS method is appealing for its simplicity and efficiency, but is intrinsically not scalable as it considers all combinations of the samples. For instance, one full round of tests for ten levels of stimuli (i.e. video qualities) requires 210 individual tests. We developed an optimized version of MLDS, referred to as adaptive MLDS [72] to reduce the redundancy of conventional MLDS tests whilst also maintaining the reliability of the results. The strategy of adaptive MLDS is to employ active learning to minimize the number redundant tests.

#### 3.3.1 Adaptive MLDS method

This method is based on the idea that with the knowledge acquired by executing a small number of tests we can already estimate the answers of the remaining tests. Then using these estimates together with the known responses we execute the MLDS method. Executing the MLDS with more responses helps the argument maximization procedure to produce more stable solutions. The estimation of the unanswered tests is based on the characteristics of the psychometric curve.

The idea comes from the notion that some of the tests are covering the range of others. In fact, all of the tests overlap with others in one way or another. The approach makes use of the intrinsic characteristics of the psychometric curve, a monotonically increasing function  $\tilde{\Psi} = f(\vec{X})$ . Consequently, for  $k < l < m$ ,  $x_k > x_l > x_m$ , if  $x_k - x_l > x_k - x_m$  in the physical domain then  $\psi_k - \psi_l \geq \psi_k - \psi_m$  in the psychological domain (Figure 3.11).

If we now observe five samples  $x_i, x_j, x_k, x_l, x_m$  such that  $i < j < k < l < m$  and we observe two tests  $T_1(x_i, x_j; x_k, x_l)$  and  $T_2(x_i, x_j; x_k, x_m)$ , the perceived qualities in the psychological domain are  $\psi_i < \psi_j < \psi_k < \psi_l < \psi_m$ . If in  $T_2$  the first pair is bigger or  $\psi_j - \psi_i > \psi_m - \psi_k$  that would mean that  $\psi_j - \psi_i > \psi_m - \psi_k \geq \psi_l - \psi_k$ . In other words, if in  $T_2$  the first pair is selected with a bigger difference, then in  $T_1$  the first pair has a bigger difference as well (Figure 3.11). There are many different combinations of tests that have this dependency for the first pair or the second pair. We can generate a list of dependencies for each pair, based on two simple rules:

- Let us assume test  $T_1(a, b, c, d)$  such that  $a < b < c < d$ ,  $\psi_b - \psi_a > \psi_d - \psi_c$  and test  $T_2(e, f, g, h)$  with  $e < f < g < h$ . If  $e \leq a < b \leq f$  and  $c \leq g < h \leq d$  then  $\psi_f - \psi_e > \psi_h - \psi_g$  (Figure 3.12).

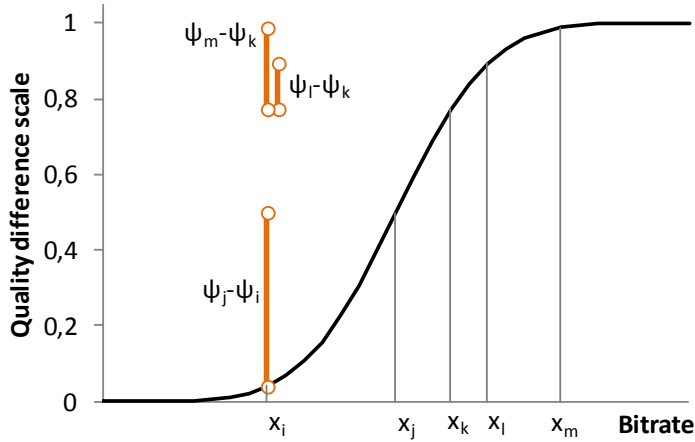


Figure 3.11: Monotonicity of the psychometric curve

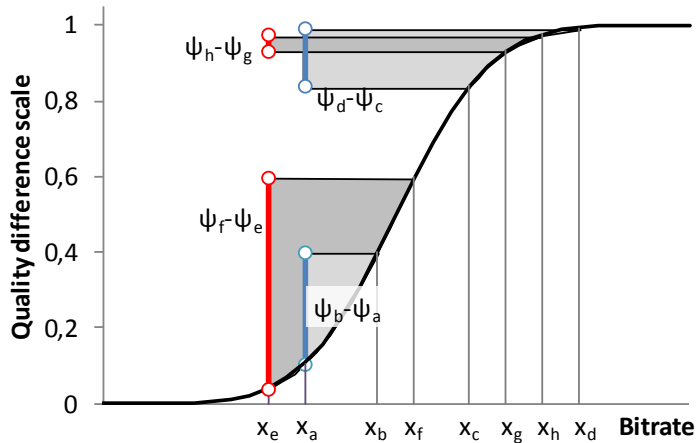


Figure 3.12: If first pair in T1 is bigger then first pair of T2 is bigger as well

- Let us assume test  $T_1(a, b, c, d)$  such that  $a < b < c < d$ ,  $\psi_b - \psi_a < \psi_d - \psi_c$ . If for test  $T_2(e, f, g, h)$  with  $e < f < g < h$  the following holds:  $a \leq e < f \leq b$  and  $g \leq c < d \leq h$  then  $\psi_f - \psi_e < \psi_h - \psi_g$  (Figure 3.13).

Thus, after introducing an initial set of responses we can start estimating the probabilities of the rest. However first we need to learn the

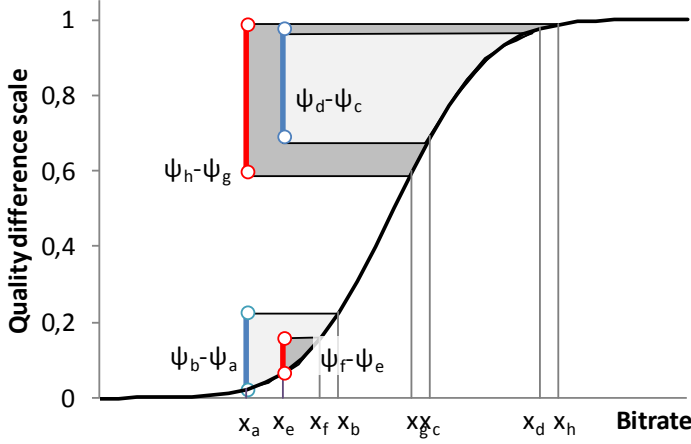


Figure 3.13: If second pair in T1 is bigger then second pair of T2 is bigger as well

probabilities of each of the known responses to be actually valid. MLDS estimates the values of the psychological parameters  $\Psi = (\psi_1, \dots, \psi_{10})$  such that the combined probabilities of each response or the overall likelihood of the dataset is maximized. Nevertheless, after the argument maximization is finished the different responses have different probabilities of being true. Having a set of initial  $\Psi$  quality values as the prior knowledge about the underlying process coming from the data, we generate the estimations for the rest of the tests. The interdependencies from the tests are far more complex, of course. Let us assume, for example, a test  $T_1$  that depends on tests  $T_2$  and  $T_3$ . If the answer from  $T_2$  indicates that the first pair has a larger difference in  $T_1$  and the answer from  $T_3$  indicates the opposite, then we need to calculate the combined probability of  $T_2$  and  $T_3$  to estimate the answer of  $T_1$ . Assuming that the responses of  $T_2$  and  $T_3$  are independent and that the probability of giving the first and second answer is the same, the combined probability of  $T_2$  and  $T_3$  is given by equation 3.8.

$$P(T_1) = \frac{P(T_2)(1 - P(T_3))}{P(T_2)(1 - P(T_3)) + (1 - P(T_2))P(T_3)} \quad (3.8)$$

Of the remaining tests that have no responses, some will have higher estimates than others. In other words, we have better estimations for some of tests than others. To improve the speed of learning, the adaptive MLDS process focuses on tests that have smaller confidence in the estimations.

This way when we receive the next batch of responses, the overall uncertainty in the estimates should be minimized. The goal of the adaptive MLDS is to develop a metric that will indicate when the amount of tests is sufficient for determining the psychometric curve. We can obtain this indication from the probabilities of the estimations. As we get more responses by asking the right questions, the estimation for the rest of the tests keeps improving. At some point adaptive MLDS will have very high probabilities of estimating correctly all of the remaining tests. This is a good indication that no more tests are necessary.

### 3.3.2 Learning convergence

The test estimation of adaptive MLDS provides a good indication for concluding the experiment. When the confidence of the estimations of the remaining tests becomes high, the probability of a surprise in the future responses of the participants goes down. With this, the need for further tests also becomes smaller. Even so, an indication of the amount of surprise from the participant responses would be useful to determine the utility of more testing.

Each batch of new data that is collected has a specific amount of information gain at each point in the experiment. The amount of information gain is proportional to the amount of surprise that the data delivered, i.e. how much the data changed the existing model. In other words when we receive responses that are completely expected our model of the differences in quality will not change at all. These responses bring no surprise and their information gain is zero. However, if the responses change our belief about the scaled quality and the model changes, then the new data has resulted in information gain. The information gain calculation is based on the Kullback-Leibler (KL) divergence [108]. The KL divergence is a way of comparing two probability distributions and produces the number of average bits that need to be used to explain this difference [109]. Using the KL divergence, the information gain in bits coming from data that change the model's distribution with mean and standard deviation to a model distribution with and is given in Equation 3.9.

$$I = \log_2 \frac{\sigma_0}{\sigma_1} + k \left( \left( \frac{\mu_1 - \mu_0}{\sigma_0} \right)^2 + \frac{\sigma_1^2 - \sigma_0^2}{\sigma_0^2} \right) \quad (3.9)$$

The information gain is giving us a tool to determine when the learning

process has converged, i.e. when additional tests would not bring any better understanding of the problem.

### 3.3.3 Experimental setup and results

To demonstrate the performance of 'adaptive MLDS', we have developed a software test-bed. The software simulates the learning process of the adaptive MLDS algorithm by sequentially introducing data from our earlier subjective study [73]. The simulation test-bed is a Java application that loads the subjective data from a file, and then sequentially introduces new data-points. The data-points are selected by the adaptive MLDS algorithm and the estimated values are used to calculate the video quality scaling in each iteration. The output is compared to the output of running MLDS on the full dataset. The root mean square error (RMSE) is computed on the differences. In parallel, a random introduction of data is also executed as a baseline for comparison. The adaptive MLDS algorithm is implemented in Java, while the MLDS software from [99] is used directly from R using a Java to R interface. To account for the variation in the results due to the random start and random data introduction in the comparison process, the simulation is repeated 100 times and results are averaged. The simulation process was computationally very demanding. Each numerical optimization was bootstrapped 1000 times. This was repeated for each step in the introduction of new batch of data and for each video. All this for a single simulation. To handle the computational demand the simulation was executed on a high-performance computing grid. Adaptive MLDS as an active learning algorithm explores the space of all possible 2AFC tests with the goal of optimizing the learning process. It also provides an indication of confidence in the model built on the subset of the data, which provides for early stopping of the experiment. The performance of adaptive MLDS is presented in figures 3.14, 3.15, 3.16, 3.17 and 3.18. Figure 3.14 shows the accuracy of the estimations for ten types of videos against the number of introduced data-points. This should be compared with plain MLDS whereby all 210 tests must be performed. As it is evident for all types of videos the accuracy of the estimations is very high. With most videos we have > 95% accuracy with just 15 tests. The Station and River bed videos show lower accuracy than the rest, but they recover quickly above 90% when around 90 - 100 data-points are presented. This indicates that for most videos we can estimate all of the tests accurately after just about 60 responses, although some videos require about 100 tests. In

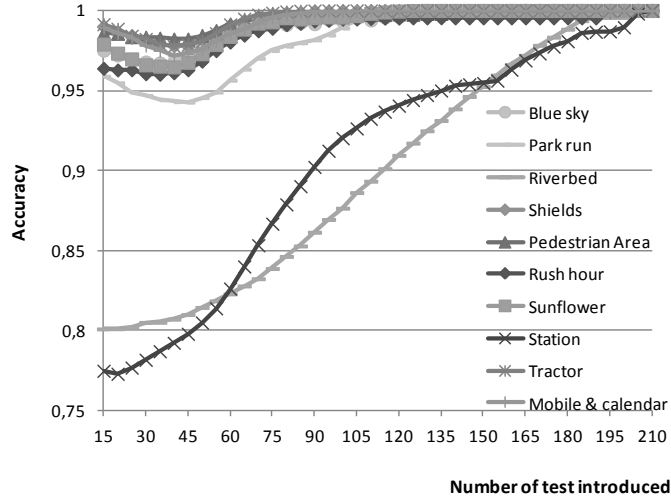


Figure 3.14: Accuracy of the estimations

Figure 3.15 we observe the accuracy of the model generated with adaptive MLDS compared to the model of the classical MLDS. The horizontal axis represents the number of points introduced at the time the calculation was executed; the vertical axis gives the RMSE (root mean square error) between the estimated values and the values computed on the whole dataset. We can clearly observe that the adaptive MLDS model differs from the 'true' model (learned from the full data set) much less than the model built by introducing data randomly. In Figure 3.16 we present the standard deviation of the different value for the RMSE at each point. The results in this figure further support the fact that adaptive MLDS produces superior results in terms of error and variability. Figure 3.17 presents the distribution of the confidence or the probabilities of those estimations. The vertical axis the number of unanswered tests is plotted with different darkness at each step of the experiment. The ratio of the shadings represents the distribution of the confidence in the estimation of those tests. Starting from the initial 15 data points most of the unknown 195 tests are estimated with 0.5 accuracy. But soon after introducing more data, the estimations rapidly improve. Between 40 and 60 collected answers the confidence in the estimations was close to 1, suggesting that the rest of the tests are not necessary and that we can correctly estimate the psychometric curve with-



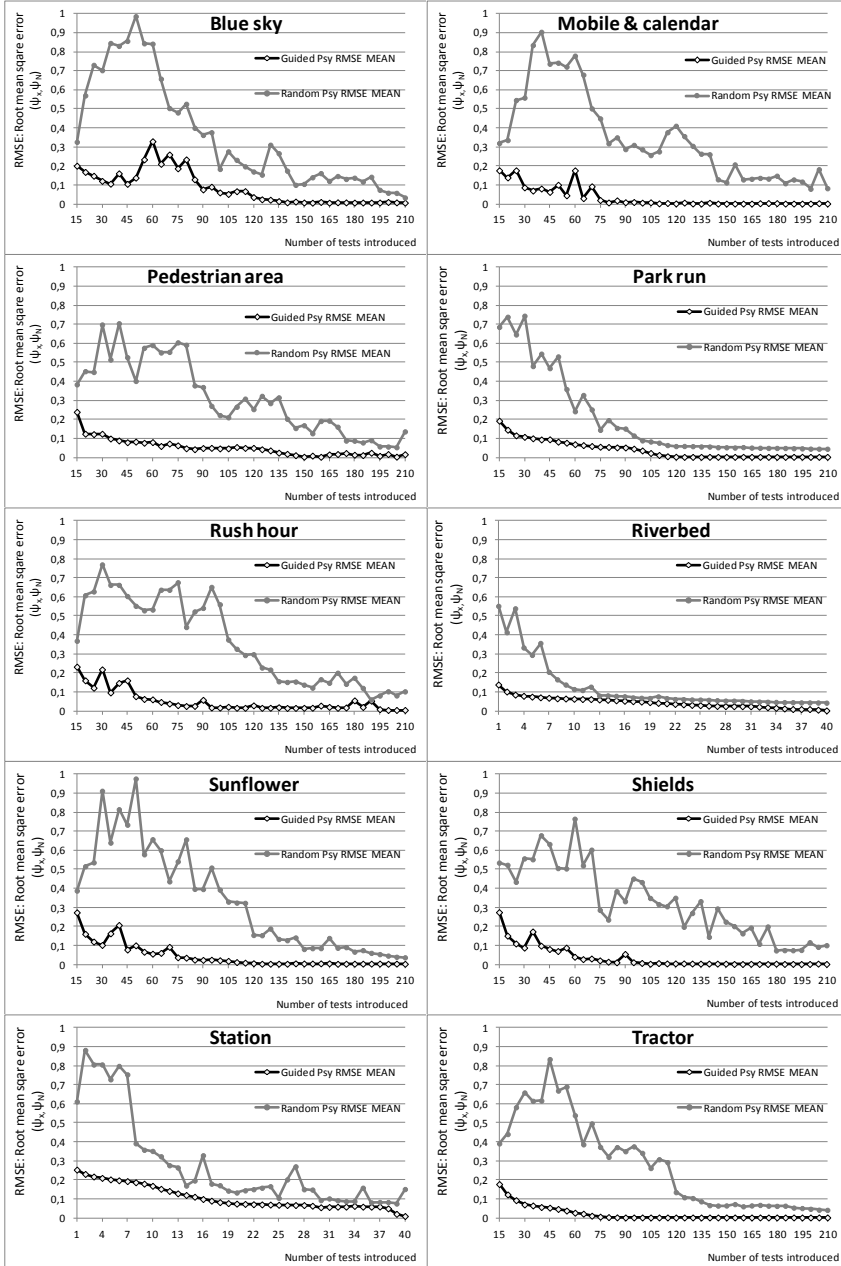


Figure 3.15: Mean RMSE for the ten types of video

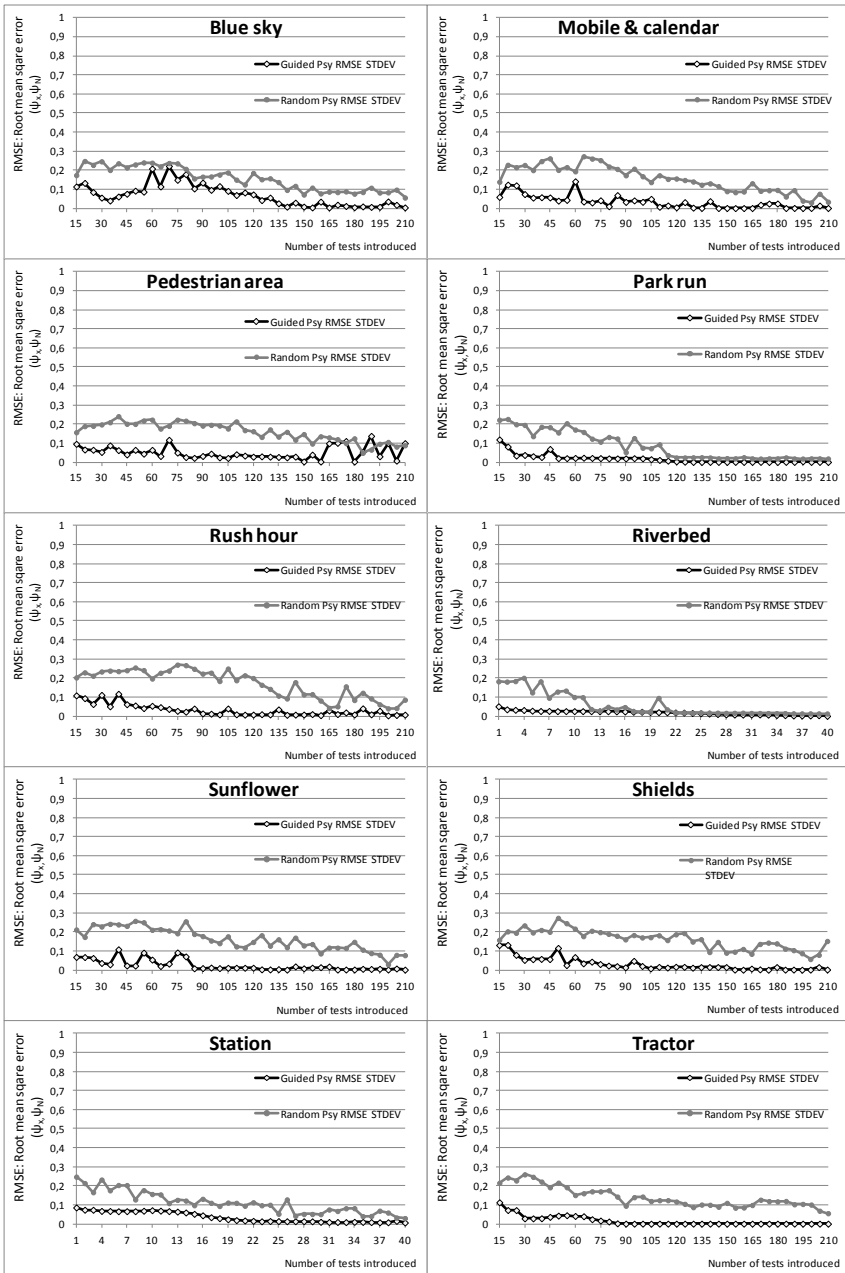


Figure 3.16: Standard deviation of the RMSE for the three types of video

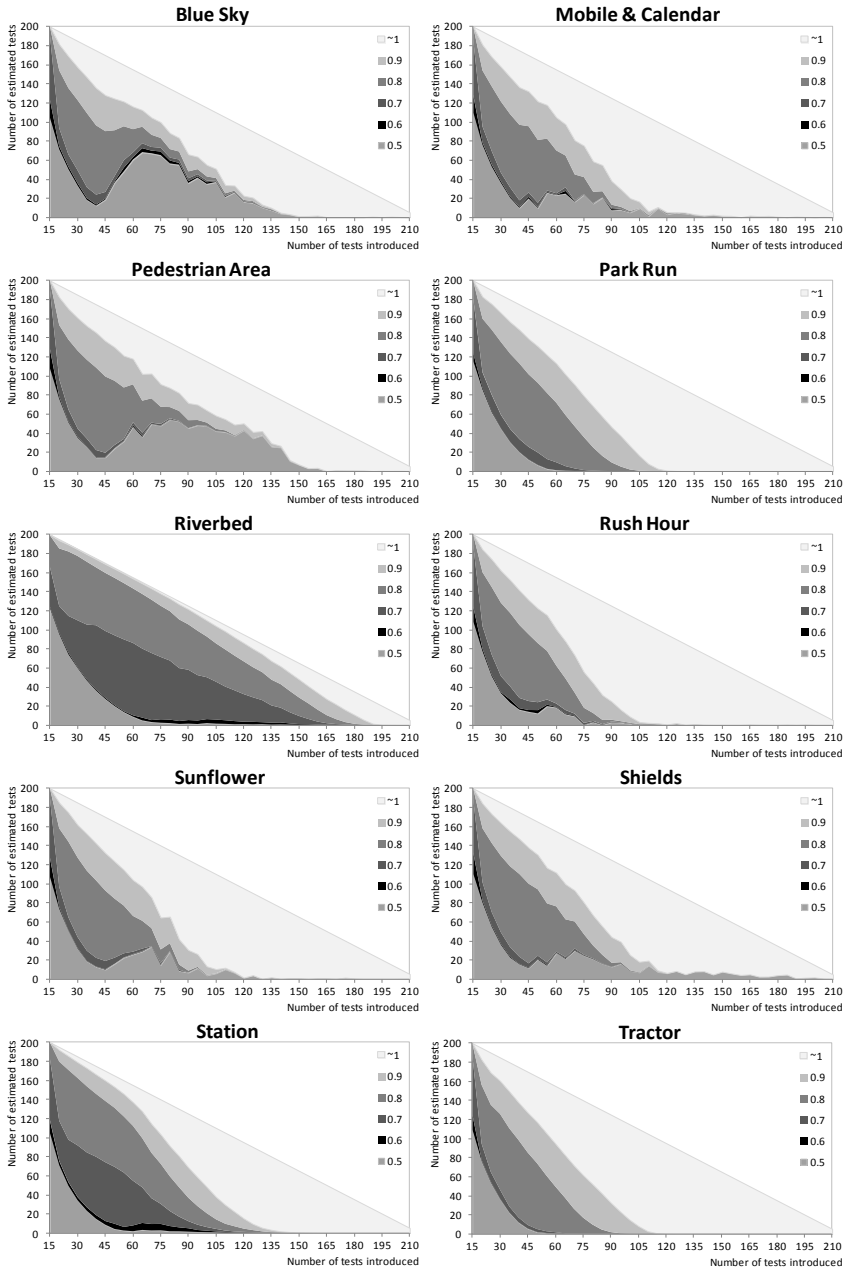


Figure 3.17: Estimation confidences for the three types of videos over the number of introduced data-points

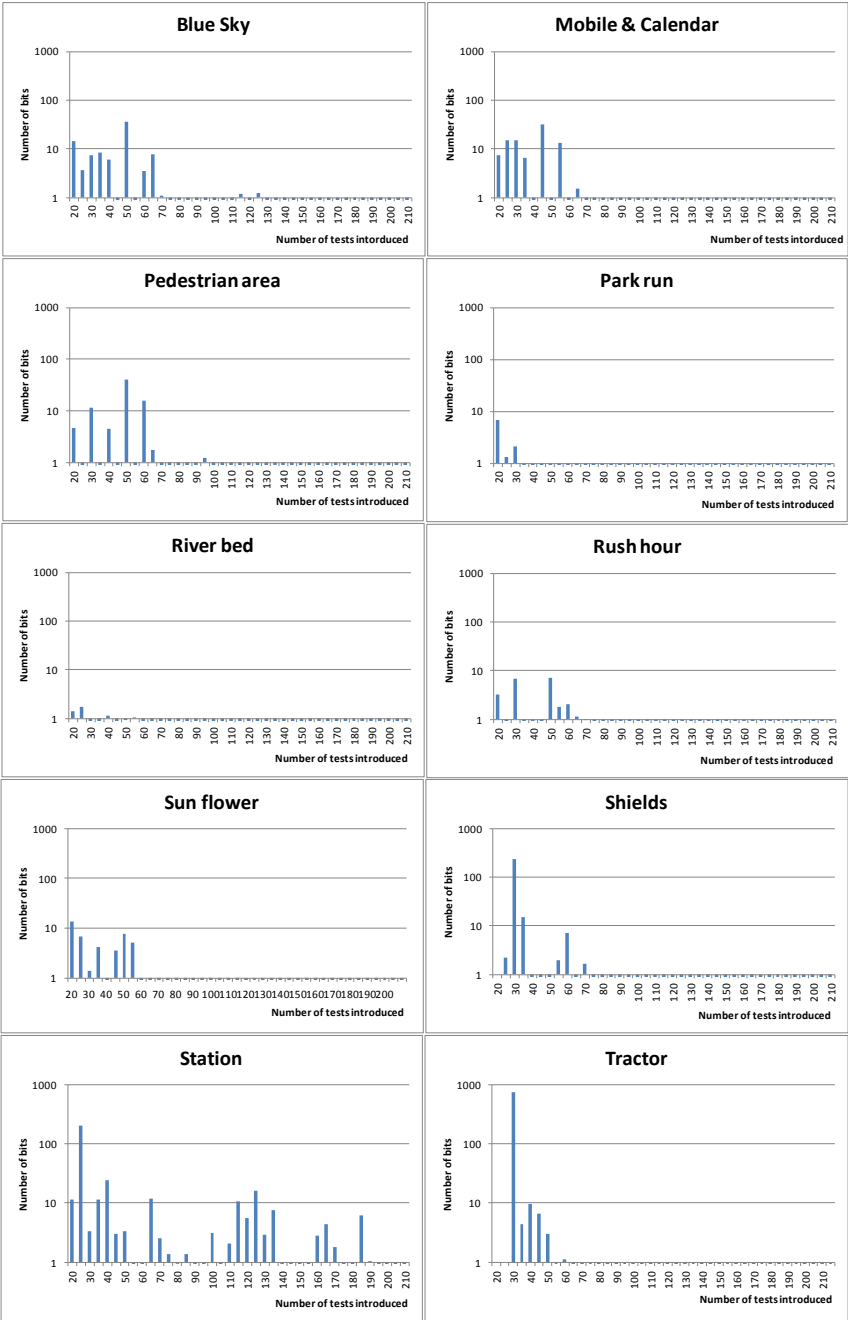


Figure 3.18: Information gain at each step inthe experiment

out them, which reinforces our claims as per Figure 3.15. The accuracy of the predicted psychometric curves is high for all datasets in this range. The RMSE is below 0.3 predicted values. The accuracy in the prediction is generally very high and improves with the introduction of more data, as shown in Figure 3.14. As expected, the Riverbed and Station videos are more difficult to learn due to high noise in the answers, which makes them also more difficult to estimate. Finally we look at the information gain for each of the videos on each step of the experiment in Figure 3.18. The results of the information gain analysis confirm that perturbations in the model were occurring only in the beginning of the experiment. Most of the videos had very small information gain after the 60<sup>th</sup> response had been received. The only exception is the station video, where the model changes even after most of the responses have been received. The results for all the other videos concur with the RMSE values of Figure 3.15. However, the information gain criterion also considers the changes in the standard deviation of the model, which are not captured by the RMSE. This decrease in the variation of the model of the Station video explains why there is still information gain even though the mean RMSE is low.

### 3.4 Conclusions

Video quality is one of the key factors of the overall QoE of video enabled network services. Moreover, due to its intensive footprint on the resources, video quality is also a key aspect for efficient QoE aware management of these services. Subjective VQA is a slow and expensive process, but it is a necessary component of QoE management because it forms the basis for validating any objective quality assessment procedures. In this chapter we have presented the state of the art of subjective VQA and our contribution to the area. Rating is still the prevalent methodology for subjective VQA, however in light of the advantages of difference scaling methods more development in this area is expected. In the following chapters the important role of subjective QoE models is further demonstrated, as they take a key role in management of multimedia services.

# 4

## QoE Management Framework

In this chapter we focus on monitoring and management of delivered QoE in multimedia systems as an enabling factor for QoE-aware multimedia services. Monitoring the QoE, due to its complexity, involves collecting a wide range of available quality performance indicators (QPI) and successfully interpreting these measurements. In this manner, QoE frameworks typically consist of a set of sensors or probes that collect performance data (or QPI) from the system that is fed through QoE models to gauge the quality. The QoE estimations delivered by the frameworks indicate the level of performance by the system, against which the service provider can execute the management strategies. This chapter starts with a discussion of existing QoE frameworks and continues to introduce a QoE management framework for an IPTV service we have developed [110] and all its supporting technologies.

### 4.1 Existing approaches

There are many proposed solutions for managing the QoE of different service platforms. They range from simple monitoring tools that evaluate a restricted set of QPI to complex management systems that correlate many parameters, arbitrate the resources, even adapt the content to the given context.

The authors of [111] have developed an utility function for each of the following network QoS (NQoS) parameters: delay, jitter, packet loss rate and bandwidth of the video stream. They computed the parameters of a generic utility function based on the results of subjective feedback. The authors further claim that managing the multimedia streams with this utility function approach is more effective than using reservation protocols in today's converged network environments. The proposed approach, however,

does not consider the interdependency between the NQoS parameters, but only their effect on QoE independent of each other.

Analysis of quality degradation due to errors during transport is presented in [112]. This methodology focuses on the stream transport statistics to determine the effects of data loss on the video. First, the authors estimate the artifacts in the video due to the transport errors. Then they try to study the visibility of those artifacts and their correlation with the perceived quality. The paper discusses a comprehensive analysis of the error handling schemes of H.264 video codec in order to predict the video artifacts. Finally, it continues to analyze the artifacts from the point of view of magnitude (spatial inconsistency and special extent), special priority (region of interest) and temporal duration. The results show that this approach can sometimes follow the trend of MOS results generated by a subjective study better than the PSNR estimations, but the method is still not sufficiently accurate, occasionally even less than PSNR.

Kim et al. present a framework for support of mobile IPTV streaming service in next generation networks (NGN) [113]. The proposed framework uses multiprotocol label switching (MPLS)-based management of the streams to deliver end-to-end QoS guarantees. Redistribution of the available resources is done by considering how much of them are available, what the terminal capability is, and details from the user profiles. Adapting the requested resources for each stream is included in order to optimize the delivered QoE, using technologies such as scalable video coding as well as context-based content extraction. This type of comprehensive adaptations of the content are resource demanding and hard to implement on a real-time system. However, if NGN with the necessary technologies for QoS guarantees are implemented successfully, the framework can deliver improvements in the utilization of the resources to pursue the desired level of QoE.

A QoE modeling and assurance framework also designed for NGN is presented in [114]. In this framework the QoE assurance is guaranteed by the service controller, which intercepts the communication request in the process of establishment. At the point of interception, the controller predicts the level of QoE to be provided within current resources and context. Finally, it adjusts respective quality-related configurations in order to optimize the QoE, given the available resource. This mediation is implemented through the Internet protocol multimedia subsystem that is part of the NGN delivery system [115]. The QoE model is context-aware and uses a comprehensive set of QPI extracted from various information factories

in the NGN service delivery system. The model incorporates service and content factors, application factors, and transport factors. However, how these objective factors affect the subjective QoE is not evaluated by this model.

Another framework for delivering QoE aware management in NGN is proposed by Zhang and Ansari [116]. The authors recognize many challenges in delivering end-to-end QoE guarantees, starting from the difficulties in measuring subjective QoE to the fluctuations in resources in wireless networks. They suggest modeling QoE with ML methods; however the solution they propose is restricted to individual parameter thresholds. This approach does not consider the interdependences between different parameters and their joint effect on the QoE. The framework includes of a management and a control block, which determines the target QoE and negotiates with the resource admission control function a way to achieve this QoE. Finally, the authors propose a mechanism that implements a global controlled degradation in QoE when the available resources are not sufficient. However, since QoE is highly non-linear, a better approach might be to refuse new service requests, rather than degrading the QoE of all users.

The following sections presents a QoE monitoring framework that we have developed. This framework is extensible to any number of parameters or QPI, and models the QoE based on subjective feedback using ML methods. Finally, it uses a novel method for calculating possible remedies, which allows for improvement of the QoE per active service or globally. As a case study, this framework is applied to a mobile IPTV system.

## 4.2 QoE management for video streaming

This section presents a framework for QoE-aware management of a video streaming service. The framework is particularly designed to work in conjunction with an IPTV service for mobile devices. However, the architecture is generic and compatible with many similar multimedia services and can be easily adapted to work with other multimedia content delivery systems.

### 4.2.1 Architecture of a video streaming systems

The typical video streaming system consists of content servers, transport network and terminal devices (Figure 4.1). The content servers and the network are commonly referred to as content distribution network (CDN).



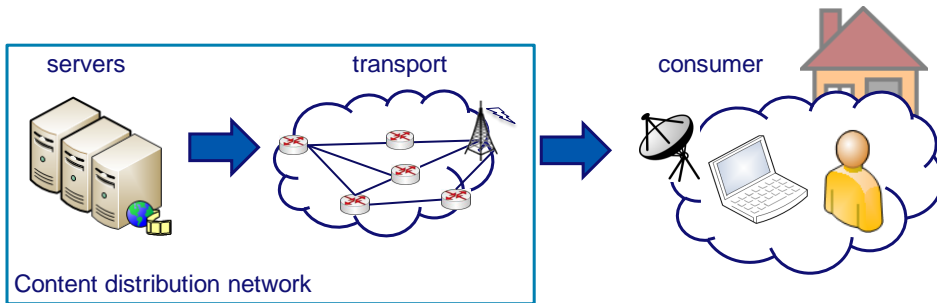


Figure 4.1: Components of a video streaming system

This allows for the content to be distributed in such a manner that good scalability is archived. The content itself, needs to be encoded and compressed prior to distribution. All these processes need to be efficiently managed in order for the service to be successful.

The management of the video encoding process has a significant impact on the overall efficiency of the system. During encoding, a trade-off between the size of the compressed video and its quality is made. In Chapter 3 we elaborated on how the video quality degrades when the coding bit-rate is restricted. On the other hand, one of the major costs in multimedia systems is incurred by the storing and transporting of large amounts of video data. Finally, the cost is not the only factor, video streaming delivery requires accurate and timely delivery. Since the network is a shared and limited resource, large video bit-rates encounter severe hurdles. Consequently, managing the encoding process efficiently involves a difficult trade-off decision between resources and quality. Considering the number of factors involved and their non-linear interdependencies, this task becomes a significant challenge.

Video streaming systems may be implemented using a variety of streaming technologies. The content can be encoded in a single layer, multilayer or in multiple levels for adaptive streaming. The multilayer and adaptive technologies offer more control in the degradation of quality when the network resources are insufficient but require more storage and computing resources. Additionally, as these technologies add complexity to the system, they also add more parameters under the management process. Multilayer video has a number of improvement layers in addition to the base layer, with different levels of bit-rate. Similarly, the adaptive streaming video has more than one parallel streams with different bit-rate levels. These additional features

improve the performance of the system, but also increase the management complexity.

The transport of the video can be implemented also with different technologies. Progressive download transport is implemented over the HTTP [117] protocol, which runs over TCP [118] in the network. In this case, delivery is guaranteed by the network protocol, so no error correction mechanisms are necessary in the application layers. However, the same TCP mechanisms that guarantee the delivery can produce delays that decrease the transport efficiency. This can result in video data arriving late and causes freezes in the playback. For this reason, progressive download is more suitable for playback of stored content when sizable buffering is feasible. The HTTP adaptive streaming [119], offers more flexibility to the client software in case of insufficient transport resources. The client can choose on-the-fly between different levels of bit-rate, according to the performance of the network, while the TCP network protocol guarantees the accuracy of the data.

RTSP/RTP is an application level protocol dedicated to multimedia streaming [120], which can be implemented on either the TCP or the UDP transport level protocols [118, 121]. Removing the TCP transmission control gives more flexibility to the video streaming protocol to implement the control that is more suited for video transmission. Removing the TCP delays allows for less buffering of the video, while other error handling methods can be implemented such as forward error control (FEC) to handle transmission errors [122].

All these mechanisms provide additional flexibility to the system in dealing with errors or lack of resources, but naturally they also add more complexity to the management process.

With so many factors being part of the management and due to the intricate relationship between the resources and the delivered quality, determining the optimal management strategy becomes a problem that is hard to scale. As more content is added, with different characteristics, and similarly new devices become part of the system, the management effort grows very substantially. In the rest of this chapter we present a set of methods and technologies involved in the implementation of our QoE framework, aimed at dealing with this increasing complexity.

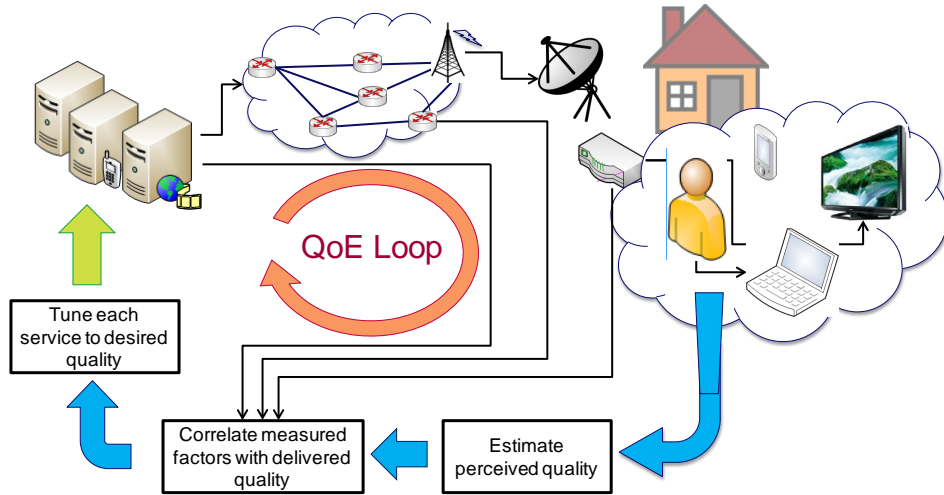


Figure 4.2: The QoE Loop architecture

### 4.2.2 A hybrid QoE management framework

The three high-level components of the video delivery services (Figure 4.1) are traditionally managed independently of each other. Server resources are managed based on utilization statistics. Similarly, network dimensioning is based on its resource utilization. Since both the servers and the transport provide only best-effort reliability, management usually relies on over-provisioning to keep the service quality high. Content encoding is commonly done in a one-size-fits-all fashion to simplify service management.

This approach leads to a sub-optimal utilization of the available resources. Furthermore, due the fact that the components are managed in a disjoint fashion, information from the transport is not used to optimize the server resource and vice versa. Finally, the video content is not adapted to the terminal devices and to the available transport resources.

In order to improve the management process, a closed loop system needs to be implement, as depicted in Figure 4.2. The information from the server, transport and terminal device domain need to be correlated with subjective feedback and sent back to the system. In this way, all available measurements can be efficiently used to optimize the management decisions. Based on this approach, we have designed a QoE monitoring and management framework that works in two phases (Figure 4.3).

The first phase is the training phase. For a number of streams, a range

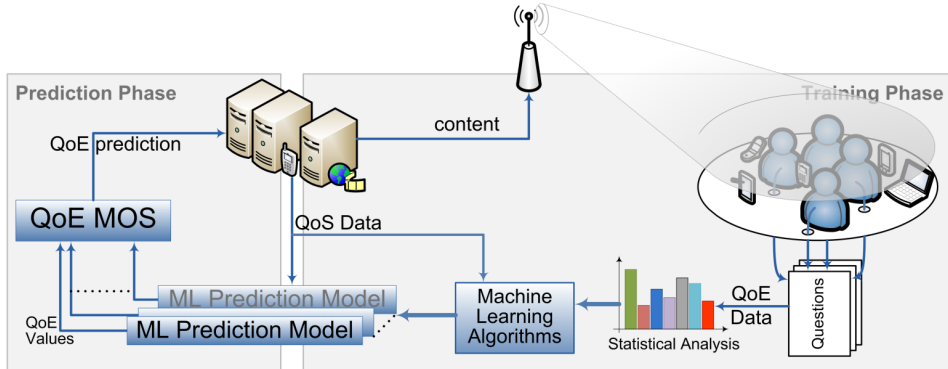


Figure 4.3: System architecture of the QoE Monitoring framework

of all available QPI and system parameters is measured. Typical available metrics are: packet loss rate; frame rate; video and audio bit-rate; spatial and temporal information; Cp and Cm indexes; video encoding quantization level or quality settings for variable bit-rate; audio sampling frequency and quantization precision, screen size; video resolution; number of playback freezes; and average freeze length. In addition to these objectively collected metrics the platform collected subjective feedback from the viewers of those streams. The subjective feedback is used to model the relationship of the QoE to the values of the measured QPI using ML methods.

The second phase is the operational phase. Using the QoE models, the framework estimates the delivered QoE of each subsequent stream. This way the system performance is monitored continuously. The effect of different management decisions can be observed on the delivered QoE. In order to further enhance the decision process the platform calculates 'remedies' or management decisions, that can effectively improve the delivered QoE, for specific streams or globally for the system.

The second phase is also expanded to include an online training capability that works in conjunction to the monitoring and management. The online learning capability offers continuous adaptation and improvements to the models, as soon as new subjective feedback becomes available. In this way the framework does not need to go back to initial phase when the performance of the models degrades.

The details of the implementation of the QoE monitoring and management framework are presented in the rest of this chapter.

### 4.3 QoE management for a Mobile IPTV service

The Mobile IPTV QoE management framework is proof-of-concept implementation of the hybrid QoE management framework described in the previous section.

This framework works in conjunction with a mobile TV streaming service of a commercial telecom provider [110, 123]. The customers can select to watch one of the available multicast video channels with a fixed quality settings. Some of the content is offered on multiple channels with different quality settings or adapted to specific devices.

Many providers find it more efficient to maintain more than one stream with different qualities rather than a single multilayer stream, due to compatibility issues with end user devices.

In the rest of this section we present the details about the objective and subjective data collected. The Algorithms used for modeling the relationships between the two in an offline and online fashion. Finally, a discussion on the computation of the QoE remedies is given, the last aspect of our QoE management framework for mobile IPTV.

#### 4.3.1 Objective Measurements

In order to monitor the service quality, a probe-based network monitoring system is in place, gathering information from the Mobile TV content distribution platform. For each stream the probes collect information, such as type of device, name of channel, stream, duration of the connection; these are captured in an Internet Protocol Detailed Record (IPDR) format [124].

In addition to this information, using mechanisms from RTSP QoS statistics are collected. Some of these values include the number of packets, packet loss ratio for audio, packet loss ratio for video, average delay, maximum delay, and jitter. The full detailed list of parameters is introduced in [123].

In a nutshell, there is a deployed system collecting the AQoS (application QoS) and NQoS (network QoS) data from the system in real-time [125]. The AQoS involves application-level QoS parameters such as video and audio bit-rate and video frame-rate. The NQoS represents the network QoS parameters such as packet loss, jitter, and delay.

The original IPTV platform gives a good overview of the network conditions, providing useful information for dimensioning the resources and managing the parameters of the content encoding. However, it cannot give

any information as to how the service is perceived by the end-user. To acquire the user perception a subjective feedback mechanism was realized as part of this case study, as explained next.

#### 4.3.2 Subjective measurements

In order to understand the subjective experience of the users, we implemented a subjective study. In this study we ask participants to use the service in various conditions and we collect their feedback on the experience. During the subjective studies, the system records the IPDR values for the specific service provided. Then, each participant fills in a questionnaire. These responses are aligned with the IPDR records correspondingly. After this, the measured objective and subjective values are used as training data for the Machine Learning algorithms. These algorithms produce the models that estimate the QoE are used in the following phase. This approach produces one prediction model for each question. The input to each model are the collected system measurements and the output is a predicted answer to one of the questions. The outputs are later combined to produce an overall QoE value. As long as there is no radical change to the environment (e.g. new device or user group) these models are expected to perform accurate predictions of a subset of QoE values.

The QoE prediction models are plugged in the QoE management framework for online use. In this manner the framework continuously evaluates the performance of the system.

### 4.4 Computational inference of QoE models

When we set out to model the performance of a service, we proceed to design a function that computes the performance from the measured service parameters. We can make one such example using the plain telephony service. Since we know that the human voice is in the range of 300Hz to 3400Hz, the service needs a transmission channel of 4KHz to cover this range. Furthermore, any delay below 200 ms is considered acceptable [126]. So a system using pulse code modulation with a sampling rate of 8KHz with less the 200ms delay can be considered to be delivering a service of high quality.

However, when we have to face a system that is of high complexity, and for which we do not fully understand psychophysical perceptual char-

acteristics, developing a model 'directly' can not be done accurately. Furthermore, if the complexity prohibits an exhaustive psychophysical study and the measurements are afflicted with noise, we need to find a different approach to model the performance. Computational intelligence (CI) methods offer a way of modeling complex relationships by observing sample measurements. This is particularly useful in situations where the environment is not fully observable and there is noise in the measurements. With sufficient amount of labeled sample measurements, these methods can be used to produce a model that will map the input data to the labels. In our case the input data is the objectively measurable parameters and the QPI, while the labels are the subjective QoE responses.

This approach also provides for a way to deal with the continuously growing complexity in multimedia systems in a scalable fashion. Since the QoE is not just about QPI thresholds, the interdependencies between the parameters have significant effect on the final outcome. For example, a certain level of video frame rate may be acceptable for 'head and shoulders' type of video, but may not be good enough for an 'action movie'. The resolution of the video and screen size are closely related. Bit-rate, quantization and frame rate are all affecting each other, and so are the spatial and temporal resolution. Finally, the expectations about the service can skew the perception as well. The expectations depend on many factors, such as the service cost or the quality of competitive services. Startup delay and freezes also have an intricate relationship. The more a video is buffered at the beginning, the smaller is the chance for freeze during playback. Many interdependencies between the parameters exists. As they are highly nonlinear, modeling them becomes challenging with traditional statistical inference methods; so adopting modern CI methods is necessary. The following sections details CI algorithms that are suited for this challenge and that were particularly applied in our mobile IPTV QoE management framework.

#### 4.4.1 Supervised learning background

Supervised learning algorithms build models based on training data or training examples [127]. Formally, the training set is given in the form of  $X = \{\bar{x}^t, y^t\}_{t=0}^N$ , where  $\bar{x}^t$  is an input vector of attributes and  $y^t$  is a class or a label of the  $t^{th}$  datapoint in a dataset of size  $N$ . For training QoE models, the attributes  $\bar{x}$  can be a set of QoS attributes such as video bit-rate, frame rate, and audio bit-rate, while the label is the value of the

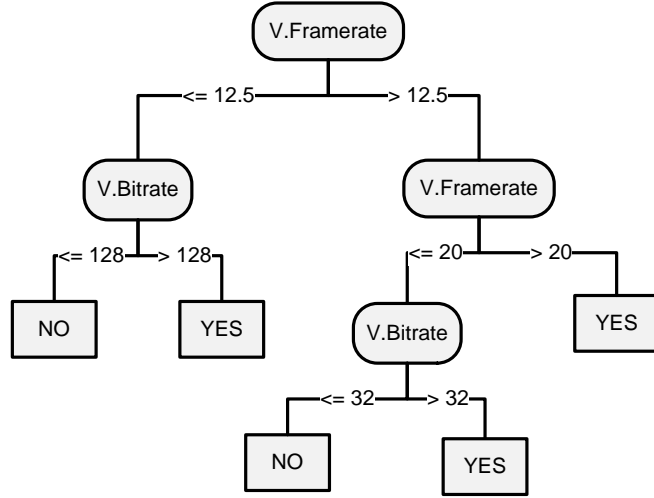


Figure 4.4: Decision Tree

QoE (good, fair, bad). The goal of the procedure is to derive a hypothesis  $h$  about the input data such that  $y = h(\bar{x})$ . This hypothesis represents our prediction model and can have many forms, such as decision tree, artificial neural net, or a support vector machine.

### Decision Trees

Decision Trees (DT) are models represented by a hierarchical tree structure where each branching node represents a test (or a question) and each leaf is associated with one possible decision (class or a label) (Figure 4.4). ML induction tree algorithms produce DT models from training data in a supervised learning fashion. The DT model, can be used for the classification of unlabeled datapoints. The datapoint values are tested at each branch starting from the root of the tree. The tested datapoint satisfies tests on the route leading only up to a single leaf of the tree. The class (or label) associated to that leaf is the classification output for that datapoint.

The basic concepts for induction of DT are captured by the ID3 algorithm presented in the seminal work by Quinlan [128]. The DT is built by adding branching nodes, starting from the root of the tree. The tree is finalized by adding the edges or the leaves. The tree is built in a recursive fashion until all the branches finish in leaves.

The tests in the branches are expressed as  $a_k = v_l$ , where  $a$  is one of



the  $k$  attributes of the dataset and  $v$  is one of the  $l$  possible values of this attribute. If the attribute is of continual nature (real number) than the test is of the form  $a_k \geq v$ . The attribute  $a$  and the value  $v$  are selected such that the two subsets (for binary trees) that this test splits the training data into are with minimum entropy in regards to their label. In other words, the dataset is split in such a way that the probability of each datapoint to belong to a single class in each of the subsets is maximized. This is achieved by calculating the information gain for each test.

The information gain  $G(S, a)$  of splitting the set  $S$  over the attribute  $a$  is given in equation 4.1.

$$G(S, a) = E(S) - \sum_{i=1}^m f_S(a_i) E(S_{a_i}) \quad (4.1)$$

$$E(S) = - \sum_{j=1}^N f_S(j) \log_2 f_S(j) \quad (4.2)$$

$E(S)$  is the entropy of the set  $S$  calculated as in equation 4.2 and  $f_S(j)$  is the proportion of datapoints in set  $S$  belonging to class  $j$ .

Selecting first the tests that split the data into more uniform sets results in a shorter tree that generalizes well. When the subsets only contain datapoints of a single label, the branching is stopped and a leaf associated with that label is attached to that branch.

Using the same principles, the C4.5 algorithm is developed as an extension of ID3 [129]. This algorithm overcomes many weaknesses of ID3, such as handling continuous attributes, training data with missing values, and many different pruning enhancements that deal with overfitting.

We have used the DT methods to build models mapping the many measured parameters of the multimedia delivery system to the subjective quality feedback collected from the users. DT models are easy to use and compute the class of an unlabeled datapoint very efficiently. Furthermore, they represent the model in an intelligible form, which is readable by humans. This is useful for our purpose, since the network operators can derive conclusions about the interdependencies between specific network QoS parameters and the expected QoE.

### Support Vector Machine

The support vector machine (SVM) is a functional type algorithm and works by first plotting the data in an  $n$ -dimensional space ( $n$  being the number of attributes). In case of nominal (or discrete) attributes, the algorithm creates another axis for each value of the nominal attribute. One such example is the type of the terminal device. For this attribute the algorithm creates one variable for each possible value. These variables take Boolean values (0 or 1) depending the presence of that particular nominal value.

After plotting the data in the  $n$ -dimensional space, the SVM algorithm builds a hyperplane that separates the data in an optimal manner [130] in regards to the two classes (labels). If there are more classes, the SVM generates one hyperplane for each combination of pair of classes. Substituting the values of the attributes, a particular data point can be placed above or below the hyperplane; that is, it belongs to one or the other class. The particular implementation of SVM used here is called Sequential Minimal Optimization [131].

The SVM is an advanced ML algorithm that in many cases outperforms the DT and other algorithms. However, the drawback in using SVM is that is more complex and requires more computing power.

### Ensemble methods

Different ML algorithms have different strengths and weaknesses. No one single algorithm is best suited for capturing all various relationships between the data and the labels. Furthermore, training models using the same algorithm on slightly different datasets can lead to different performance in the models because of the different amount of noise they encountered in the data.

Ensemble methods use a technique for combining multiple learners into a group and utilizing their differences to improve the performance of a single classifier. These methods were developed in an attempt to turn 'weak' classifiers (classifiers that perform slightly better than random) into stronger ones [132].

Ensemble methods deploy multiple classifiers trained with different strategies, which combine their predictions into a more accurate group prediction. Their strength is also in improving the generalization capabilities of the standalone classifier [133].

Bagging is an ensemble method of bootstrap aggregation according to which one base classifier of the ensemble is trained on the whole dataset  $D$  and the remaining classifiers on a sub sample of  $D$  - sampled uniformly with replacement. For classification, the bagging ensemble uses equal weight voting of all the classifiers to output the class of the datapoint.

Boosting is another ensemble learning method. An example to boosting is the AdaBoost algorithm [134]. AdaBoost generates a sequence of base models  $h_1, h_2, \dots, h_k$  using weighted training sets (weighted by  $D_1, D_2, \dots, D_N$ , where  $N$  is the size of the dataset). To train the first model  $h_1$ , the weights are initialized to the values of  $1/N$ . To train the consecutive models, the algorithms adapt the weights in such a manner that the training is focused more on the datapoints that were misclassified by the previous classifier. The weights of the datapoints that were classified correctly are multiplied by a coefficient  $\beta$  ( $\beta < 1$ ), which may be calculated in different ways depending on the different implementations of AdaBoost. The misclassified samples weights remain unmodified. Finally, the weights are normalized so that they resemble a probability distribution. Models that misclassify more than half of the datapoints are removed from the ensemble.

In our QoE framework, a specific combination of classifiers and ensembles were used to optimize the performance of the system, as discussed next.

#### 4.4.2 QoE models for mobile IPTV

Here we present two sets of models for QoE evaluation. Models developed as precursors in the lab and models developed for the commercial IPTV framework case study, respectively.

##### Subjective QoE models developed in the lab

To validate our ML approach, subjective QoE models were initially built in the lab on typical parameters of video streaming services, such as video bit-rate and frame-rate and audio bit-rate. In this study [135] the subjective evaluation is implemented using the 'Method of Limits' [92]. Thus is used to detect the thresholds by changing a single stimulus in successive, discrete steps. A series terminates when the intensity of the stimulus becomes detectable, as described in Chapter 3. For the particular case, we record the segment when the customer has decided that the multimedia quality is unacceptable. The purpose is to determine the user thresholds of

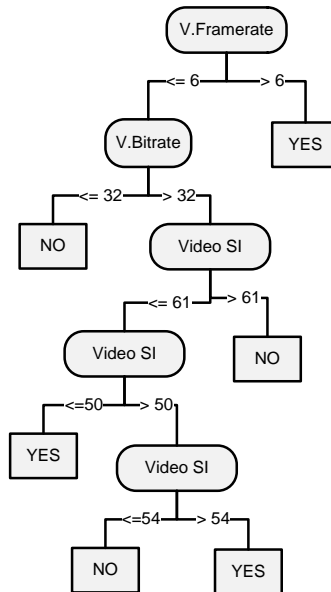


Figure 4.5: Decision Tree for the Mobile dataset

acceptability in relation to QoS parameters, taking into account the type of content and terminal. You can see an example of one test for the Mobile terminal in Table 4.1. The user was satisfied with the quality while the video bit-rate was at or above 96Kbit/s. This example generates eight data-points of which six are with a class label of 'Yes' (satisfactory QoE) and two with 'No' (unsatisfactory QoE).

The same tests are performed on different users showing them different video content as well as repeating the tests on three different terminals: mobile, laptop and personal digital assistant (PDA). After compiling the results into three sets for each type of terminal, we used the sets as training data for building prediction models. The J48 algorithm, an implementation of C4.5 [129] in the Weka platform [136], and SMO [131], an implementation of a Support Vector Machine, were used to build the models. The resulting prediction models are shown in Figures 4.5 to 4.10.

The ML models are evaluated for prediction accuracy on a test dataset. The test dataset is usually part of the available labelled data that is reserved for testing and is not used for training. The goal of this estimation is to evaluate how well the model generalizes the concepts found in the data, which could not be accurately accomplished if the same data were used for

Table 4.1: Example of a series of tests in the subjective study

Segment Time (seconds)	Video bitrate (kbit/s)	Audio bitrate (kbit/s)	Frame rate	QoE
1 (1-20)	384	12.2	25	Yes
2 (21-40)	303	12.2	25	Yes
3 (41-60)	243	12.2	20	Yes
4 (61-80)	194	12.2	15	Yes
5 (81-100)	128	12.2	12.5	Yes
6(101-120)	96	12.2	10	Yes
7(121-140)	64	12.2	6	No
8(141-160)	32	12.2	6	No

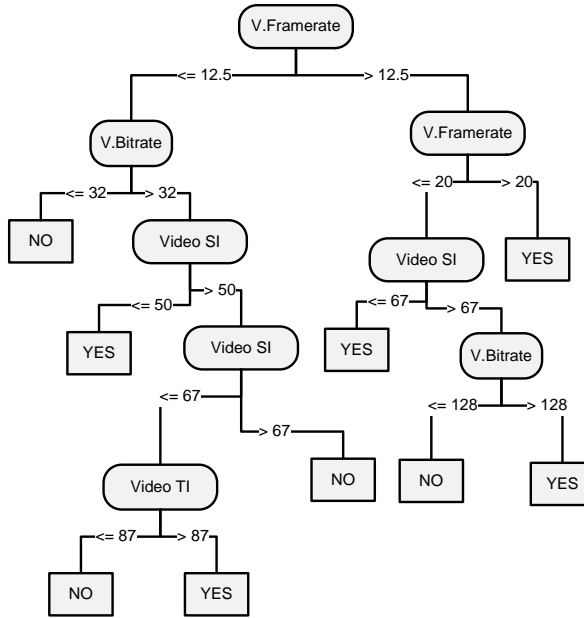


Figure 4.6: Decision Tree for the PDA dataset

training and testing.

Test datasets can be generated by reserving approximately 30% of the available training data for testing. However, when the training data set is of limited size, which is common for subjective data, carving out 30% of the data for testing is a significant restriction for the training process. In such cases the model evaluation can be implemented using a cross-validation methodology [137]. The 10-fold cross-validation method splits the dataset on 10 equal size sets. Then a model is trained on 9 of the sets and the 10th is used for testing. The procedure is repeated 10 times for each combination of 9 training sets and the one test set. The average performance of the models on the test sets is considered as the performance of the model trained on the whole dataset. The results of the 10-fold crossvalidation for our models are given in Figure 4.11.

The results from Figure 4.11 demonstrate that the ML models can be efficiently used for estimating subjective quality.

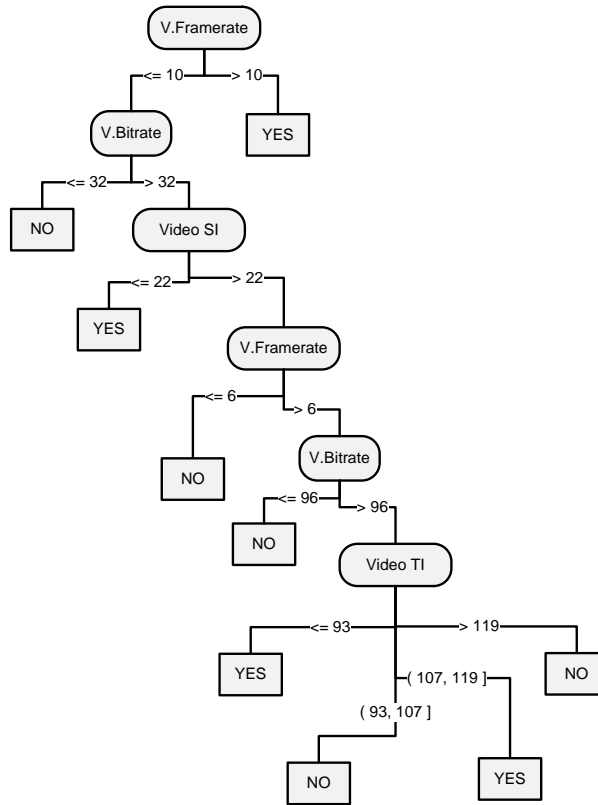


Figure 4.7: Decision Tree for the PDA dataset

```

1.4555 * (normalized) Video SI
+ 1.0459 * (normalized) Video TI
- 5.0892 * (normalized) Video Bitrate
- 3.7632 * (normalized) Video Framerate
- 0.4582

```

Figure 4.8: SVM hyperplane for the Mobile dataset

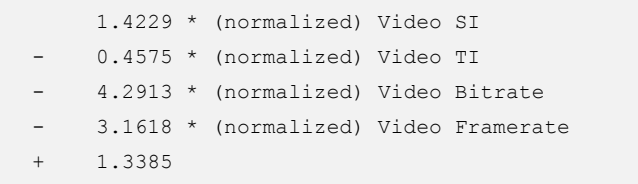


Figure 4.9: SVM hyperplane for the PDA dataset

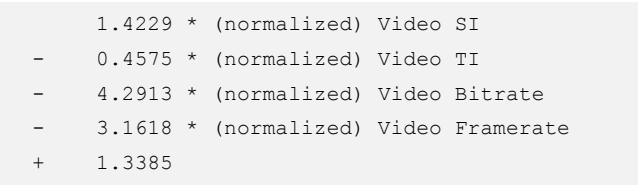


Figure 4.10: SVM hyperplane for the Laptop dataset

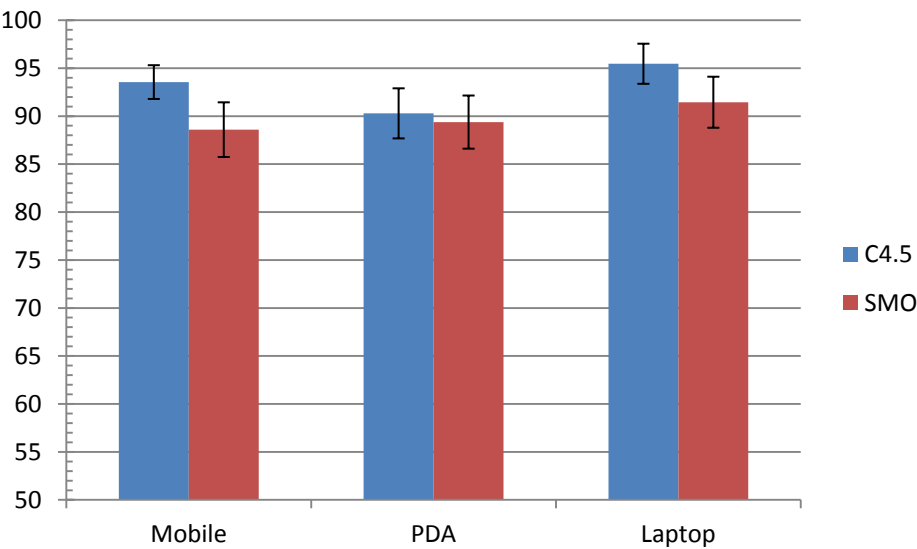
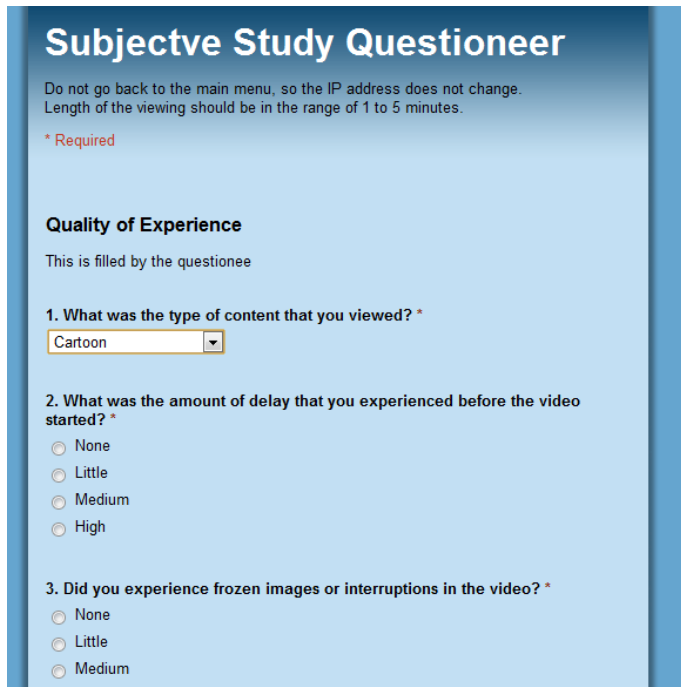


Figure 4.11: Performance of the DT and SVM models on the three datasets





The screenshot shows a web-based questionnaire titled "Subjective Study Questionnaire". At the top, there is a blue header with the title in white. Below the header, a light blue background contains the following text: "Do not go back to the main menu, so the IP address does not change. Length of the viewing should be in the range of 1 to 5 minutes." followed by a red asterisk and the word "Required". The section is titled "Quality of Experience" and includes a note "This is filled by the questionee". There are three numbered questions: 1. "What was the type of content that you viewed? \*" with a dropdown menu showing "Cartoon"; 2. "What was the amount of delay that you experienced before the video started? \*" with radio button options for "None", "Little", "Medium", and "High"; 3. "Did you experience frozen images or interruptions in the video? \*" with radio button options for "None", "Little", and "Medium".

Figure 4.12: Subjective study questionnaire

### Subjective QoE models for mobile IPTV

Using the demonstrated approach we proceed to model the data collected by the QoE framework.

The data for this study is collected by the viewers after watching each of the available videos. The collection is implemented using a Web based interface (Figure 4.12).

The complete list of questions presented to the participants is the following:

1. What was the type of content that you viewed?
2. What was the amount of delay that you experienced before the video started?
3. Did you experience frozen images or interruptions in the video?
4. Did you experience interruptions in the audio?

5. How much pixelation (big blocks of color) did you experience?
6. Did you experience noise or distortions in the audio?
7. Did you experience problems with the synchronization of the audio and video?
8. How did you find quality of colors?
9. How did you find definition (sharpness) of the video?
10. What was your overall perception of the quality?

For the first question the participants could chose from 7 possible answers for the types of content give bellow:

1. News
2. Music Videos
3. Entertainment
4. Documentary
5. Movie or TV Series
6. Cartoon
7. Sports

In questions 2 though 7 the participants are asked to respond between one of the 4 given values bellow:

1. None
2. Little
3. Medium
4. High

In questions 8 and 9 the response is between the following 4 values:

1. Excellent

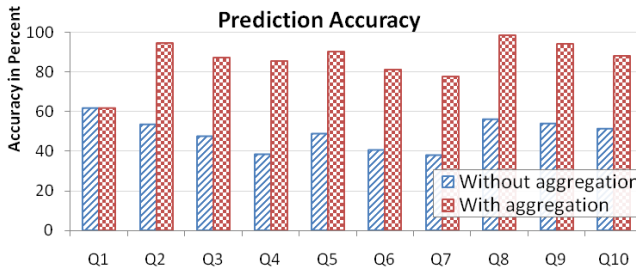


Figure 4.13: Prediction accuracy with and without output aggregation

2. Acceptable
3. Poor
4. Unacceptable

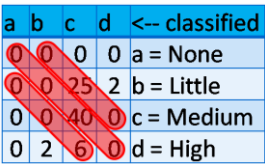
The final 10th question offered a choice of 5 distinct values:

1. Excellent
2. Very good
3. Not so good
4. Very bad

The collected subjective dataset contains 55 features from the IPDR log files and the 10 subjective responses from the viewers. For each of the 10 subjective responses a separate model is trained. Finally a combination of the output of the 10 models is used to predict the overall QoE.

Each of the models is trained using C4.5 and SVM as base learners of an AdaBoost ensemble. The performance of the QoE models is given in Figure 4.13.

The models were trained using categorical labels, which make them easy for humans to read and understand, for example "Excellent" or "Not Good". But from the prediction point of view these labels are not ordered, they are considered the same as we would consider the labels "Red" or "Blue". So when we are calculating the prediction accuracy, only the exact predictions are taken into account as accurate. We do not know how many near misses we have. Most of these near misses would provide for good



a	b	c	d	<-- classified
0	0	0	0	a = None
0	0	25	2	b = Little
0	0	40	0	c = Medium
0	2	6	0	d = High

Figure 4.14: Confusion Matrix for high accuracy with tolerance  $\pm 1$

management input. For instance, a prediction of "Very Bad" and "Not Good" might lead to the same management decision since both cases are not satisfactory. If we take this into account and tolerate a small error rate for the output, such as errors with a distance of one or less from the actual value, the accuracy of the models significantly increases. For graphical representation we can look at the confusion matrix in Figure 4.14; the main diagonal represents the accurate cases (actual value row and predicted value column). If we add the values in the two adjacent diagonals, we can get the new accuracy with tolerance of  $\pm 1$  and thus get a higher effective accuracy of our classifier (Figure 4.13).

4.5 On-line inference of QoE models

Supervised learning models give us the possibility to model the QoE from subjective data. However, multimedia systems are evolving rapidly, with the introduction of new services and new devices. The models trained on the pre-determined conditions become less accurate as conditions change. Retraining the models with new subjective studies regularly is costly and inefficient. Instead we propose the use of an online learning approach, where subjective feedback is received continuously and is used for updating the models with the new data. In order for this method to be incorporated, we need to have a mechanism for continuous collection of subjective feedback and new set of methods for inference of models that work in on-line fashion.

The rest of this section introduces the on-line supervised learning methods used in our QoE management framework.

4.5.1 Online supervised learning background

The online learning algorithms are ML algorithms designed to train models in a supervised learning fashion from labelled data. The only difference

with offline or batch learning is that the data is processed sequentially and continuously. The main motivation for online learning is modelling fast data streams without having to retain a sizable amount of data. In order to achieve this, the algorithms need to be able to add new concepts to their models online, i.e. as new data becomes available. Moreover, they also need to be able to 'forget' or remove concepts from the models that are not present in the incoming data.

Changes in perceived QoE commonly occur when new types of content or new services are introduced, but also when the user's expectations increase with the advances in technology. This kind of change can be quite frequent in multimedia services. To circumvent the need to redo the subjective studies and recreate the models, we included online learning technology in the framework.

In the following sections we detail the online learning methods used in the framework, providing an analysis of their performance.

### Hoeffding trees

Hoeffding trees (HT) [138] is an algorithm for decision tree induction that is designed to handle extremely large training sets delivered by fast data streams. The training set is commonly so large that it is not expected for the training data to remain in memory. It is actually processed from the input stream in a single pass. The fact that the data is processed sequentially, or one data-point at the time, characterizes this approach as online learning.

At any point in time, the learner has only a partial view of the data, since the rest of the dataset has not yet been introduced. This means that the selected attribute for the test in a node cannot be made with full confidence for any split criteria, but it has to be made with a more relaxed one. The algorithm selects the best attribute at a given node, by considering only a small subset of the training examples that satisfy all the tests leading to that node. As the data is being introduced, the first datapoints are used to choose the root test. Once the root attribute is selected, the succeeding examples will be passed down to the corresponding leaves and used to choose the appropriate attributes for the tests in the new branching nodes that replace existing leaves, and so on. The number of examples that justify a branching at each node is made by relying on a statistical result known as Hoeffding bound. Given  $n$  observations of a random variable  $r$  with a range  $R$ , the calculated mean of  $r$  is  $\hat{r}$ . The Hoeffding bound states with

probability  $1 - \delta$  that the true mean of the variable is  $\hat{r} - \epsilon$  whereby  $\epsilon$  is given in equation 4.3

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}} \quad (4.3)$$

Defining the attribute selection criterion as  $G(a)$ , then  $\Delta G = G(a_1) - G(a_2) > 0$ , assuming that the  $a_1$  attribute is more favourable (or with larger information gain) than  $a_2$ . Given the desired  $\epsilon$ , the Hoeffding bound guarantees that  $a_1$  is the better selection with probability  $\delta$  if  $n$  examples are seen, where  $\Delta G > \epsilon^2$ .

In addition to the relaxed information gain criteria for generating the branches, the HT algorithm also implements mechanisms for pruning existing branches. At each branch, attribute selection is also being tested against a 'dummy' test on the attribute  $a_0$  that substitutes the branch with a leaf. If this test results in better gain for the sufficient number of tests given by the Hoeffding bound, then this branch is pruned.

The resulting tree constantly adapts as new data is being introduced, capturing more and more relationships between the attribute values and the labels. Moreover, it also deletes relationships that disappear from the newly introduced datapoints as new trends in the data appear. This makes this method applicable for building QoE models based on continuously collected subjective feedback.

Two further incremental improvements to the HT algorithm that are used in the application to our framework are explained next.

#### *Hoeffding option trees*

Option trees generalize the regular decision trees by adding a new type of node, an option node [139]. Option nodes allow several branchings instead of a single branching per node. This effectively means that multiple paths are followed below the option node and classification is commonly done by a majority-voting scheme of the different paths. Option Decision Trees can reduce the error rate of Decision Trees by combining multiple models and combining predictions while still maintaining a single compact classifier. For the proposed methodology, the combination of the predictions of different paths is done with weighted voting [140], which sums up individual probability predictions of each leaf.

#### *Hoeffding Option Trees with functional leaves*

The usual way a decision tree is built by assigning a fixed class to each leaf during the training. This class is equal to the class of the majority of

the training datapoints that reach this node. There is another approach based on which the leaves are not associated with a fixed class but are simple classifiers themselves (referred to as functional leaves). These classifiers are trained only on the data that falls on the leaf during the training of the whole tree. This approach can outperform both a standalone decision tree as well as a standalone classifier [141]. In further research on functional leaves the authors of [142] show that, for incremental learning models, naïve Bayes classifiers used as functional leaves improve the accuracy over the majority class approach. However, this cannot be a rule of thumb. There are exceptional cases shown in [143], where a standard Hoeffding Option Tree will outperform the tree with functional nodes. The author of [143] proposes an adaptive approach, where the training algorithm adaptively decides to use the functional or majority votes, based on the current performance of each of them. This implementation is adopted in our framework to enable efficient online learning.

### Online ensemble methods

Earlier in this chapter we have discussed the benefits of using ensemble methods for improving the performance of supervised learning algorithms. However, modifications for the ensemble techniques are also necessary to enable the online mode of operation.

First of all, the online ensemble methods need to have online algorithms as base learners, so that they can update their base models as new data arrives. However, in batch or offline learning the ensemble algorithm has the freedom to split the set in different subsets or change the dataset weight distribution for each learner. On the other hand, in online learning the data arrives one at the time, so the online ensemble algorithms need to adapt their strategy accordingly.

In online bagging [144, 145], instead of resampling the data with replacement as in offline bagging, the algorithm presents the datapoint  $(\bar{x}, y)$  to each learner multiple times. The number of presentations of a datapoint to a base learner is  $K$  times, where  $K$  is sampled from a *Poisson*(1) distribution. The authors of [145] claim that the online bagging classifier converge to the normal bagging classifier performance as the number of training examples tend grow to infinity.

Online boosting [144] is designed to be an online version of AdaBoost [132] algorithm. As we described previously AdaBoost generates a sequence of base models  $h_1, h_2, \dots, h_k$  such that each consecutive model's training is

focused on the datapoints misclassified by previous models. The online version of this algorithm repeats the presentation of the datapoints similarly to the online version of bagging. The only difference is that the number of presentations  $K$  to a based learner  $t$  is sampled from a distribution  $Poisson(\lambda)$ , where the parameter  $\lambda$  is increased if the previous model ( $h_{t-1}$ ) misclassified the datapoint or decreased if  $h_{t-1}$  classified the datapoint correctly.

### 4.5.2 Subjective QoE models with continuous learning

To evaluate the online learning approach, we implement a set of experiments where subjective data is introduced to the learners sequentially in an online fashion and their performance is evaluated. The subjective data is collected with the method of limits [74] and have two labels, 'acceptable' and 'not acceptable'. We consider different kinds of streamed content on three different terminals, as described in the previous chapters.

For the implementation of the online learning algorithms we used the Massive Online Analysis (MOA) [146] ML platform for data stream mining, which has implementations of Hoeffding Option Tees and Oza Bagging algorithm. The MOA platform is an extension of the WEKA ML data mining platform [136]. In our case, the viewer feedback is considered as scarce and expensive, so we can only expect small amount of data arriving. In light of this issue we have modified some of the parameters of the algorithm to serve our purpose, mainly the  $n_{min}$  grace period from 200 (default value) to 1. It is not meaningful to wait for 200 datapoints until we start building the DT when we only have 3500 datapoints available.

An adaptation of the model evaluation procedure is also necessary. In MOA there is the assumption of abundance of data, and the estimation of the accuracy of the prediction models is done by interleaving testing and training. In this way there is part of the data that is dedicated for testing, and this data is not used for training of the models. Consequently, the accuracy of these models could be reported as lower in cases of small amount of available data. The approach for model evaluation in cases of scarce data is cross-validation as described in the previous section.

We implemented a 10-fold cross validation scheme to calculate the accuracy of the classifier. This validation scheme splits the data into 90% for training and 10% for testing, and then it repeats this process ten times. Each time different combination of datapoints is used for training and testing.



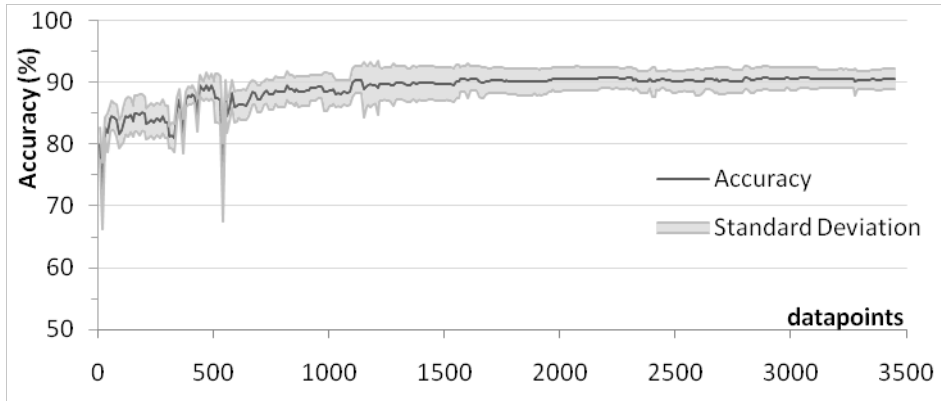


Figure 4.15: Hoeffding Option Tree results

The results of the execution of the Online Learning using the Hoeffding Option Tree algorithm that the classification accuracy rises fast to over 80% with fewer than 100 datapoints, i.e. user-generated feedback instances. After around 1000 datapoints the classifiers converges to its accuracy of approximately 90% (Figure 4.15). In the same manner the standard deviation of the accuracy falls quickly to below 3% (with just a few exceptions) and then falls to around 2% after introducing 1600 points.

We obtained qualitatively similar findings from the execution of the ensemble Online Learning algorithm Oza bagging Hoeffding Option Tree (Figure 4.16). We can see that both algorithms reach very high prediction accuracy (90%) very rapidly (order of a thousand of datapoints). As expected, the ensemble approach gains accuracy faster. Furthermore the classifier's standard deviation of accuracy over the different folds of the cross-validation is much lower than in the stand-alone classifier.

Overall, we have presented results that show that we can already achieve an accuracy of over 80% by learning only 100 datapoints which are randomly selected feedback.

In order to evaluate the overall benefit of using online learning to handle changes in the environment, we developed the experiment for testing the concept drift. Changes in trends of the labelled data are referred to as concept drift. Classifiers need to adapt to concept drift by modifying or deleting existing concept in the models to accommodate the new patterns in the data. The test is implemented by training the classifier on one type of data from the subjective study and then introducing a new type. At

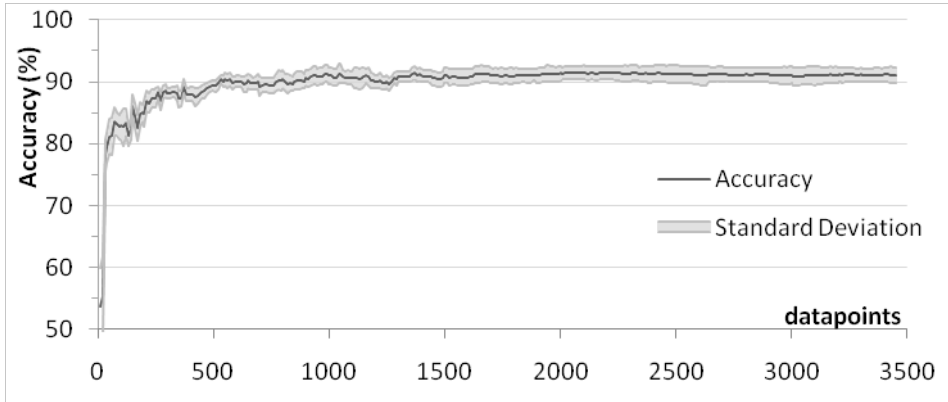


Figure 4.16: OzaBagging Hoeffding Option Tree results

the moment when the new data is introduced the model is not aware of the change and predicts based on knowledge from the previous data. Then new data is introduced from the second dataset. The algorithm updates the model according to the introduced data.

In our algorithm stack, different algorithms behave differently. The Hoeffding Option Tree discards some nodes and induces new ones. The weights on the different paths of the option tree might be modified, and the online ensemble classifier can decide to update the weights to the individual classifiers if their accuracy decreases. With the proposed experimental setup we can monitor the introduction of new concepts in the dataset and the speed of adaptation to the model. The results of the concept drift experiment are presented in Figure 4.17 for a single HOT classifier and in Figure 4.18 for an OzaBagging ensemble of HOT classifiers.

The first dataset contains only video with Temporal Information smaller than 110, which includes about 60% (2010 out of 3370) of the data. The second set contains the remaining 40% of data. This are samples with Temporal Information higher than 110, typically content with higher dynamics. The result from using the Hoeffding Option Tree algorithm shows drop in the accuracy and increases in the standard deviation at the moment when the new dataset is introduced. However, the accuracy is recovered very fast and converges to above 90% in fewer than 200 datapoints. This is a very encouraging result that shows the capabilities of this algorithm to adapt to changes. In this experiment the model was trained on content with small TI (slow changing content) and then we introduced high changing content

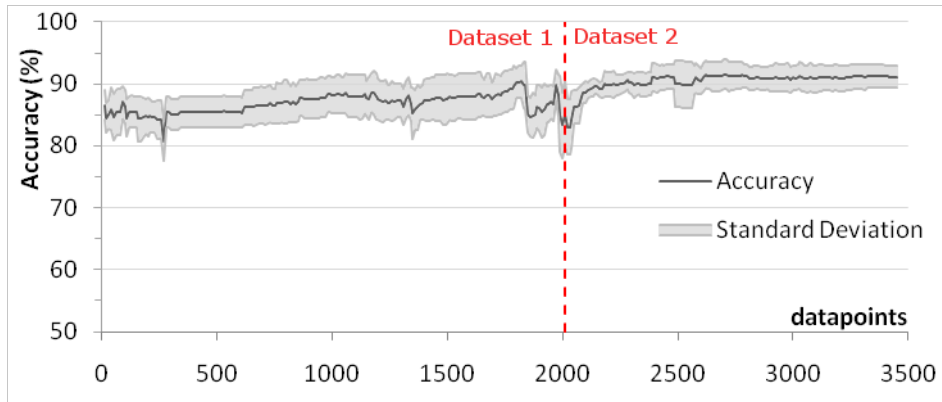


Figure 4.17: Results of the Hoeffding Option Tree with concept drift experiment

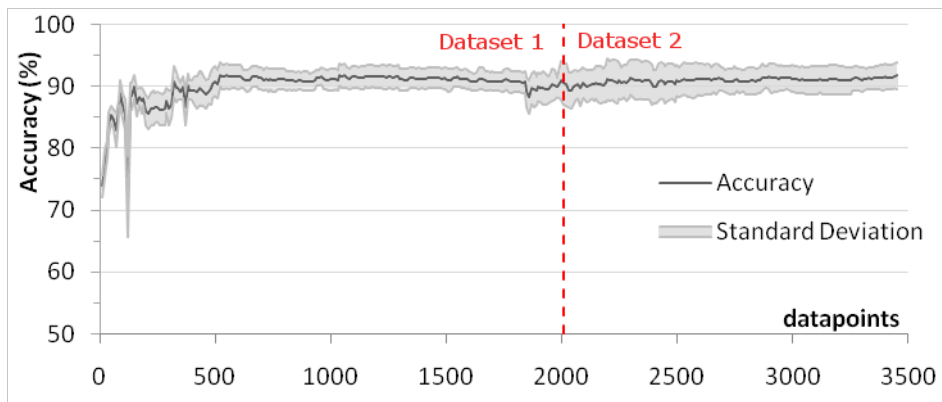


Figure 4.18: OzaBagging Hoeffding Option Tree with concept drift results

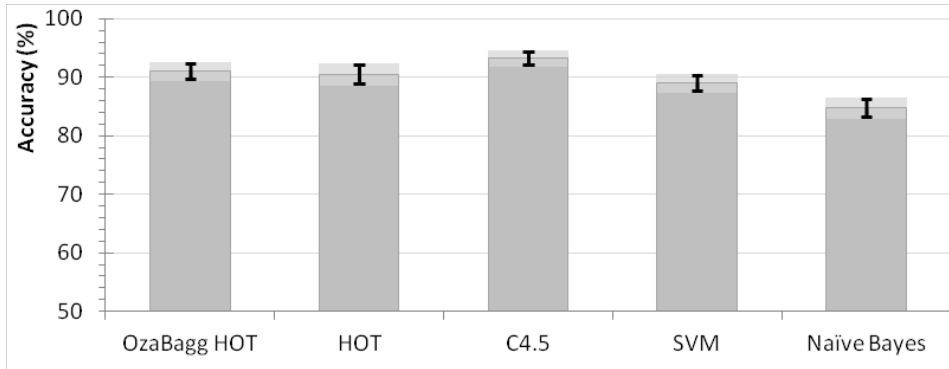


Figure 4.19: Comparison with standard ML algorithms

(high TI). Even with this rather drastic change the accuracy recovered very fast.

The results obtained with the Oza Bagging Hoeffding Option Tree ensemble are even better (Figure 4.18). This algorithm is much more robust to changes and deals with the concept drift with close-to-none loss in accuracy and limited rise in the standard deviation. This result shows the tremendous value of online learning, the robustness of the ensemble approach and justifies the added complexity in using an ensemble versus a standalone classifier.

To demonstrate the statistical significance and viability of our approach, Figure 4.19 illustrates how Hoeffding Option Tree (HOT) and Oza Bagging HOT (OzaBagg HOT) compare with 3 standard ML approaches, namely Naïve Bayes, Support Vector Machine (SVM) and C4.5. Since the offline algorithms have the advantage to learn on the whole dataset, the performance of the online algorithms is expected as good as the offline in the best case. It is evident, that C4.5 performs best with 93% accuracy but the online learning algorithms we used follow closely behind with 90.5% and 91.1%.

We demonstrate the usefulness of this approach by testing it on data that was previously derived via conventional subjective studies. The QoE prediction models show high accuracy and high adaptability to concept drift in the dataset. The fact that the accuracy of the online learning algorithms are approaching the accuracy of the standard batch ML algorithms (of above 90%) demonstrates the applicability of the approach. Unfortunately, due to project constraints and limited access to the commercial

IPTV platform, we could not evaluate our online learning system on a real deployment.

## 4.6 QoE Remedies

In addition to accurate measurement of QoE, efficient management of multimedia services also necessitates efficient provisioning of resources towards the delivered quality. However, as complex relationships exist between the parameters that govern these resources and the delivered quality, selecting the appropriate values is not trivial.

When the measured QoE is unsatisfactory, how do we determine the optimal way to improve it? On the other hand, when the delivered QoE is high, are we over-provisioning certain resources? Can we achieve the same QoE with fewer resources and utilize them more efficiently?

To answer these questions, we need to expand the functionalities of the QoE management framework to include mechanisms that allow for calculating the needed changes, which will provide the targeted QoE. We define these mechanisms as 'QoE remedies' and we proceed to describe them in the rest of this section.

### 4.6.1 Estimating the QoE remedies

The QoE remedies are changes to parameters under management that need to be applied to a specific instance of the service so that its QoE is improved. These changes (or distance in values) can be on a single parameter, such as the bit-rate of the video. Or they can be on a combination of multiple parameters, such as frame-rate and the resolution of the video. The remedies can also refer to measurements such as the incurred packet loss. However, decreasing the packet loss may not be directly under the control of a single parameter and may include the utilization of multiple resources. For example, the level of packet loss can be affected by increasing the network resources or by using more advanced transmission data protection schemes. In any case, the QoE remedies deliver a deeper level of understanding of the correlation between the system resources and the resulting service QoE. This level of understanding enables more informed management of the service and alleviates the need for a trial and error approach. Furthermore, if the suggested remedies can be associated with costs, the management system can autonomously select the best option. In this manner a high

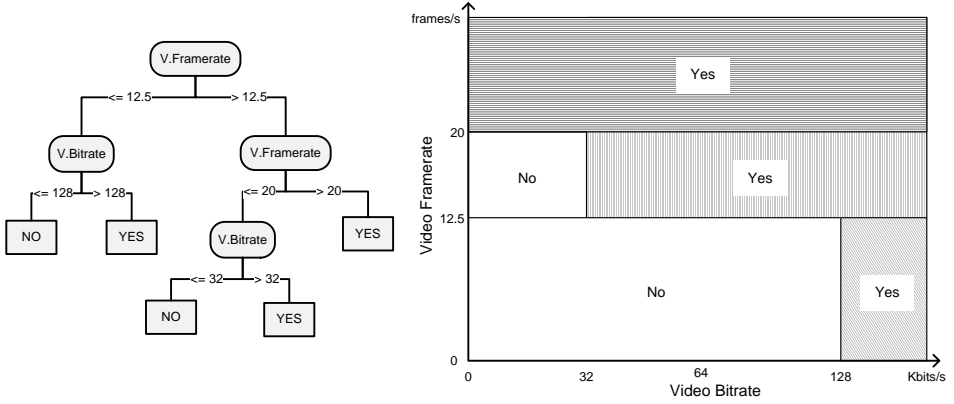


Figure 4.20: Simple decision tree in 2D space

level of autonomic behaviour can be achieved as the framework will take over many of the monitoring and management processes.

To accomplish this task we implemented an algorithm that, based on the QoE prediction model, estimates the minimum needed changes in the measured stream parameters to improve the QoE [147]. This technique is enabled by the DT prediction models we use for estimating the QoE. The algorithm represents a QoE prediction DT model in the geometric space, defined by the dataset parameters. For each parameter it defines a single dimension in the parameter hyperspace. Each of the datapoints from the dataset can be represented as a point in this hyperspace. The DT is finally represented by hyper-regions formed by the leaves of the DT (Figure 4.20).

Each node of the DT represents a binary split (for binary trees) that forms a hyperplane in the parameter hyperspace. At the bottom, the leaves of the tree carve out hyper-regions, which are bounded by these hyperplanes. These hyper regions are associated with a class label membership, according to the leaf they correspond to. Every datapoint in the dataset actualizes the tests of a route leading to only a single leaf on the DT. Correspondingly, every datapoint falls within a single hyper-region and is therefore classified with the label for that region.

Our algorithm represents the DT in the hyperspace by generating a set of hyper regions that correspond to the tree leaves (Figure 4.21). Each hyper-region contains a set of split rules that define the hyper-surface, which define the boundaries of the hyper-region. The split rules are either representing an inequality of the type  $Parameter_1 \geq Value_1$  or of the type

```

Start from the root node and call a recursive method
FindLeaves

FindLeaves:

1) If the node has children
   a) Call FindLeaves on each child
   b) Add the SplitRule on each of the Hyper Regions ( $\overline{\Phi}$ )
      that are returned
      i) If the leaf split is categorical add a Split Rule:
         Attribute = 'value'
      ii) If the leaf split is continual add on the leaves
           from the left side SplitRule: Attribute < value,
           and on the leaves from the right side Attribute >
           value
   c) Return the set of Hyper Regions ( $\overline{\Phi}$ )
2) Else, you are in a leaf
   a) Create an Hyper Region object
      i) Assign the class of the leaf to the  $\Phi$ 
      ii) Return  $\Phi$ 

```

Figure 4.21: Hyper-region building algorithm

$Parameter_1 = Value_1$  depending on whether  $Parameter_1$  is continual or categorical. If the leaf is on the left side of a continual  $Parameter_1$  split then the split inequality will be 'more than or equal to', if it is on the right side the split inequality will be 'less than'. Having a list of *HyperRegion*-s we can easily determine where each datapoint from the dataset belongs to, by testing the datapoint on the split rules of each hyper region. The hyper region is associated with the same class label as the leaf it represents, so all datapoints that belong to that region are classified according to this label.

In order to improve the QoE of a particular instance of the service, we need to look at the measured value that was acquired by the monitoring system for that instance. If the measurement is classified with a QoE value that is not satisfactory, we look at the distance to the hyper regions that are associated with a satisfactory QoE value. The distance to each of the desired regions is the difference in parameter values that are needed in order to move the datapoint to the desired regions. The output of the algorithm is a set of distance vectors, which define the parameters that need to be changed and their change values.

To illustrate the matter better we can take an example from the lap-

top dataset from [148]. The prediction model built from this dataset is given in Figure 4.20. If we look at the datapoint given in Table 4.2 we can see that this datapoint will be classified by the model as  $QoE = No('NotAcceptable')$ . Since the  $V.Framerate$  is less than 12.5 and the  $V.Bitrate$  is less than 32 the datapoint reaches a leaf with '*NotAcceptable*' class associated with it. Now, what is the best way to improve the QoE of this stream?

Video SI	Video TI	V.Bitrate	V.Framerate
67	70	32	10

Table 4.2: Example Datapoint

First of all there are parameters that can be only observed, such as *VideoSI* and *VideoTI* that characterize the type of the content. In this dataset structure we are looking into increasing the  $V.Bitrate$  and  $V.Framerate$ . If we increase the  $V.Bitrate$  for this particular datapoint by one step to  $64kbits/s$  we can see that the datapoint now arrives at one of the bottom leaves of the DT, but it is still classified as  $QoEAcceptable = No$ . On another hand, if we increase the  $V.Framerate$  to  $15fps$  we can see that the datapoint is classified as  $QoEAcceptable = Yes$  without adding more bandwidth.

We can deduce a rule from the model that a video with these characteristics needs to have higher  $V.Framerate$  for it to be perceived with high quality. However, this rule is not easily evident from only looking at the model. We can also imagine a system with large number of parameters where changing one or more parameters affects the QoE in a complex way. Further down this line of reasoning, if we want to make a system-wise improvement that will increase the QoE of most streams we cannot easily derive which parameters are best to be increased and by how much.

In the case of the example datapoint the algorithm returns the two possible paths:

- Increasing the Framerate to above 12.5f/s;
- Increasing the V. Bitrate to above 32kbits/s and the Video TI to above 87.

Since we know that increasing the *VideoTI* is not an option, because this is a measurement of an inherent characteristic of the video, the only



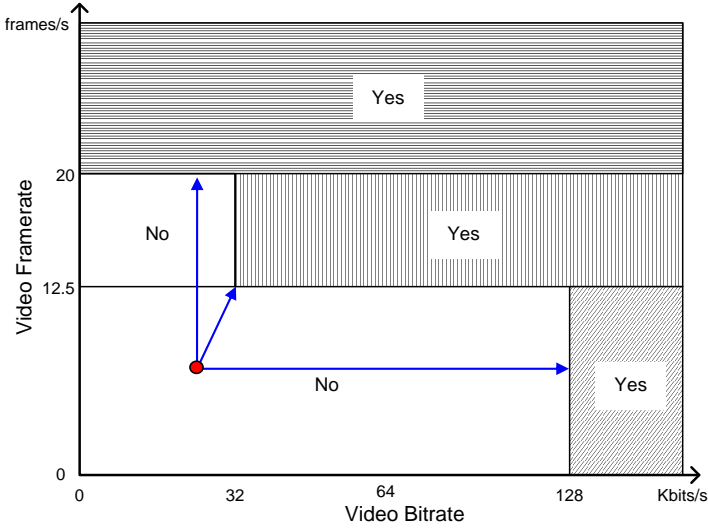


Figure 4.22: Change vector remedies for an example instance

option available is to increase the frame rate. In a general case, there can be many different paths to a hyper-region with the desired class. To automate the process, we can assign cost functions to the change of the attribute values and automatically calculate the cheapest way to reach the desired QoE. In this manner attributes that are observations and cannot be controlled, such as the *VideoTI*, can be excluded from the evaluation by giving them infinitely large cost.

Given a datapoint and a target label, the algorithm produces a set of change vectors. Each of the change vectors applied to the datapoint moves the datapoint to a hyper-region classified with the target label. In other words, each change vector is one possible fix for the datapoint (Figure 4.22).

$$\bar{\Phi} = FindLeaves(DT, QoE) \quad (4.4)$$

$$\Delta\bar{\varphi}_i = Distance(\Phi_i, \bar{d}) \quad (4.5)$$

$$\Delta\varphi_{optimum} = min_i(Cost(\Delta\varphi_i)) \quad (4.6)$$

In equation 4.4, the function returns a set of regions with a targeted QoE value. The distance function in 4.5 calculates the vector of distances for each attribute to the target region in  $\Delta\bar{\varphi}$ . The optimal distance vector is the one with minimal cost 4.6 for the given input datapoint  $\bar{d}$ . The Cost

function in 4.6 is dependent on the application. Each system has explicit and implicit costs associated with changes of specific parameters.

#### 4.6.2 QoE remedies for mobile IPTV

The remedies algorithm has been implemented by extending the Weka [136] platform, so that algorithms such as J48 [129] that induce decision trees can be used to calculate the hyper-regions. Furthermore, we can now measure the distance of any datapoint classified by the DT to the desired hyper-region.

The boxed nodes represent the leaves and map to the hyper-regions as we have seen in Figure 4.22. There are 17 hyper-regions, out of which, only two are with excellent QoE value. The algorithm generates the remedy output specific for each particular broadcasting system.

A target QoE values needs to be defined, and a specific cost for changing a parameter needs to be given as well. If the target value is 'excellent' QoE, the algorithm will calculate the minimum cost of changing specific parameters so that the datapoint falls in one of the two 'excellent' hyper-regions.

A more elaborate QoE improvement is also possible where not all datapoints are targeted for the excellent regions, but the management is executed based on the utility of improving a QoE of a stream in regards to the costs. Then multiple levels of remedies can be suggested by the algorithm with varying costs, and the provider can chose to apply mechanisms to implement the remedies based on their utility to the customers.

This methodology presents a pragmatic solution for estimating and maintaining QoE with a wide range of applicability. Its success and usability depends on the quality of the prediction models, while as architecture it is flexible enough to be used in many different environments.

Since Online Learning techniques also generate DT models they are compatible with the QoE remedies. Both technologies working together deliver flexible remedy response adapting to the changes in the environment reported by the user QoE feedback.

## 4.7 Conclusions

In this chapter an approach for QoE enabled management is presented in the form of a framework. The framework correlates signals coming from

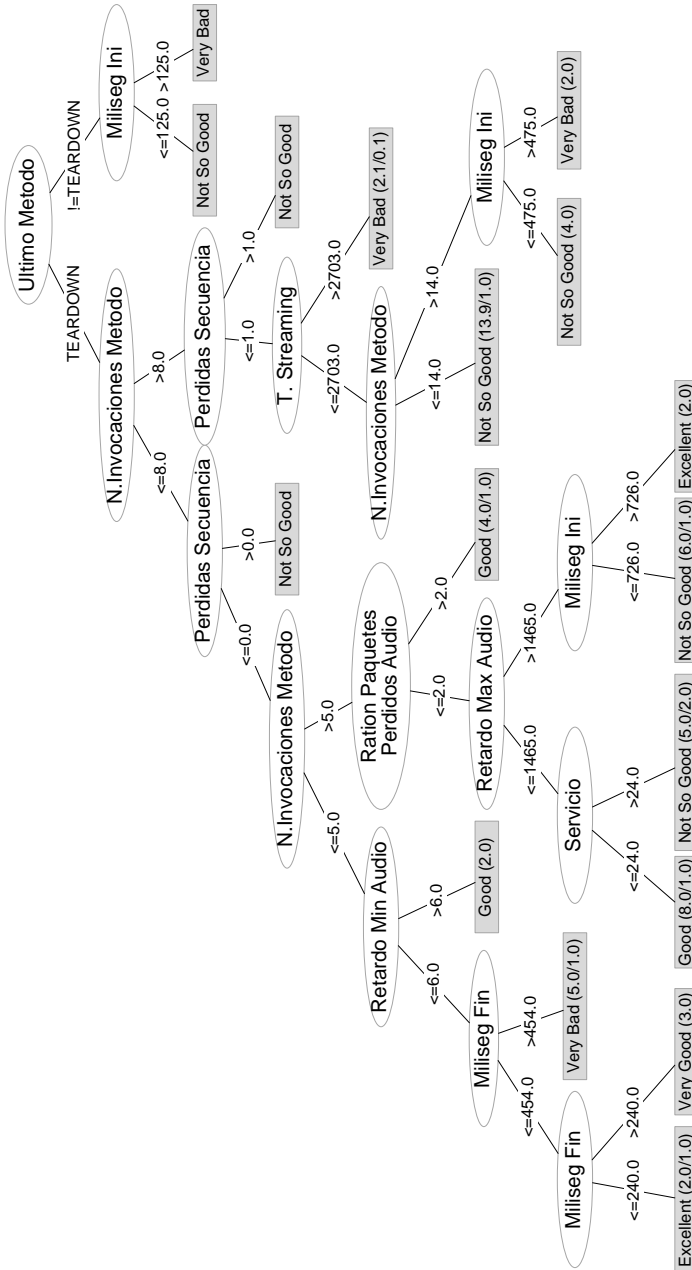


Figure 4.23: DT built from the IPTV subjective feedback

the monitoring system with subjective feedback from the users to develop QoE estimates and management decisions. A description of its application to a mobile IPTV service is also given.

The subjective data used for developing the QoE models in this chapter relied mostly on rating and method of limits. However, as described in Chapter 3, 2AFC methods show superior performance for collecting subjective data. This framework can further benefit from incorporating this kind of subjective data into the QoE models. Certain challenges still remain for the future work. For example, combining MLDS utility curves for certain QPI with QoE models from rating feedback can improve the management decisions due to the accuracy of these curves. Further modifications to the online learning and the remedy algorithms will also be necessary for fully integrating the MLDS models.

This management framework is designed to address management of resources in multimedia systems reactively, taking into account measurements received from the monitoring system. However, a different set of challenges are faced when control decisions need to be implemented in a proactive fashion. In the following chapter we discuss proactive control in multimedia services, solutions we have developed and the CI technologies that enable these solutions.



# 5

## QoE Active Control

The typical role of managing network multimedia services includes optimizing the trade-off between resources and the delivered quality. Typically this means achieving an efficient system where the most users are serviced with the most quality. Successful management also entails adjusting to changes in the environment. Reacting to increase in demand or to the requirements of new devices is essential to meeting customer expectations. However, not all management decisions can be taken reactively. Many video streaming or video conferencing services require proactive control strategies to optimize their performance. Most services are faced with real-time fluctuations in available resources and real-time requirements that need to be met, which makes real-time adaptation a crucial function.

These types of challenges are addressed with active control systems where decisions are taken pro-actively to insure the optimal performance of the service. In this chapter we present a framework for QoE aware active control of multimedia services, based on optimal control and reinforcement learning (RL).

To demonstrate the applicability of our approach we implement a decision support solution using the suggested approach for the case of adaptive video streaming client.

### 5.1 QoE active control framework

The QoE active control (QAC) framework implements control solutions for multimedia systems that can be formulated as a Markov Decision Process (MDP) [149]. MDP is a discrete time stochastic control process, which at each iteration is in a state  $s$  and decides between a finite number of actions  $a$  available in that state. After a specific action is taken, the system transitions to state  $s'$  and a corresponding reward or penalty is accumulated. The transition to the next state is not deterministic; and it is associated with a probability distribution. Solving the MDP means determining the

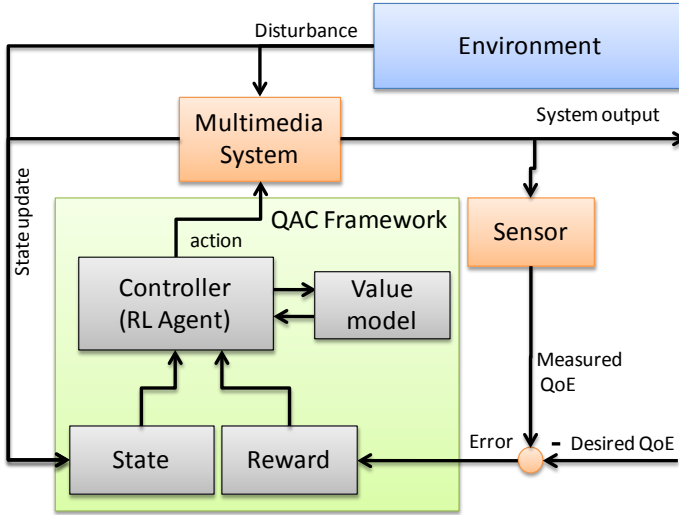


Figure 5.1: QoE active control framework

optimal action at each state that will maximize the reward or minimize the penalty, all the way until the goal is reached.

The QoE active control framework defines the penalty proportionally to the difference between the maximally achievable QoE and the one actually delivered by the service. Instead of proceeding to model the probabilities in the transition matrix, the framework relies on RL to compute the value of each action  $a$  in state  $s$ . This way the framework does not need to compute a vast space of conditions and the probabilities for transitioning between them. It rather needs to explore the space of states and decisions in a process of iterative learning episodes until it discovers the optimal strategy. The state is defined by the condition of the system and the measurement of available resources, and can be partially observable.

Finally, by relying on subjective QoE models to compute the penalty, the framework optimizes the performance of the system in a user centric QoE aware manner. The architecture of the QAC framework is presented in Figure 5.1.

To demonstrate the applicability of the framework we implement a controller for HTTP adaptive streaming client based on this approach. The following sections describe the HTTP adaptive streaming environment and the existing approaches for control in adaptive streaming. This is followed

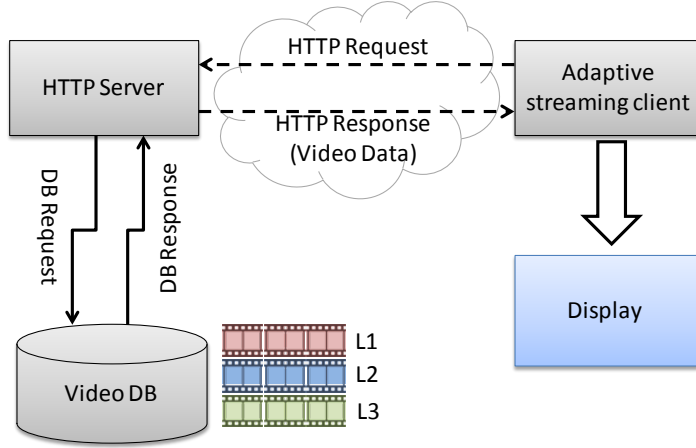


Figure 5.2: HTTP Adaptive Streaming architecture

by a presentation of the intelligent streaming agent, a solution based on our QoE active control framework. Finally, a detailed description of the RL background, the modelling of the penalty function, the state as well as the performance analysis of the agent is presented.

## 5.2 HTTP adaptive streaming client

For a service to achieve high quality video streaming it needs to accomplish a continuous reproduction of the content with sufficiently high bit-rate and without any errors. However, in best-effort networks multiple sources are competing for the same resources and therefore no guarantees are given that resources will be available when needed. Since video streaming is a data-intensive process, it is particularly susceptible to variations in throughput. If the resources are insufficient the video playback will freeze, unless the video is streamed at a lower bit-rate.

To address the variability in available resources, adaptive streaming technologies are developed such as HTTP streaming [150] and SVC [151]. These technologies allow for continuous adaptation of the bit-rate so that a controlled degradation in quality, or quality of experience (QoE), can be achieved.

The HTTP adaptive streaming architecture consists of the same high-level components as other video streaming architectures: servers, transport



system and terminal devices. In this particular case the servers can be any typical HTTP web servers. The transport system is implemented over the Internet and the terminal devices are any device that can run the adaptive streaming application (Figure 5.2).

The video content is transported using the HTTP protocol. The Hypertext transfer protocol (HTTP) is a application-level protocol typically used for transport of web-based content. It is implemented on top of the TCP/IP protocol. HTTP functions in a request-response manner, according to the client-server computing model.

In the case of HTTP adaptive streaming the HTTP servers serve the video content upon receiving a request by a client application. The video content is organized in small segments, or chunks, of few seconds. Different versions of the video with different levels of quality are typically offered. The client can choose to request chunks at specific quality levels (L1, L2, etc), based on its estimate of the available network throughput and its own control strategy.

### 5.2.1 DASH standard

MPEG DASH (Dynamic Adaptive Streaming over HTTP) is a standard for the adaptive streaming over HTTP [150]. The idea behind standardizing adaptive streaming over HTTP is due to the incompatibilities between proprietary implementation that exist today.

Adaptive streaming is implemented by producing different instances of the source video files with different levels of quality. The main characteristic of this approach is that it uses a HTTP server for delivering the video content. In contrast, other video streaming solutions require dedicated video streaming server applications, which complicate the deployment and management of the service. In addition to the widely understood and deployed HTTP servers, adaptive streaming is further advantaged by the use of HTTP packets, which are firewall friendly and can utilize the HTTP caching mechanisms that already exist in the network.

Apple's HTTP Live streaming, Microsoft's Smooth Streaming and Adobe's HTTP dynamic streaming all use HTTP streaming as their underlying delivery method. Yet each implementation uses its own manifest and segment formats. As the standards vary slightly, the players are not compatible with each other. The goal of the standardization effort is to alleviate this divergence.

The DASH standard defines manifest files that allow the clients to iden-

tify the different video streams available. In the standard, the video streams are referred to as Media Presentation and the manifest file is referred to as the Media Presentation Description (MPD).

The MPD is a XML document describing the characteristics of the multimedia content. It has a hierarchical structure [152]. The MPD consist of one or multiple periods, which gave time segments of the multimedia content. Each period consists of one or more adaptation sets. In turn, the adaptation set consist of one or more media components. So one adaptation set contains different bit-rates of the video component, while another adaptation set contains the audio. The media components are defined as representations and consist of multiple segments. The segments are the chunks of media content that the client is requesting from the server.

The client that implements the DASH model needs to first parse the MDP XML file. Next it downloads the representations that are suitable in terms of their characteristics (bit-rate, resolution, and frame-rate) for the available computational and network resources. The client continues to do so as the content is played. The decisions are taken sequentially as the conditions evolve based on the client's control strategy.

The original media description in the MDP file does not contain information about the characteristics of the video itself, such as video SI and TI. These characteristics can help the clients to determine the appropriate subjective QoE models for the content of interest; so we have to incorporate the video characteristics for purposes of QoE management. Subjective QoE models enable more efficient streaming strategies from the point of view of the delivered QoE. Even though the MDP does not specify the use of this type of information, for the implementation of the QAC framework we used extra miscellaneous fields for this purpose.

### 5.2.2 Existing heuristic strategies

As part of the implementation of the QAC framework for adaptive streaming, we started off by evaluating the behaviour of existing clients, through a series of experiments [153]. For the evaluation we implemented a test bed consisting of a HTTP server, an impairment node and a video streaming client (Figure 5.3).

The test bed consists of a server with Apache HTTP server an impairment node running Linux netm kernel module and the client device. The client device also includes a network monitoring software Wireshark, for observing the client network behaviour. We tested the following streaming

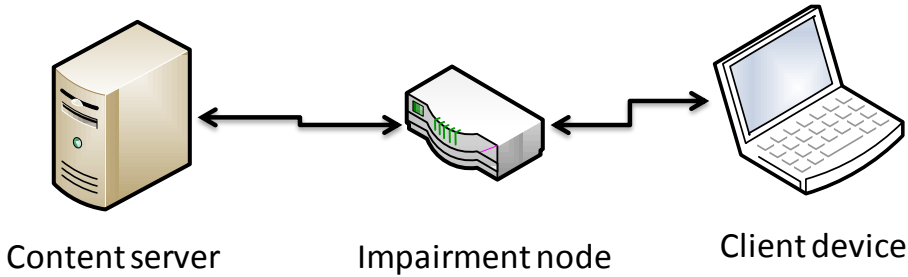


Figure 5.3: Test bed for adaptive streaming client evaluation

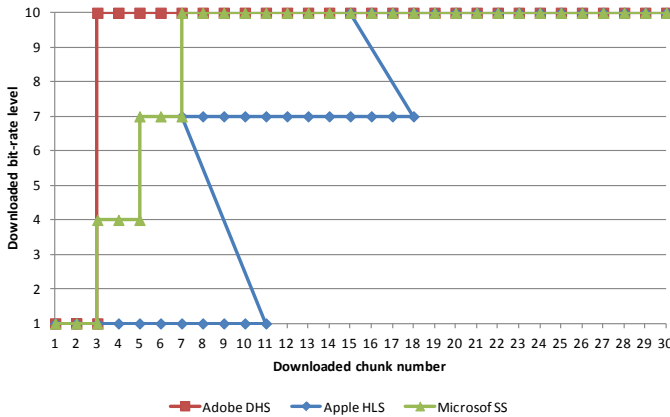
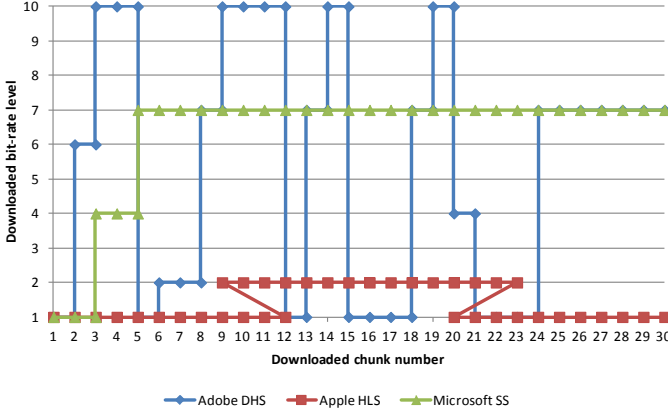


Figure 5.4: Adaptive client behaviour with no throttling

platforms: Apple HTTP Live Streaming (HLS); Adobe HTTP Dynamic Streaming (HDS); and Microsoft smooth streaming (SS). For testing the Microsoft SS, an HTTP module is added to the Apache HTTP server that offers the Smooth Streaming capabilities. For Adobe HDS and Microsoft SS a mp4 container is used, while for Apple HLS uses the MPEG-TS container. The video content is 1 minute long, compressed on 10 different levels ranging from  $64kb/s$  to  $2048Kb/s$ . The videos are segmented in 2 second chunks.

The test is made to determine the QoE performance of adaptation algorithms in face of various network conditions.

The results of the experiments are illustrated in Figure 5.4. This figure is a graphical representation of the sequence with which the video chunks

Figure 5.5: Adaptive client behaviour with  $1Mb/s$  limit

are downloaded. The horizontal axis represents the video chunk number in the sequence and the vertical axis represents the bit-rate at which the chunk is downloaded. If a point is present at  $x = 3$  and  $y = 4$ , then the third chunk is downloaded at  $level = 4$  ( $384kb/s$ ). The sequence of downloads is represented with a line connecting the points.

In all cases, all clients start the download at the lowest bit-rate. This is represented by a point at  $(x = 1, y = 1)$ . Some players, during playback decide to replace the already buffered content with content of higher or lower bit-rate. In this case the figure shows the sequence line going back to a chunk with a lower number at a higher or lower bit-rate level. This is particularly characteristic for the Apple HLS client. The other two clients, when switching levels only repeat the download of the last video chunk.

When exposed to constant throughput exceeding the maximum bit-rate, Adobe DHS starts off downloading just few chunks at the lowest level and then proceeds to the maximum level. On the other hand Apple HLS downloads more chunks on the lowest bit-rate, than makes one intermediate step and ends up at the highest level. Finally, Microsoft SS gradually increases the quality with two intermediate steps, and finally settling on the maximum level.

When the network throughput is limited to  $1Mb/s$  (level 8), the Adobe DHS does four cycles from minimum to maximum bit-rate and finally converges on  $640kb/s$  (Figure 5.5). The Apple HLS client also does two steps, but converges to only  $256kb/s$ . Microsoft SS performs much better, in two

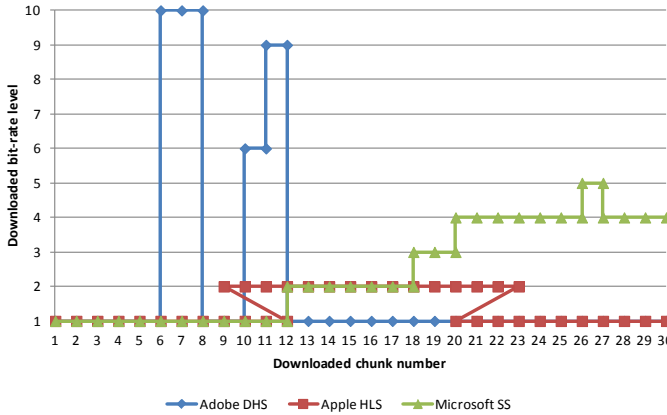


Figure 5.6: Adaptive client behaviour with changing conditions

steps it converges to  $768kb/s$ .

We simulate high variability in network conditions by shaping the throughput in a square wave (pulse train) fashion. The pulses oscillate between  $1Mb/s$  (level 8) and  $100kb/s$  (sufficient for level 1 only) of throughput. The length of the square wave is 10 seconds, remaining 5 seconds at each level. The effective throughput in this case, with sufficient buffering (more than 5 seconds), is  $550kb/s$ .

In these conditions, Adobe HDS frequently cycles through the bit-rate levels and drops to the lowest one for longer periods (Figure 5.6). The client cannot find the effective throughput successfully. Furthermore, it reaches buffer depletion and freezes the playback. On the other hands, the Apple HLS client does not cycle, but remains in the low bit-range rate. The Microsoft SS, starts at very low bit-rates and cautiously increases over time. This client finally settled on  $384kb/s$ , again outperforming the other two clients.

Overall the Adobe HDS makes attempts to improve the QoE by going to higher bit-rates, but ends up causing freezes and frequent quality changes that result in much lower QoE. The Apple HLS avoids freezes, but is very cautious with the bit-rate level. Furthermore, we noticed that it buffers a large amount of the video beforehand. This requires more memory, but is also not friendly to other network users. The Microsoft SS avoids freezes as much as possible, and changes the quality in smoother steps. It strikes a better balance between the bit-rate and the risk of buffer depletion and

uses a buffer of no more than 30 seconds.

In a similar study, the Microsoft SS, the Netflix and the Adobe HDS client were tested [154]. The Microsoft SS is found to be effective under unrestricted and constantly restricted bandwidth. It converges quickly to the maximum available bit-rate and is conservative the quality switching decisions. The Netflix player shows a comparable performance, which is expected since they both use the Microsoft SS platform. The Adobe HDS client does not converge to the appropriate bit-rate even when the throughput is stable. All these findings well aligned to our observations.

In a third study, an evaluation of the changing network conditions in a vehicular environment is implemented [155]. The results show that Microsoft SS achieves the highest average bit-rate and the lowest amount of switching. Apple HLS utilizes the lowest bit-rate and Adobe DHS's performance is the poorest because it introduces freezes.

Overall the three studies reach very similar conclusions. Regardless of the fact that the Microsoft SS performance shows high avoidance of freezes and quality changes, the client does not have any understanding as to how its decisions affect the video QoE. Furthermore, achieving this level of performance without a doubt requires sophisticated heuristics. Adapting them to new devices and content requires significant resources. Instead of this 'design and tuning' approach, we propose an approach of learning or inference, which shortens the development and maintenance time, while providing for high efficiency.

### 5.3 The intelligent streaming agent

The intelligent streaming agent (ISA) takes a different approach in developing its decision strategy, compared to any of the existing solutions. As discussed in the previous section, the evaluated streaming clients use strategies designed by human experts relying on their intuition and experience. However, such heuristic-based solutions are hard-coded can neither adjust to broad range of conditions, nor adapt as new conditions appear. ISA, on the other hand, infers the optimal strategy by exploration. The inference is guided by the value of each strategy, which is determined by the QoE reward accumulated using that strategy. The QoE reward estimation is based on subjective QoE models.

ISA is an adaptation of the QAC frameworks for HTTP adaptive streaming. It consists of an RL agent for control strategy inference and models

for state, action and reward. The models are specifically developed for the particular domain.

The reward function consists of the following factors that affect the QoE in video streaming:

- subjective QoE for the specific video;
- incurred impairments from freezes during playback;
- incurred impairments from change in quality during playback.

The actions are the available bit-rates that the client can choose to download.

The system state captures the conditions of the system that relate to the delivered QoE. These include: the buffer size; buffer utilization; video characteristics; and network performance measurements. The video characteristics include: the size of the video; the current position in the video; resolution; frame-rate; and video SI and TI. The latter two enable the agent to take different actions for different types of video. Since the reward/penalty is calculated using the subjective QoE models, it will differ for different types of video (e.g. content that is low in SI and TI can be compressed more efficiently and requires lower bit-rates). Similarly, the actions need to correspond to the characteristics of the videos.

Based on these three components (state, actions, reward), the agent can now explore and discover the best strategies that optimize the delivered QoE. This inference is implemented using RL training algorithms.

### 5.3.1 Reinforcement learning background

The system under control is issued actions based on the difference between its output and the target or reference output, as defined in control theory (Figure 1.1). This architecture is well suited when the system output needs to follow a certain reference value over time. However, in some applications a series of actions need to be taken in order to reach a goal or to maintain the system in the desired state. In this case a single action is not of concern, but a strategy to generate sequences or actions is needed.

The purpose of reinforcement learning algorithms is to determine the good strategies that, for a range of specific situations, and as the environment conditions change, will generate actions that achieve the desired system performance.

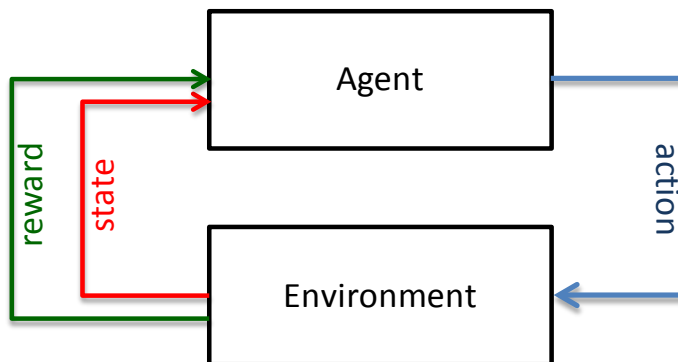


Figure 5.7: Reinforcement Learning framework

Supervised learning is not a good approach for this kind of problems, because it is very impractical to generate examples for all the possible conditions that the system can be in and label the best action in each condition. Furthermore, in many cases there is no single best action; rather the sequence of actions determines the performance of the system.

A simple example is given by turn-based games. Many games can be won in many different ways or by different sequences of moves. The success of the play is determined by the end of the game. The value of a single move without the context of the sequence is undetermined.

The reinforcement learning framework addresses this type of problems as an agent operating in a given environment (Figure 5.7) [156]. The agent takes actions that affect the environment and receives the state and reward according to the changes in the environment. The agent takes decisions on each action from a set of available actions in the current state. After an action has been taken, the state changes accordingly. The reward is calculated based on the changes in the state. The goal of the agent is to optimize its decision in order to accumulate maximum reward.

The environment can be partially observable, so that the state is defined by a probability distribution. Furthermore, the environment can be stochastic, where the actions lead to probabilistic changes in the state.

The inference in RL is implemented by exploring different actions in various states in order to determine the most valuable strategy [127]. Depending on how the reward or penalty function is defined, the best strategy can be: to reach a goal in a minimum number of steps; or to achieve a maximum number of favourable states. If the agent receives a penalty for each



additional step towards the goal, the agent will learn to get to the goal as fast as possible. On the other hand, if reaching certain states on the way is more favourable, the agent will learn take actions accordingly. The agent can even learn to take risks and weigh-in penalties vs. rewards for certain actions (e.g. trial-and-error inference).

The learning process is implemented by updating the model for the state-action values. Since the reward is often delayed, the updates for the values need to be propagated back, to actions taken in the past. Different RL algorithms implement the learning in a different manner.

Model based learning is an approach where no exploration is done, but the optimal actions can be determined using dynamic programming [157]. For this approach the state needs to be relatively small and fully observable. This means that the probability  $P(s_{t+1}|s_t, a_t)$  of transitioning to state  $s_{t+1}$  from state  $s_t$  given action  $a_t$  needs to be available for all possible transitions and actions. As well as the probability for the received reward  $p(r_{t+1}|s_t, s_{t+1})$ .

Full understanding of the environment is not usually available, or the cost of computing  $P(s_{t+1}|s_t, a_t)$  and  $p(r_{t+1}|s_t, s_{t+1})$  is too high for the given state-action space. Hence, the agent needs to explore the environment in order to build the value model for the action-state.

When the agent is exploring, it can observe the value of future states and the reward collected on the way. This information can be used to update the value of the current state. Algorithms that implement this approach are referred to as temporal difference algorithms because they compare the current value of the state (or state-action pair) and compare it to the value of the next state and the reward received.

The online update of the value is implemented by using the delta rule. The simplified version of the update is given in equation 5.1, where the system can be in only one state. The value of action  $a$  in this state is defined as  $Q(a)$ . The reward received at time  $t + 1$  is  $r_{t+1}$ . The value of taking the action at time  $t + 1$  ( $Q_{t+1}$ ) is updated based on the previous value ( $Q_t$ ) and the received reward ( $r_{t+1}$ ).

$$Q_{t+1}(a) \leftarrow Q_t(a) + \eta(r_{t+1} - Q_t(a)) \quad (5.1)$$

$\eta$  is the learning factor, which is gradually decreased over time for convergence. The convergence occurs when the value of taking the actions is equal to the reward  $Q_t(a) \leftarrow r_{t+1}(a)$

In a more general case, the value of action  $a$  in state  $s$  is  $Q(s, a)$ . In this

case the system evolves through different states as the actions are taken. Now the value of state-action pair is related to all the rewards that follow after the action has been taken. The update rule is given in equation 5.2

$$\hat{Q}(s_t, a_t) \leftarrow \hat{Q}(s_t, a_t) + \eta(r_{t+1} + \gamma \max_{a_{t+1}} \hat{Q}(s_{t+1}, a_{t+1}) - \hat{Q}(s_t, a_t)) \quad (5.2)$$

Instead of just looking at the reward after taking one action, now we need to value the state-action by rewards that are also coming in the following states, hence we take  $r_{t+1} + \gamma \max_{a_{t+1}} \hat{Q}(s_{t+1}, a_{t+1})$ , as the value for update. The expression  $\max_{a_t} \hat{Q}(s_{t+1}, a_{t+1})$  represents the maximum value that can be achieved with any action. The value of the future states propagated back is discounted by the factor  $\gamma$ , so that the values do not grow to unmanageable sizes as the size of the state space grows.

As the rewards and the state space transitions are probabilistic, the expression  $r_{t+1} + \gamma \max_{a_{t+1}} \hat{Q}(s_{t+1}, a_{t+1})$  is basically a sample from those probabilities. The  $Q$  values are now estimates  $\hat{Q}$ , which converge to the mean values of the probability distributions.

The approach is utilized in the Q-learning algorithm (Algorithm 1).

---

**Algorithm 1** The Q-Learning algorithm

---

```

for all episodes do
  Initialize  $s$ 
  repeat
    Choose an action  $a$  using  $\epsilon - greedy$  exploration
    Take action  $a$ , observe  $r$  and  $s'$ 
     $Q(s, a) \leftarrow Q(s, a) + \eta(r + \gamma \arg \max_{a_{t+1}} Q(s', a') - Q(s, a))$ 
     $s \leftarrow s'$ 
  until  $s$  is in terminal state
end for

```

---

The Q-Learning algorithm always uses the action that produces maximum-valued states to update the current value. This approach is referred to as off-policy learning [127].

However, the action that the agent selects to move to the next state is actually selected by the exploration strategy. The typical exploration strategy is  $\epsilon - greedy$ . In this strategy the agent selects with probability  $\epsilon$  uniformly between all the possible actions and with probability  $(1 - \epsilon)$  it selects the best known action.

Alternatively on-policy methods use the action chosen by the exploration strategy to update the value. The SARSA method is an example of this approach (Algorithm 2)

---

**Algorithm 2** Sarsa

---

```

for all episodes do
  Initialize  $s$ 
  Choose an action  $a$  using  $\epsilon - greedy$  exploration
  repeat
    Take action  $a$ , observe  $r$  and  $s'$ 
    Choose an action  $a'$  using  $\epsilon - greedy$  exploration
     $Q(s, a) \leftarrow Q(s, a) + \eta(r + \gamma Q(s', a') - Q(s, a))$ 
     $s \leftarrow s', a \leftarrow a'$ 
  until  $s$  is in terminal state
end for

```

---

The current algorithm only updates the previous action/state value, one step in the past. Converging requires passing multiple times over the same states. A way to improve the performance of the RL algorithms is to use Eligibility Traces (ET). ET are records of when the algorithm passed over certain state-actions. Every time the agent at state  $s$  takes action  $a$ , the trace  $e(s, a)$  is set to 1. At the same time all other traces are decayed by  $\gamma\lambda$  (Equation 5.3).

$$e_t(s, a) = \begin{cases} 1 & \text{if } s = s_t \text{ and } a = a_t, \\ \gamma\lambda e_{t-1}(s, a) & \text{otherwise.} \end{cases} \quad (5.3)$$

Now instead of updating just the last visited state we update all the states proportional to their eligibility trace. So, if the state-action is recent in the past the update is more significant, and if the state-action has been visited in more distant past the update is less significant. The temporal difference in SARSA at time  $t$  is  $\delta_t$  (Equation 5.4).

$$\delta_t = r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \quad (5.4)$$

The update with the eligibility traces takes the form given in Equation 5.5.

$$Q(s, a) \leftarrow Q(s, a) + \eta \delta_t e_t(s, a), \forall s, a \quad (5.5)$$

The  $\lambda$  parameter is the temporal credit. For  $\lambda = 0$  the algorithms is updated only one step in the past as in SARSA. The closer  $\lambda$  gets to 1 the longer the updates are made into the past action-states. This algorithm is referred to as SARSA( $\lambda$ ) (Algorithm 3).

---

**Algorithm 3** SARSA( $\lambda$ )

---

```

Initialize all  $Q(s, a)$  arbitrarily,  $e(s, a) \leftarrow 0, \forall s, a$ 
for all episodes do
  Initialize  $s$ 
  Choose an action  $a$  using  $\epsilon - greedy$  exploration
  repeat
    Take action  $a$ , observe  $r$  and  $s'$ 
    Choose an action  $a'$  using  $\epsilon - greedy$  exploration
     $\delta \leftarrow r + \gamma Q(s', a') - Q(s, a)$ 
     $e(s, a) \leftarrow 1$ 
    for all  $s, a$  do
       $Q(s, a) \leftarrow Q(s, a) + \eta \delta e(s, a)$ 
       $e(s, a) \leftarrow \gamma \lambda e(s, a)$ 
       $s \leftarrow s', a \leftarrow a'$ 
    end for
  until  $s$  is in terminal state
end for
```

---

However, when the size of the state space starts to grow, the RL algorithms performance starts to deteriorate. In many cases the states can be very similar and the state-actions can have similar values. This means that we can compress the space by making a model that maps the large state space into a smaller one.

One particular example for our case is the state when the available bandwidth supersedes the bit-rate of the highest quality video. In this case we can take the action to download the highest quality, regardless if the bandwidth is twice the bit-rate or ten times the bit-rate. So, these two states can be mapped into a single state, since the value for the action will be the same. This way the algorithm does not need to visit all possible states to have an approximate value for the actions in that state.

This is a supervised learning approach, where we need to train a model based on the examples available. Now instead of having a table of  $Q(s, a)$  for all possible  $(s, a)$  pairs, we need a model that maps  $(s, a)$  into the value  $Q$  ( $Q = f(s, a)$ ). This model can be represented as a linear combination

of all the features  $\Phi$  parameterized by a vector  $\vec{\theta}$ . The features are binary parameters that characterize the state-action pair. In this case,  $Q$  is calculated as given in equation 5.6.

$$Q = \sum_{i \in \Phi} \theta(i) \quad (5.6)$$

Instead of updating the values of the table  $Q(s, a)$ , now we need to updated the parameters  $\vec{\theta}$ . Since the RL algorithm updates the state online, a suitable approach is to use gradient descent to update the  $\vec{\theta}$  parameters. In this method these parameters are update as given in equation 5.7.

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha \delta_t \vec{e}_t \quad (5.7)$$

The number of eligibility traces ( $e_t$ ) now do not correspond to all possible state-action pair, but to all the features that define the  $(s, a)$  pair.

Finally to complete the gradient descent SARSA( $\lambda$ ) algorithm we need a mechanism to extract the features from the state-action pairs. Since the features need to be binary this is implemented with a tiling technique [156]. Each discrete parameter in the state-action pair adds a binary digit for each possible value that it can contain. The continual parameters need to be discretized by tiling the space they occupy. The tiles create discrete regions of space. If the value of the parameter falls on a specific tile, its corresponding feature value is 1 or, otherwise, is set to zero. For reasons of efficiency, the continual parameters are joined together in a multidimensional space where the tiles form hyper volumes. This discretization causes loss of fidelity. In order to decrease this effect, more sets of tiles are laid over the parameter space and randomly shifted by different margins, so that they cover slightly different areas.

Our ISA framework implements the gradient-descent SARSA( $\lambda$ ) algorithm for training.

### 5.3.2 Intelligent streaming agent architecture

The components of our intelligent video client are shown in Figure 5.8. The download manager downloads the video chunks as indicated by the controller. It also measures network throughput statistics and updates the state model accordingly.

The downloaded video data is stored in the video buffer. The display interface pulls the highest quality video data from the buffer and updates

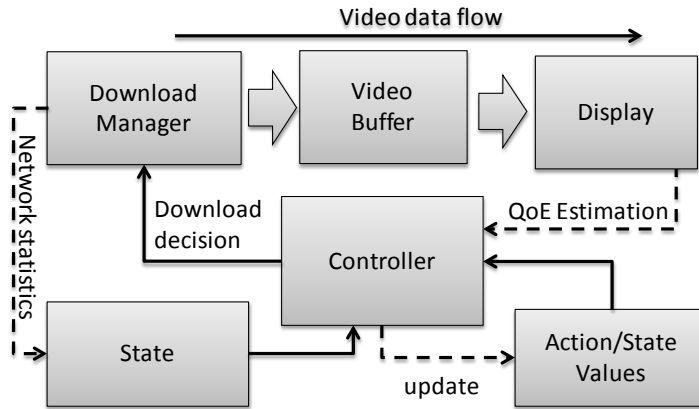


Figure 5.8: The client application architecture

the device display. The display also reports the quality level of the displayed segment to the controller, so that the QoE reward/penalty estimation can be calculated. The controller incorporates a RL agent that continuously selects the optimal decision based on the state of the system and the value for each action at that state. The controller updates the state-action value model based on the reward/penalty accumulated.

The state model contains information about:

- the video spatial and temporal information;
- the video resolution, frame-rate and other objective features;
- the video length;
- the current position in the video stream;
- short term and long term prognosis of the network throughput.

The state is discretized to form a set of binary features using the tiling technique, as described in the previous chapter.

To goal of the agent is to maximize the reward or minimize the penalty. In this architecture the reward is negative and only reaches zero when the highest possible level of quality was achieved. It is calculation is modelled on subjective QoE data.

The controller training implements the linear gradient-descent SARSA( $\lambda$ ) algorithm. Its training is implemented by simulating video

streaming playback in a network environment with self-similar background traffic.

Details about the implementation of the reward/penalty calculation, the network throughput prognosis and training performance of the agent are presented in the rest of the chapter.

### 5.3.3 Reward/penalty function

The reward/penalty function for the RL agent calculates the delivered QoE based on the quality of the playback. More precisely the QoE function calculate degradation or a negative reward. The function returns 0 when the playback has encountered no freezes and the video was reproduced with maximum possible bit-rate. As soon as the player chooses to reproduce a lower bit-rate segment, the function returns a negative value. The value is proportional with the degradation given by the subjective MLDS model for the particular type of video. Furthermore, the function includes a quality model for video freezes and for changes in the level of quality. Every time there is a change in quality or a freeze in playback, there is a drop in delivered QoE. Since the changes in quality and the freezes in playback are bursty and localized impairments, their effects on the quality is not constant but a function of time.

The effect of the impairment on the QoE starts at the moment it is introduced. Then it increases in time with a negative gradient. When the impairment stops, the effects on the QoE decay is again with a negative gradient. The next impairment may come before the effects from the previous one have diminished. In this case the effect is cumulative. This type of impairments has a negative effect on QoE, which is proportional to the frequency of their occurrence and their amplitude. Capturing all these effects, we calculate the QoE as given in equation 5.8.

$$\begin{aligned}
 QoE &= w_s f_{subjective}(\text{bit-rate}, video_{si}, video_{ti}) + \\
 &+ w_f f_{freeze}([(t_{p_1}, t_{s_1}), (t_{p_2}, t_{s_2}) \dots (t_{p_n}, t_{s_n})]) + \\
 &+ w_l f_{lvlChange}([\delta_1, t_{l_1}), (\delta_2, t_{l_2}) \dots (\delta_n, t_{l_n})])
 \end{aligned} \tag{5.8}$$

The  $f_{subjective}$  function calculates the degradation due to restrictions in bit-rate. It takes into account the characteristics of the video (spatial and temporal information) and returns a relative value of degradation. A typical subjective quality curve obtained with the Maximum Likelihood Difference Scaling method [72] is presented in Figure 3.9.

The  $f_{freeze}$  function calculates the degradation incurred by the freezes in the playback. The value is based on research done on the psychological effects of this type of impairment. The  $f_{freeze}$  inputs a list of pair values. The first ( $t_{p_i}$ ) is the time at which the  $i^{th}$  freeze started and the second ( $t_{s_i}$ ) is the time at which the playback continued. The effect of the freezes is cumulatively collected over from the beginning to the end of playback as given in equation 5.9.

$$f_{freeze} = \int_0^{t_{end}} I_f(t) dt \quad (5.9)$$

The amplitude of the degradation ( $I_f(t)$ ) is proportional to the length of the impairment in time. However, this proportion is not linear. The freeze of 1 seconds is does not cause half the impairment of a freeze of 2 seconds. Nor a freeze of 20 seconds is half as damaging as an impairment of 40 seconds. The gradient of degradation is high in the beginning and drops over time. If we define the impairment on a relative scale from 0 to 1, we can use an exponential decay function to model the degradation as given in equation 5.10 where  $\lambda_d$  is the half life (or the time the quality needs to decrease to half of its initial value). After the freeze has ended and the playback is restarted the impairment remains in the memory of the viewer for a period of time. This period of forgetting or forgiving is also be modelled with a decay function. However, in this case the impairment amplitude is decayed to 0, or the quality rises up to 1. So in the period after the restart of the playback the quality due to freezes can be calculated as given in equation 5.10.

$$I_f(t) = \begin{cases} e^{-\frac{t-x_p}{\lambda_d}} & \text{if playback stopped} \\ 1 - e^{-\frac{t-x_s}{\lambda_r}} & \text{if playback continues} \end{cases} \quad (5.10)$$

where  $x_p$  and  $x_s$  are the shifts on the time axis and are calculated as given in equation 5.11 and 5.12 respectively.

$$x_p = t_p + \lambda_d \ln(I_f(t_p)) \quad (5.11)$$

where  $t_p$  is the moment the impairment started, and  $I_f(t_p)$  is the amplitude of quality at  $t_p$ .

$$x_s = t_s + \lambda_r \ln(1 - I_f(t_s)) \quad (5.12)$$



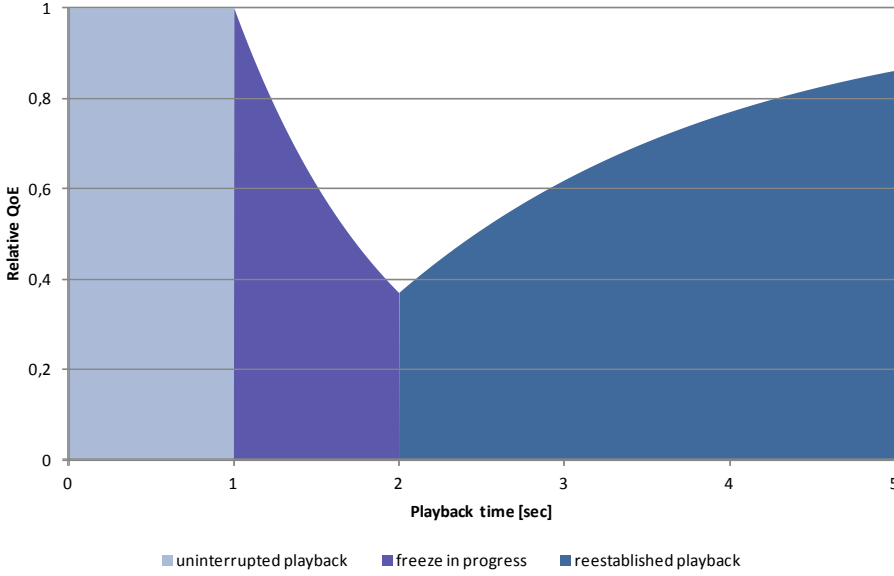


Figure 5.9: Relative impairment from a freeze

where  $t_s$  is the moment the impairment stopped, and  $I_f(t_s)$  is the amplitude of the quality at  $t_s$ .

The parameter  $\lambda_r$  is the 'half life' of the repair time after the impairment. Both  $\lambda_d$  and  $\lambda_r$  need to be estimated through subjective studies.

A depiction of the impairment of a freeze during playback is given in Figure 5.9.

The similar approach is taken for the  $f_{lvlChange}$  (equation 5.13). However, the amplitude of the impairment here is the difference in the distance between levels of quality. And after each occurrence of the change the impairment effect decays to zero (Figure 5.10).

$$f_{lvlChange} = \int_0^{t_{end}} I_l(t) dt \quad (5.13)$$

The amplitude of the quality degraded by level change is given in equation 5.14.

$$I_l = 1 - e^{-\frac{x-x_l}{\lambda_l}} \quad (5.14)$$

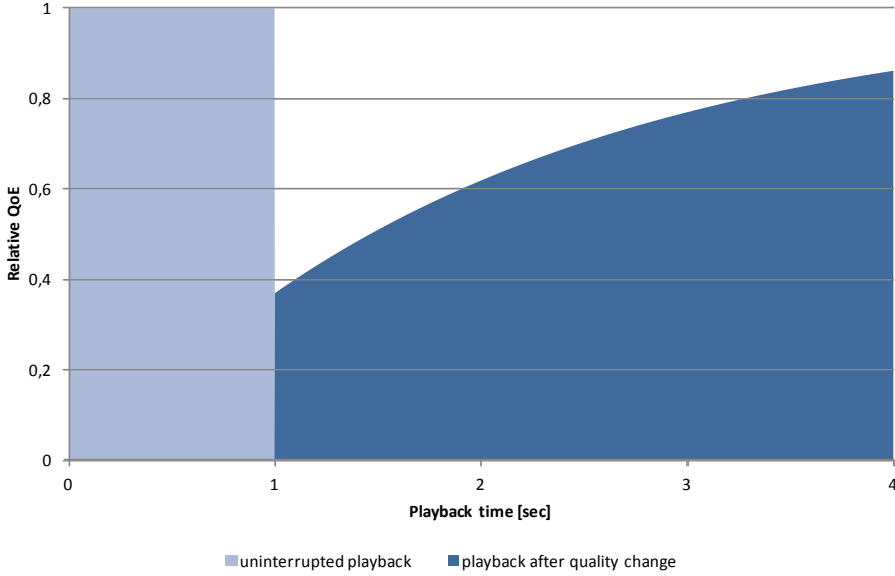


Figure 5.10: Relative impairment from quality change

where  $\lambda_l$  is the half life of the level change impairment decay and  $x_l$  is calculated as given in 5.15.

$$x_l = t_l + \lambda_l \ln(1 - I_l(t_l) \frac{N_l - |\delta_l|}{N_l}) \quad (5.15)$$

where  $t_l$  is the moment the impairment happened, and  $I_l(t_l)$  is the amplitude of the quality at  $t_l$ ,  $N_l$  is the number of quality levels and  $\delta_l$  is the difference in quality level between the video that was played before  $t_l$  and the video that was played after  $t_l$ .

Since the three types of impairments have different impact on QoE, they are weighted differently:  $w_s$ ,  $w_f$  and  $w_l$ . These weights can be adjusted by executing a subjective trial and estimating their effect on the QoE.

### 5.3.4 Estimating network throughput trends

The streaming agent can observe the speed with which the chunks of video are arriving and make assumptions on the available network throughput. This is a passive assessment of available network resources, which requires no additional components and resources. This is why it is a commonly

used solution in network streaming clients. Since the network is a shared resource, the available throughput can be highly volatile. However, the resources do not stochastically appear and disappear. The resources are consumed by other services in the network. As these background services (from the point of view of the agent) start and stop, they tend to form certain trends in the available resources.

By looking at the arrival rate of the video data, the agent needs to make assumptions or forecasts about the trends in the available throughput. The accuracy of these forecasts affects the performance of the agent in terms of the delivered QoE.

If the agent had perfect information about the available throughput and could make perfect predictions, we would be dealing with a fully observable system whose actions would be determined via dynamic programming. However, we are dealing with a more challenging situation where future traffic is unknown. We need to "learn" good strategies based on the best estimations available. Estimating trends based on sequences of samples is also referred to as filtering [158]. The challenge in filtering is to overcome the random fluctuations in the data, or the noise, and to detect the trends. However, there is a trade-off between the filtering of high frequency fluctuations and the speed of detecting trends. If the filter is disregarding a wider range of fluctuations it becomes slow to react to trends [159]. On the other hand, if the filter is following the changes in the input too closely, its predictions are short sighted and include the noise. This way average error between the predictions of the filter and the actual measured values is high.

Many strategies to deal with this challenge have been proposed. One of the most popular approaches is the exponentially weighted moving average (EWMA) filter [160]. The filter observes values  $O_t$  and outputs the estimations  $E_t$ , where  $E_t$  is calculated as given in equation 5.16.

$$E_t = \alpha E_{t-1} + (1 - \alpha) O_t \quad (5.16)$$

The  $\alpha$  parameter is the smoothing parameter of the filter. High value produce more smoothing, giving a filter that slowly reacts to changes in the data trends. Correspondingly, lower  $\alpha$  value make the filter less stable and more agile.

Since both stability and agility are desirable features of the filter in [161] suggest an adaptive approach where the values of  $\alpha$  is not constant, but it changes adaptively 5.17.



Figure 5.11: Performance of typical filters

$$E_t = \alpha_t E_{t-1} + (1 - \alpha_t) O_t \quad (5.17)$$

The Vertical Horizontal Filter (VHF) solution [159] proposes that the smoothing parameter is computed as in equation 5.18.

$$\alpha_t = \beta \frac{\Delta_{max}}{\sum_{i=t-M}^t |O_i - O_{i-1}|} \quad (5.18)$$

$\Delta_{max}$  is the gap between the maximum and the minimum value in the  $M$  most recent observations.  $\beta$  is empirically set to 0.33.

A stability filter dampens the estimates in proportion to the variance of spot observations [162]. The goal is to increase the smoothing when the network exhibits unstable behaviour while keeping the filter stable. On the other hand, keeping low smoothing when the network is more stable results in the filter closely following the trends. To compute the level of instability, this filter uses another EWMA filter as given in equation 5.19.

$$U_t = \beta U_{t-1} + (1 - \beta) |O_t - O_{t-1}| \quad (5.19)$$

Where  $\beta = 0.6$  (selected empirically) and  $x_t$  is the value measured at time  $t$ . The smoothing parameter  $\alpha$  is set as in equation 5.20.

$$\alpha_t = \frac{U_t}{U_{max}} \quad (5.20)$$

where  $U_{max}$  is the largest instability seen in the 10 most recent observations.

The error based filter is another variation of the adaptive filtering approach, where the gain is adapted according to how well the filter predicts the measurements [162]. When the filter does not accurately predict the future values, the gain is decreased so that the filter estimation will converge more quickly. The error observations are  $|E_t - O_t|$ . These error observations are filtered through a secondary filter and the estimation error is finally as defined in equation 5.21.

$$\Delta_t = \gamma \Delta_{t-1} + (1 - \gamma) |E_{t-1} - O_t| \quad (5.21)$$

where  $\gamma = 0.6$  (selected empirically). For the Error based filter the gain is calculated as in equation 5.22

$$\alpha_t = 1 - \frac{\Delta_t}{\Delta_{max}} \quad (5.22)$$

where  $\Delta_{max}$  is the largest instability seen in the 10 most recent observations.

After observing the performance of the filters on the simulated background traffic (Figure 5.11) we selected a combination of low smoothing (fast) EWMA, high smoothing (slow) EWMA, stability and VHF estimations output as part of the state. In this manner the agent can use the strengths of the different filters to deduce the best strategies when exposed to different traffic patterns.

### 5.3.5 Simulating the background traffic

modelling Internet traffic is a lively research area [163]. Different theories propose that the Internet traffic is self-similar [164]. Measurements also demonstrate self-similarity of Internet traffic [165]. Self-similarity implies that the traffic distribution is of the same kind at all time scales. Natural examples of self-similar forms are fractals. They are geometrically similar over all spatial scales. The Internet traffic, however, is statistically self-similar over different time scales [165].

Part of the reason for self-similarity lies in the long tailed distribution of file sizes. Most files are small, very few files are very big. The distribution

of file size is long tailed. Empirically, from file sizes on the world-wide-web (WWW) the distribution follows a Pareto model [164].

One particular model fits most of these characteristics and is a good fit for modelling Internet traffic. The Poisson Pareto burst process (PPBP) is a simple but accurate traffic model [166].

The length of the bursts of background traffic is distributed with a long-tailed Pareto distribution. The number of new sources in each iteration is distributed with a Poisson distribution. An aggregation of the traffic generated of these sources is self-similar.

The PPBP also has the highly attractive property that its variance-time curve (the variance of the total traffic arriving in an interval of length  $t$ , as a function of  $t$ ) is asymptotically, for large  $t$ , the same as fractional Brownian noise with Hurst parameter  $H > 0.5$ , which is the form that has been observed in real traffic in many studies [167].

The number of new processes started in each iteration is drawn from a Poisson distribution (Equation 5.23).

$$P(x = k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (5.23)$$

The length of the file downloaded by each process is sampled from a Pareto distribution 5.24.

$$P(X > x) = \begin{cases} \left(\frac{x_m}{x}\right)^\alpha & \text{for } x \geq x_m, \\ 1 & \text{for } x < x_m. \end{cases} \quad (5.24)$$

Example background traffic generated by the PPBP is given in Figure 5.12.

### 5.3.6 Agent performance

We trained the agent in an environment of simulated background traffic and followed the rate of its learning by measuring the delivered QoE after each training episode. The values for the weights of the QoE function were selected as 1 for  $w_s$ , 2 for  $w_l$  and 10 for  $w_f$ . With this selection the penalty for the level change is twice as big as the one for the constant level of quality, and the penalty for a freeze is ten times as big. These values are selected intuitively, based on some simplistic tests. The accurate proportions between the effects of each of the three factors on the QoE need to be established through a comprehensive subjective study.

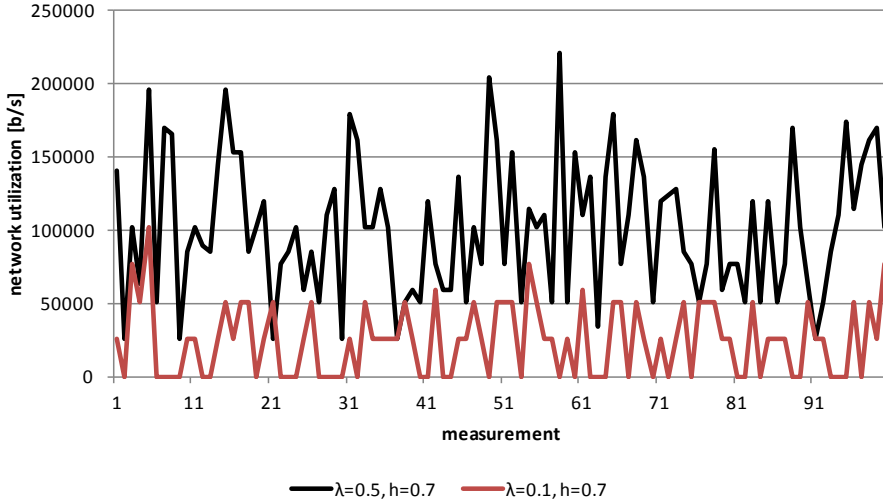


Figure 5.12: Simulated background traffic with  $\lambda = 0.1$  and  $\lambda = 0.5$  with *hurst* = 0.7

The agent undertakes a training regimen of 1000 episodes. Over each episode a video streaming session is simulated. Background traffic is generated using the PPBP model, where the *hurst* parameter is set to 0.7 and  $\lambda$  varies between 0.1 and 1. This creates conditions between very low and very high background traffic. During the training episodes, the penalty is calculated based on the quality level of the played video, the quality changes and the freezes. The penalty incurred over each consecutive episode is given in Figure 5.13.

The agent learns to avoid freezes quickly, since this is heavily penalized in the QoE function. The results show that the RL agent is fully capable of inferring the appropriate strategy defined by the penalty function, and can be successfully implemented into a video streaming client.

Future developments can possibly implement training on network traces in addition to the simulations. This opens possibilities for further exploration of the predictive capabilities on natural patterns in the network.

## 5.4 Conclusions

Many multimedia services need to include online control mechanisms to optimize their service. Often these control mechanisms are designed with

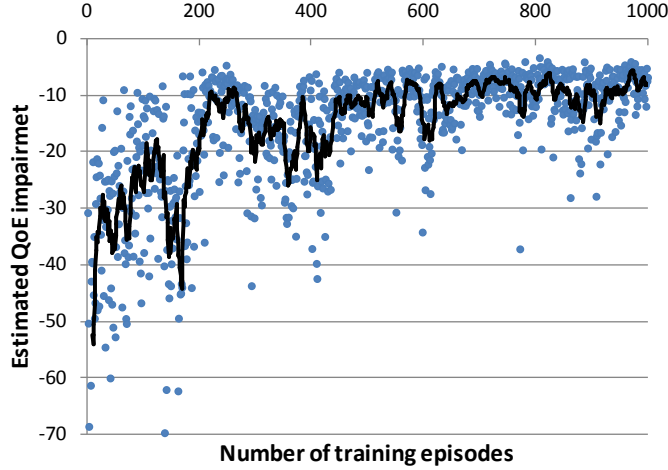


Figure 5.13: Performance of the RL intelligent agent during training

heuristics based on the experience and intuition of the system architects. However, with the growth in complexity of the multimedia systems designing efficient heuristics becomes more difficult. The solution presented in this chapter provides a framework for implementing online control mechanisms that does not require design, but rather infers the optimal strategies. We presented a proof-of-concept implementation of the framework in the form of a HTTP adaptive streaming client to demonstrate the capabilities of this approach.

The implemented solution provides few advances over existing approaches. It relies on subjective models for optimizing the delivered quality. The accuracy of the subjective models in estimating the delivered QoE offer possibility for superior decisions, compared to the existing approaches. There is no design of heuristics involved. The agent is flexible and highly adaptable. It can train and improve continuously. It infers complex patterns in the traffic and develops appropriate strategies. Finally, since the agent implements autonomic learning, updating the strategy to new content or devices is easily implemented by updating the penalty calculation. The agent in turn will infer the new specific strategies for the novel components. Adding new or more advanced sensors for the network conditions can be implemented by adding more features to the state. There is no need for redesign of new heuristic rules either.



The QAC framework provides a flexible solution to the problem of adaptive streaming. However, it can also be used more generally in other control problems where it is hard to model the system deterministically. Particularly in solutions where human perception (subjective) factors in are key to the performance of the system.

# 6

## Conclusions

Multimedia services have become essential to how we interact with each other, exchange information, or entertain. Multimedia content has even entered the domain of printed media, as books and magazines are now augmented with content for tablets and mobile phones.

With ubiquitous connection, video streaming has gone mobile. We watch movies, TV and user generated content on the move, whether we are at home or in the train. These technologies further enable sharing experiences by streaming what we see to other users, as we see it. This trend shows no signs of slowing down, on the contrary, with developments of wearable computing devices and augmented reality, this trend is expected to increase.

Evidently the number of services is growing with the growth of device capabilities and underlying network and display technologies. With a plethora of devices available, adapting the service efficiently is a challenging proposition. With standardized technologies, such as SD TV, the service parameters are defined precisely. Hence, the delivered quality is easy to measure. However, faced with a wide range of devices with different features selecting the appropriate resolution or frame-rate for each possible condition is not trivial. Should the service providers spend all available resources to deliver each pixel of a video to a perfect accuracy in order for the service to have acceptable quality? Is this feasible, and more importantly, is it necessary?

The introduction of this thesis gives a short discussion on the limitation of the HVS and the masking effects resulting from those limitations. These limitations have been successfully utilized to optimize the quality of multimedia content. Encoding algorithms save on precious bits by disregarding details that are not noticeable by the viewer. Accuracy mechanisms in networks have been substituted for error correction or error concealment mechanisms so that delays are minimized. The specifications for a high quality service have become vaguer. Pushing the limits of technologies to

deliver richer features, higher resolution, shorter delays, real-time streaming very often means relaxing some expectations such as reliability or accuracy. Faced with the dilemma of offering more and assuring high quality, service providers are faced with the difficulties of estimating the quality of their service.

This brings us to QoE as a new metric that addresses need to measure the quality in these newly developed conditions. However, faced with the complexity of current multimedia systems, the many factors that affect QoE and the continually evolving environment, measuring QoE is a challenge on its own.

In Chapter 2, we address the objective measurements of factors that contribute to QoE. QoE is evidently a subjective metric, but there are many objective factors that can deliver valuable insight into the level of delivered quality. The results from our evaluation of objective QoE methods demonstrate that, even very simple algorithms, restricted to specific conditions can measure important aspects of quality degradation. On the other hand, complex and sophisticated metrics produce results well correlated with subjective QoE in much wider set of conditions. Understanding how to utilize the objective measurements for evaluating QoE is important for efficient QoE management. Measurement of objective factors is precise, low costs and can be easily automated. Each of these characteristics is important for an efficient QoE management framework.

Nevertheless, the key aspect for understanding the QoE is successful subjective quality measurement. Chapter 3 discusses existing subjective QoE methods and their drawback. In this chapter we propose a new and more effective way for evaluating video QoE via difference scaling, rather than absolute rating, as a way to achieve accurate subjective video QoE measurement. The MLDS method does not deliver absolute quality ratings, but it provides models that illustrate the utility of the resources against the delivered quality. These models are very well suited for delivering efficient management decisions, as they can be combined with cost functions to derive continuous utility functions.

Understanding how to measure objective QoE factors and the delivered subjective QoE is not necessarily enough to guarantee efficient management of a multimedia system. In Chapter 4 we present the challenges faced in QoE-based management of a video streaming system. The sheer number of factors involved in this approach practically prohibits the use of subjective QoE modelling. Correlating monitoring data, objective and

subjective measurements is necessary to manage the QoE of the service. In this chapter we introduce an approach based on CI technologies, as a way to deal with the complexity in determining the highly non-linear relationships among the many monitored parameters and the delivered QoE. We present methods for capturing these relationships into QoE prediction models. The subjective QoE prediction models provide for estimating the subjective QoE based on objective measurements. We further present on-line learning solutions to deal with the continuous evolution of technology and expectations in the context of multimedia streaming services. Finally, we present a method for calculating QoE remedies, as a way to determine which management decisions can deliver satisfactory QoE to the end user.

In Chapter 5 we turn our focus to the many time-sensitive multimedia systems where rapid control decisions determine the level of delivered quality. Typically, video streaming services need to make choices that determine their QoE, based on available resources. Scaling back on bit-rate can mean avoiding a playback freeze that will lower the service quality significantly. This type of systems face similar level of complexity as QoE management systems. The typical approach for designing control strategies is based on heuristic algorithmic solutions. However, due to the high complexity and inevitable changes in the environment these solutions often do not provide the best performance in all conditions. We propose a solution based on 'learning' instead of 'deterministic design'. Our solution relies on a reinforcement learning agent to determine the optimal strategies that results in maximum performance. We find that this approach infers an efficient control strategy in regards to the predefined reward function effectively.

To conclude, this thesis presents a suite of methods and frameworks that address important aspects of QoE in multimedia services. Our approach is heavily influenced by the paradigm of continuous learning and adaptability, rather than deterministic design. In light of the growing complexity and rapid evolution of multimedia services, computational intelligence methods offer a promising direction.

Future challenges in adopting this approach still remain. Difference scaling methods offer important benefits for subjective evaluation, but methods for measuring the subjective effects of many factors are still missing. Furthermore, the integration of these models into commercial management systems requires a shift in the management paradigm from 'How good is the QoE?' to 'How much will resource X improve the QoE?'. Even though the difference scaling methods lack the 'directness' of the traditional rating,

they more than make up for it with the assessment accuracy.

Predictive capabilities, enabled by RL, improve the performance of active control agents because they allow for better anticipation of changes in the environment. However, if RL agents are left to continuously learn and adapt to the local environment they will develop unique strategies. This non-uniformity in deployed products is unusual for service providers. For further adoption of this type of approach an evaluation of long-term performance and stability needs to be evaluated.

As with any new approach, certain level of maturity is necessary for a wide-spread adoption. Nevertheless, as the current management challenges arising from the complexity and the diversity continue to grow the continuous learning approach presents a valid direction, not only for video enabled services, but also for other multimedia services.

# List of Figures

1.1	A feedback loop in a control system . . . . .	3
1.2	Control loop in a multimedia system . . . . .	4
2.1	Snapshots from 4 of the used videos. Starting from top left image in clockwise order: river bed, park run, sunflower, mobile and calendar. . . . .	29
2.2	PSNR calculated quality degradation from CBR compression over different bit-rates . . . . .	31
2.3	SSIM calculated quality degradation from CBR compression over different bit-rates. . . . .	32
2.4	VQM calculated quality degradation from CBR compression over different bit-rates . . . . .	32
2.5	MOVIE calculated quality degradation from CBR compression over different bit-rates . . . . .	33
2.6	Spatial and temporal information of the videos part of the objective VQA . . . . .	34
2.7	Scene complexity and level of motion in the videos part of the objective VQA . . . . .	34
3.1	Subjective Variability [72] . . . . .	40
3.2	Psychometric function . . . . .	44
3.3	Signal of 1 unit superimposed over noise with 0 mean and standard deviation of 1 . . . . .	45
3.4	The shaded area corresponds to the probability that the signal is positive . . . . .	46
3.5	The shaded area corresponds to the probability that the signal is negative . . . . .	47
3.6	Video display layout in MLDS Application . . . . .	48
3.7	Results of the MLDS experiment by video type . . . . .	49
3.8	Standard error of the MLDS results by video type . . . . .	50
3.9	Fitting a cumulative Gaussian on the 'bs' MLDS model . . . . .	50
3.10	Fitted psychometric curves from the MLDS results . . . . .	51
3.11	Monotonicity of the psychometric curve . . . . .	53

3.12	If first pair in T1 is bigger then first pair of T2 is bigger as well . . . . .	53
3.13	If second pair in T1 is bigger then second pair of T2 is bigger as well . . . . .	54
3.14	Accuracy of the estimations . . . . .	57
3.15	Mean RMSE for the ten types of video . . . . .	58
3.16	Standard deviation of the RMSE for the three types of video . . . . .	59
3.17	Estimation confidences for the three types of videos over the number of introduced data-points . . . . .	60
3.18	Information gain at each step in the experiment . . . . .	61
4.1	Components of a video streaming system . . . . .	66
4.2	The QoE Loop architecture . . . . .	68
4.3	System architecture of the QoE Monitoring framework . . . . .	69
4.4	Decision Tree . . . . .	73
4.5	Decision Tree for the Mobile dataset . . . . .	77
4.6	Decision Tree for the PDA dataset . . . . .	79
4.7	Decision Tree for the PDA dataset . . . . .	80
4.8	SVM hyperplane for the Mobile dataset . . . . .	80
4.9	SVM hyperplane for the PDA dataset . . . . .	81
4.10	SVM hyperplane for the Laptop dataset . . . . .	81
4.11	Performance of the DT and SVM models on the three datasets . . . . .	81
4.12	Subjective study questionnaire . . . . .	82
4.13	Prediction accuracy with and without output aggregation . . . . .	84
4.14	Confusion Matrix for high accuracy with tolerance $\pm 1$ . . . . .	85
4.15	Hoeffding Option Tree results . . . . .	90
4.16	OzaBagging Hoeffding Option Tree results . . . . .	91
4.17	Results of the Hoeffding Option Tree with concept drift experiment . . . . .	92
4.18	OzaBagging Hoeffding Option Tree with concept drift results . . . . .	92
4.19	Comparison with standard ML algorithms . . . . .	93
4.20	Simple decision tree in 2D space . . . . .	95
4.21	Hyper-region building algorithm . . . . .	96
4.22	Change vector remedies for an example instance . . . . .	98
4.23	DT built from the IPTV subjective feedback . . . . .	100
5.1	QoE active control framework . . . . .	104
5.2	HTTP Adaptive Streaming architecture . . . . .	105
5.3	Test bed for adaptive streaming client evaluation . . . . .	108

---

5.4	Adaptive client behaviour with no throttling . . . . .	108
5.5	Adaptive client behaviour with $1Mb/s$ limit . . . . .	109
5.6	Adaptive client behaviour with changing conditions . . . . .	110
5.7	Reinforcement Learning framework . . . . .	113
5.8	The client application architecture . . . . .	119
5.9	Relative impairment from a freeze . . . . .	122
5.10	Relative impairment from quality change . . . . .	123
5.11	Performance of typical filters . . . . .	125
5.12	Simulated background traffic with $\lambda = 0.1$ and $\lambda = 0.5$ with <i>hurst</i> = 0.7 . . . . .	128
5.13	Performance of the RL intelligent agent during training . .	129



# Bibliography

- [1] S. Zeki, “A vision of the brain,” 1993.
- [2] W. Hendee and P. Wells, *The perception of visual information*. Springer Verlag, 1997.
- [3] R. Tannenbaum, “Theoretical foundations of multimedia,” *Ubiquity*, vol. 2000, no. August, p. 2, 2000.
- [4] “Quality - definition and more from the free merriam-webster dictionary,” <http://www.merriam-webster.com/dictionary/quality>. [Online]. Available: <http://www.merriam-webster.com/dictionary/quality>
- [5] N. Kano, N. Seraku, F. Takahashi, and S. Tsuji, “Attractive quality and must-be quality,” *The Journal of the Japanese Society for Quality Control*, vol. 14, no. 2, pp. 39–48, 1984.
- [6] D. Hoyle, *ISO 9000 Quality Systems Handbook-updated for the ISO 9001: 2008 standard*. Routledge, 2012.
- [7] Patrick Le Callet, Sebastian Möller, Andrew Perkis, and eds., “Qualinet white paper on definitions of quality of experience,” European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), Lausanne, Switzerland, Tech. Rep. 1.1, Jun. 2012.
- [8] R. Jain, “Quality of experience,” *Multimedia, IEEE*, vol. 11, no. 1, pp. 96–95, 2004.
- [9] A. Takahashi, D. Hands, and V. Barriac, “Standardization activities in the itu for a qoe assessment of iptv,” *Communications Magazine, IEEE*, vol. 46, no. 2, pp. 78–84, 2008.
- [10] J. Katz, *Clinical Audiology*. Williams & Wilkins, 2002.
- [11] C. Owsley, R. Sekuler, and D. Siemsen, “Contrast sensitivity throughout adulthood,” *Vision research*, vol. 23, no. 7, pp. 689–699, 1983.

- [12] S. Winkler, *Digital video quality*. Wiley, 2005.
- [13] J. Yang and W. Makous, "Implicit masking constrained by spatial inhomogeneities," *Vision research*, vol. 37, no. 14, pp. 1917–1927, 1997.
- [14] M. Rabbani and P. Jones, "Digital image compression techniques." SPIE-International Society for Optical Engineering, 1991.
- [15] S. Kanumuri, P. Cosman, A. Reibman, and V. Vaishampayan, "Modeling packet-loss visibility in mpeg-2 video," *Multimedia, IEEE Transactions on*, vol. 8, no. 2, pp. 341–355, 2006.
- [16] A. Reibman and V. Vaishampayan, "Quality monitoring for compressed video subjected to packet loss," in *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, vol. 1. IEEE, 2003, pp. I–17.
- [17] M. Wada, "Selective recovery of video packet loss using error concealment," *Selected Areas in Communications, IEEE Journal on*, vol. 7, no. 5, pp. 807–814, 1989.
- [18] W. Tan and A. Zakhor, "Real-time internet video using error resilient scalable compression and tcp-friendly transport protocol," *Multimedia, IEEE Transactions on*, vol. 1, no. 2, pp. 172–186, 1999.
- [19] M. Zink, O. Künzel, J. Schmitt, and R. Steinmetz, "Subjective impression of variations in layer encoded videos," *Quality of Service I-WQoS 2003*, pp. 155–155, 2003.
- [20] K. Tan, M. Ghanbari, and D. Pearson, "An objective measurement tool for mpeg video quality," *Signal Processing*, vol. 70, no. 3, pp. 279–294, 1998.
- [21] R. Steinmetz, "Human perception of jitter and media synchronization," *Selected Areas in Communications, IEEE Journal on*, vol. 14, no. 1, pp. 61–72, 1996.
- [22] L. Mued, B. Lines, S. Furnell, and P. Reynolds, "The effects of lip synchronization in ip conferencing," in *Visual Information Engineering, 2003. VIE 2003. International Conference on*. IET, 2003, pp. 210–213.

- [23] J. Hawkins and S. Blakeslee, *On intelligence*. St. Martin's Griffin, 2005.
- [24] V. Menkovski, G. Exarchakos, and A. Liotta, "The value of relative quality in video delivery," *Journal of Mobile Multimedia*, vol. 7, no. 3, pp. 151–162, 2011.
- [25] —, "Tackling the sheer scale of subjective qoe," in *Mobile Multimedia Communications*. Springer, 2012, pp. 1–15.
- [26] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 23–50, 1998.
- [27] M. Farias, M. Moore, J. Foley, and S. Mitra, "17.2: Detectability and annoyance of synthetic blocky and blurry video artifacts," in *SID Symposium Digest of Technical Papers*, vol. 33, no. 1. Wiley Online Library, 2012, pp. 708–711.
- [28] R. Apteker, J. Fisher, V. Kisimov, and H. Neishlos, "Video acceptability and frame rate," *MultiMedia, IEEE*, vol. 2, no. 3, pp. 32–40, 1995.
- [29] C. Shannon, W. Weaver, R. Blahut, and B. Hajek, *The mathematical theory of communication*. University of Illinois press Urbana, 1949, vol. 117.
- [30] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, 1971.
- [31] L. Superiori, C. Weidmann, and O. Nemethova, "Error detection mechanisms for encoded video streams," *Video and Multimedia Transmissions over Cellular Networks: Analysis, Modelling and Optimization in Live 3G Mobile Networks*, p. 125, 2009.
- [32] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Signal Processing: Image Communication*, vol. 15, no. 1, pp. 77–94, 1999.
- [33] J. Chakareski and B. Girod, "Rate-distortion optimized video streaming over internet packet traces," in *Image Processing, 2005. ICIP*

2005. *IEEE International Conference on*, vol. 2. IEEE, 2005, pp. II–161.
- [34] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, “A subjective study to evaluate video quality assessment algorithms,” in *SPIE Proceedings Human Vision and Electronic Imaging*, vol. 7527. Citeseer, 2010.
- [35] G. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 74–90, 1998.
- [36] S. Winkler and P. Mohandas, “The evolution of video quality measurement: from PSNR to hybrid metrics,” *Broadcasting, IEEE Transactions on*, vol. 54, no. 3, p. 660668, 2008. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4550731](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4550731)
- [37] S. Chikkerur, V. Sundaram, M. Reisslein, and L. Karam, “Objective video quality assessment methods: A classification, review, and performance comparison,” *Broadcasting, IEEE Transactions on*, vol. 57, no. 2, pp. 165–182, 2011.
- [38] V. Menkovski and A. Liott, “QoE for mobile streaming,” in *Mobile Multimedia - User and Technology Perspectives*, D. Tjondronegoro, Ed. InTech, Jan. 2012. [Online]. Available: <http://www.intechopen.com/books/mobile-multimedia-user-and-technology-perspectives/qoe-for-mobile-streaming>
- [39] P. Teo and D. Heeger, “Perceptual image distortion,” in *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, vol. 2. IEEE, 1994, pp. 982–986.
- [40] M. Eckert and A. Bradley, “Perceptual quality metrics applied to still image compression,” *Signal processing*, vol. 70, no. 3, pp. 177–200, 1998.
- [41] A. Eskicioglu and P. Fisher, “Image quality measures and their performance,” *Communications, IEEE Transactions on*, vol. 43, no. 12, pp. 2959–2965, 1995.
- [42] B. Girod, “What’s wrong with mean-squared error?” in *Digital images and human vision*. MIT press, 1993, pp. 207–220.

- [43] Z. Wang, A. Bovik, and L. Lu, "Why is image quality assessment so difficult?" in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, vol. 4. IEEE, 2002, pp. IV–3313.
- [44] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.
- [45] F. Pan, X. Lin, S. Rahardja, K. Lim, Z. Li, D. Wu, and S. Wu, "Fast mode decision algorithm for intraprediction in h. 264/avc video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 7, pp. 813–822, 2005.
- [46] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.
- [47] Z. Wang, L. Lu, and A. Bovik, "Video quality assessment based on structural distortion measurement," *Signal processing: Image communication*, vol. 19, no. 2, pp. 121–132, 2004.
- [48] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *Broadcasting, IEEE Transactions on*, vol. 50, no. 3, pp. 312–322, 2004.
- [49] K. Seshadrinathan and A. Bovik, "Motion-based perceptual quality assessment of video," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 72 400X–72 400X.
- [50] R. Mehrotra, K. Namuduri, and N. Ranganathan, "Gabor filter-based edge detection," *Pattern Recognition*, vol. 25, no. 12, pp. 1479–1494, 1992.
- [51] I. Gunawan and M. Ghanbari, "Reduced-reference video quality assessment using discriminative local harmonic strength with motion consideration," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 1, pp. 71–83, 2008.
- [52] L. Ma, S. Li, and K. Ngan, "Reduced-reference video quality assessment of compressed video sequences," 2012.

- [53] D. Taubman, M. Marcellin, and M. Rabbani, "Jpeg2000: Image compression fundamentals, standards and practice," *Journal of Electronic Imaging*, vol. 11, no. 2, pp. 286–287, 2002.
- [54] F. Pereira and T. Ebrahimi, *The MPEG-4 book*. Prentice Hall, 2002.
- [55] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Practice and Standards*. Massachusetts: Kluwer Academic Publishers, 2002.
- [56] Z. Liu, L. Karam, and A. Watson, "Jpeg2000 encoding with perceptual distortion control," *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 1763–1778, 2006.
- [57] F. A. Kingdom and P. Whittle, "Contrast discrimination at high contrasts reveals the influence of local light adaptation on contrast processing," *Vision research*, vol. 36, no. 6, pp. 817–829, 1996.
- [58] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3445–3462, 1993.
- [59] J. Li, "Visual progressive coding," in *SPIE proceedings series*. Society of Photo-Optical Instrumentation Engineers, 1998, pp. 1143–1154.
- [60] N. Kamaci, Y. Altunbasak, and R. Mersereau, "Frame bit allocation for the h. 264/avc video coder via cauchy-density-based rate and distortion models," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 8, pp. 994–1006, 2005.
- [61] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and mpeg video coders," *Image Processing, IEEE Transactions on*, vol. 3, no. 5, pp. 533–545, 1994.
- [62] J. Ribas-Corbera and S. Lei, "Rate control in dct video coding for low-delay communications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, no. 1, pp. 172–185, 1999.
- [63] Y. Kim, Z. He, and S. Mitra, "A novel linear source model and a unified rate control algorithm for h. 263/mpeg-2/mpeg-4," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01)*.

- 2001 *IEEE International Conference on*, vol. 3. IEEE, 2001, pp. 1777–1780.
- [64] Z. He, Y. Kim, and S. Mitra, “Low-delay rate control for dct video coding via  $\rho$ -domain source modeling,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 8, pp. 928–940, 2001.
- [65] A. Webster, C. Jones, M. Pinson, S. Voran, and S. Wolf, “An objective video quality assessment system based on human perception,” in *SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, 1993, pp. 15–26.
- [66] N. Kanopoulos, N. Vasanthavada, and R. Baker, “Design of an image edge detection filter using the sobel operator,” *Solid-State Circuits, IEEE Journal of*, vol. 23, no. 2, pp. 358–367, 1988.
- [67] J. Hu and H. Wildfeuer, “Use of content complexity factors in video over ip quality monitoring,” in *Quality of Multimedia Experience, 2009. QoMEX 2009. International Workshop on*. IEEE, 2009, pp. 216–221.
- [68] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, “Study of subjective and objective quality assessment of video,” *Image Processing, IEEE Transactions on*, vol. 19, no. 6, pp. 1427–1441, 2010.
- [69] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the h. 264/avc video coding standard,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, 2003.
- [70] NTIA, “Video quality metric (vqm) software,” <http://www.its.bldrdoc.gov/resources/video-quality-research/software.aspx>, 2012.
- [71] K. Seshadrinathan, R. Soundararajan, A. Bovik, and L. Cormack, “A subjective study to evaluate video quality assessment algorithms,” vol. 7527, 2010.
- [72] V. Menkovski and A. Liotta, “Adaptive psychometric scaling for video quality assessment,” *Signal Processing: Image Communication*, 2012.

- [73] V. Menkovski, G. Exarchakos, and A. Liotta, "The value of relative quality in video delivery," *Journal of Mobile Multimedia*, vol. 7, no. 3, p. 151162, 2011. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2230537>
- [74] F. Agboma and A. Liotta, "Addressing user expectations in mobile content delivery," *Mob. Inf. Syst.*, vol. 3, no. 3,4, p. 153164, Dec. 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1376820.1376823>
- [75] W. H. Ehrenstein and A. Ehrenstein, "Psychophysical methods," in *Modern techniques in neuroscience research*, 1999, pp. 1211–1241. [Online]. Available: <http://uni-leipzig.de/~isp/isp/history/texts/PSYPHY-M.PDF>
- [76] ITU, "500-11, methodology for the subjective assessment of the quality of television pictures, recommendation itu-r bt. 500-11," *ITU Telecom. Standardization Sector of ITU*, 2002.
- [77] F. De Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi, "Subjective assessment of h. 264/avc video sequences transmitted over a noisy channel," in *Quality of Multimedia Experience, 2009. QoMEX 2009. International Workshop on*. IEEE, 2009, pp. 204–209.
- [78] Q. Huynh-Thu and M. Ghanbari, "Modelling of spatio-temporal interaction for video quality assessment," *Signal Processing: Image Communication*, vol. 25, no. 7, pp. 535–546, 2010.
- [79] I. Recommendation, "910, subjective video quality assessment methods for multimedia applications, recommendation itu-t p. 910," *ITU Telecom. Standardization Sector of ITU*, 1999.
- [80] M. Barkowsky, M. Pinson, R. P  pion, and P. Le Callet, "Analysis of freely available subjective dataset for hdtv including coding and transmission distortions," in *Fifth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-10)*, 2010.
- [81] S. Stevens, "On the psychophysical law." *Psychological review*, vol. 64, no. 3, p. 153, 1957.



- [82] D. Krantz, "A theory of context effects based on cross-context matching," *Journal of Mathematical Psychology*, vol. 5, no. 1, pp. 1–48, 1968.
- [83] ———, "A theory of magnitude estimation and cross-modality matching," *Journal of Mathematical Psychology*, vol. 9, no. 2, pp. 168–199, 1972.
- [84] D. Krantz, R. Luce, P. Suppes, and A. Tversky, *Foundations of measurement volume I: additive and polynomial representations*. Dover Publications, 2006, vol. 1.
- [85] R. Shepard, "On the status of 'direct' psychophysical measurement," *Minnesota studies in the philosophy of science*, vol. 9, pp. 441–490, 1978.
- [86] ———, "Psychological relations and psychophysical scales: On the status of direct psychophysical measurement," *Journal of Mathematical Psychology*, vol. 24, no. 1, pp. 21–57, 1981.
- [87] A. Watson, "Proposal: Measurement of a jnd scale for video quality," *IEEE G-2.1. 6 Subcommittee on Video Compression Measurements*, 2000.
- [88] S. Winkler, "On the properties of subjective ratings in video quality experiments," in *Quality of Multimedia Experience, 2009. QoMEx 2009. International Workshop on*. IEEE, 2009, pp. 139–144.
- [89] V. Q. E. Group *et al.*, "Report on the validation of video quality models for high definition video content," Technical report, Tech. Rep., 2010.
- [90] P. Corriveau, C. Gojmerac, B. Hughes, and L. Stelmach, "All subjective scales are not created equal: The effects of context on different scales," *Signal processing*, vol. 77, no. 1, pp. 1–9, 1999.
- [91] P. Brooks and B. Hestnes, "User measures of quality of experience: why being objective and quantitative is important," *Network, IEEE*, vol. 24, no. 2, pp. 8–13, 2010.
- [92] G. T. Fechner, *Elements of psychophysics*. Holt, Rinehart and Winston, 1966.

- [93] J. D. McCarthy, M. A. Sasse, and D. Miras, “Sharp or smooth?: comparing the effects of quantization vs. frame rate for streamed video,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '04. New York, NY, USA: ACM, 2004, p. 535542. [Online]. Available: <http://doi.acm.org/10.1145/985692.985760>
- [94] E. H. Weber, *E.H. Weber On The Tactile Senses*. Psychology Press, Jan. 1996.
- [95] A. Watson and L. Kreslake, “Measurement of visual impairment scales for digital video,” in *Human Vision, Visual Processing, and Digital Display, Proc. SPIE*, vol. 4299, 2001.
- [96] L. L. Thurstone, *The measurement of values*. Oxford, England: Univer. Chicago Press, 1959, vol. vii.
- [97] L. Maloney and J. Yang, “Maximum likelihood difference scaling,” *Journal of Vision*, vol. 3, no. 8, 2003.
- [98] C. Charrier, L. Maloney, H. Cherifi, and K. Knoblauch, “Maximum likelihood difference scaling of image quality in compression-degraded images,” *JOSA A*, vol. 24, no. 11, pp. 3418–3426, 2007.
- [99] K. Knoblauch, L. Maloney *et al.*, “Mlds: Maximum likelihood difference scaling in r,” *Journal of Statistical Software*, vol. 25, no. 2, pp. 1–26, 2008.
- [100] D. Green, J. Swets *et al.*, *Signal detection theory and psychophysics*. Wiley New York, 1966, vol. 1974.
- [101] H. Bergsten, *JavaServer Pages 3rd*. United States: OReilly Media, Inc, 2003.
- [102] “JW player: Overview | LongTail video | home of the JW player,” <http://www.longtailvideo.com/players>. [Online]. Available: <http://www.longtailvideo.com/players>
- [103] L. Aimar, L. Merritt, E. Petit, M. Chen, J. Clay, M. Rullgrd, C. Heine, and A. Izvorski, “VideoLAN - x264, the best H.264/AVC encoder,” <http://www.videolan.org/developers/x264.html>. [Online]. Available: <http://www.videolan.org/developers/x264.html>

- [104] R. Team *et al.*, “R: A language and environment for statistical computing,” *R Foundation Statistical Computing*, 2008.
- [105] K. Knoblauch, “Psyphy: Functions for analyzing psychophysical data in r,” *R package version 0.0-5*, URL <http://CRAN.R-project.org/package=psyphy>, 2007.
- [106] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap (Chapman & Hall/CRC Monographs on Statistics & Applied Probability)*. Chapman and Hall/CRC, 1994.
- [107] F. Wichmann and N. Hill, “The psychometric function: I. fitting, sampling, and goodness of fit,” *Attention, Perception, & Psychophysics*, vol. 63, no. 8, pp. 1293–1313, 2001.
- [108] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951. [Online]. Available: <http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.aoms/1177729694>
- [109] K. Eckschlagner and K. Danzer, *Information Theory in Analytical Chemistry*. John Wiley & Sons, May 1994.
- [110] V. Menkovski, G. Exarchakos, A. Liotta, and A. Sánchez, “Measuring quality of experience on a commercial mobile tv platform,” in *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*. IEEE, 2010, pp. 33–38.
- [111] M. Mu, A. Mauthe, and F. Garcia, “A utility-based qos model for emerging multimedia applications,” in *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST’08. The Second International Conference on*. IEEE, 2008, pp. 521–528.
- [112] M. Mu, R. Gostner, A. Mauthe, G. Tyson, and F. Garcia, “Visibility of individual packet loss on h. 264 encoded video stream—a user study on the impact of packet loss on perceived video quality,” 2009.
- [113] J. Kim, T. Um, W. Ryu, and B. Lee, “Iptv systems, standards and architectures: Part ii-heterogeneous networks and terminal-aware qos/qoe-guaranteed mobile iptv service,” *Communications Magazine, IEEE*, vol. 46, no. 5, pp. 110–117, 2008.

- [114] M. Volk, J. Sterle, U. Sedlar, and A. Kos, "An approach to modeling and control of qoe in next generation networks [next generation telco it architectures]," *Communications Magazine, IEEE*, vol. 48, no. 8, pp. 126–135, 2010.
- [115] S. Esaki, A. Kurokawa, and K. Matsumoto, "Overview of the next generation network," *NTT Technical Review*, vol. 5, no. 6, 2007.
- [116] J. Zhang and N. Ansari, "On assuring end-to-end qoe in next generation networks: challenges and a possible solution," *Communications Magazine, IEEE*, vol. 49, no. 7, pp. 185–191, 2011.
- [117] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol–http/1.1," 1999.
- [118] W. Stevens and G. Wright, *TCP/IP Illustrated: the protocols*. Addison-Wesley Professional, 1994, vol. 1.
- [119] T. Stockhammer, P. Fröjdh, I. Sodagar, and S. Rhyu, "Information technologympeg systems technologiespart 6: Dynamic adaptive streaming over http (dash)," *ISO/IEC, MPEG Draft International Standard*, 2011.
- [120] H. Schulzrinne, "Real time streaming protocol (rtsp)," 1998.
- [121] J. Postel, "User datagram protocol," *Isi*, 1980.
- [122] P. Frossard, "Fec performance in multimedia streaming," *Communications Letters, IEEE*, vol. 5, no. 3, pp. 122–124, 2001.
- [123] V. Menkovski, G. Exarchakos, A. Liotta, and A. Cuadra-Sánchez, "Managing quality of experience on a commercial mobile tv platform," *International Journal On Advances in Telecommunications*, vol. 4, no. 1 and 2, pp. 72–81, 2011.
- [124] "TM forum - TM forum IPDR program," <http://www.tmforum.org/InDepth/10344/home.html>. [Online]. Available: <http://www.tmforum.org/InDepth/10344/home.html>
- [125] A. Cuadra-Sanchez and C. Casas-Caballero, "End-to-end quality of service monitoring in convergent iptv platforms," in *Next Generation Mobile Applications, Services and Technologies, 2009. NGMAST'09. Third International Conference on*. IEEE, 2009, pp. 303–308.

- [126] M. Karam and F. Tobagi, "Analysis of the delay and jitter of voice traffic over the internet," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. IEEE, 2001, pp. 824–833.
- [127] E. Alpaydin, *Introduction to machine learning*. MIT press, 2004.
- [128] J. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [129] —, *C4. 5: programs for machine learning*. Morgan kaufmann, 1993, vol. 1.
- [130] V. Vapnik and S. Kotz, *Estimation of dependences based on empirical data*. Springer-Verlag New York, 1982, vol. 41.
- [131] J. Platt *et al.*, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.
- [132] R. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [133] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [134] Y. Freund, R. Schapire *et al.*, "Experiments with a new boosting algorithm," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*. MORGAN KAUFMANN PUBLISHERS, INC., 1996, pp. 148–156.
- [135] F. Agboma and A. Liotta, "QoE-aware QoS management," in *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, ser. MoMM '08. New York, NY, USA: ACM, 2008, p. 111116. [Online]. Available: <http://doi.acm.org/10.1145/1497185.1497210>
- [136] I. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.
- [137] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International joint Conference on artificial intelligence*, vol. 14. Lawrence Erlbaum Associates Ltd, 1995, pp. 1137–1145.

- [138] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 71–80.
- [139] R. Kohavi and C. Kunz, "Option decision trees with majority votes," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*. Citeseer, 1997, pp. 161–169.
- [140] B. Pfahringer, G. Holmes, and R. Kirkby, "New options for hoeffding trees," *AI 2007: Advances in Artificial Intelligence*, pp. 90–99, 2007.
- [141] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid," in *Proceedings of the second international conference on knowledge discovery and data mining*, vol. 7, 1996.
- [142] J. Gama, R. Rocha, and P. Medas, "Accurate decision trees for mining high-speed data streams," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 523–528.
- [143] G. Holmes, R. Kirkby, and B. Pfahringer, "Stress-testing hoeffding trees," *Knowledge Discovery in Databases: PKDD 2005*, pp. 495–502, 2005.
- [144] N. Oza, "Online bagging and boosting," in *Systems, man and cybernetics, 2005 IEEE international conference on*, vol. 3. IEEE, 2005, pp. 2340–2345.
- [145] N. Oza and S. Russell, "Experimental comparisons of online and batch versions of bagging and boosting," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001, pp. 359–364.
- [146] A. Bifet, G. Holmes, B. Pfahringer, R. Kirkby, and R. Gavaldà, "New ensemble methods for evolving data streams," 2009.
- [147] V. Menkovski, G. Exarchakos, A. Liotta, and A. Sánchez, "Estimations and remedies for quality of experience in multimedia streaming," in *Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services (CENTRIC), 2010 Third International Conference on*. IEEE, 2010, pp. 11–15.

- [148] V. Menkovski, A. Oredope, A. Liotta, and A. Sánchez, “Predicting quality of experience in multimedia streaming,” in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia*. ACM, 2009, pp. 52–59.
- [149] M. Puterman, *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, Inc., 1994.
- [150] “ISO/IEC 23009-1:2012 - information technology – dynamic adaptive streaming over HTTP (DASH) – part 1: Media presentation description and segment formats.” [Online]. Available: [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=57623](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623)
- [151] G. Van der Auwera, P. David, M. Reisslein, and L. Karam, “Traffic and quality characterization of the h. 264/avc scalable video coding extension,” *Advances in Multimedia*, vol. 2008, no. 2, p. 1, 2008.
- [152] I. Sodagar, “The mpeg-dash standard for multimedia streaming over the internet,” *MultiMedia, IEEE*, vol. 18, no. 4, pp. 62–67, 2011.
- [153] F. Bertone, V. Menkovski, and A. Liotta, “Adaptive P2P streaming,” in *Streaming Media with Peer-to-Peer Networks*. IGI Global, 2012, pp. 52–73. [Online]. Available: <http://www.igi-global.com/chapter/adaptive-p2p-streaming/66305>
- [154] S. Akhshabi, A. Begen, and C. Dovrolis, “An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http,” *ACM MMSys*, vol. 11, pp. 157–168, 2011.
- [155] C. Müller, S. Lederer, and C. Timmerer, “An evaluation of dynamic adaptive streaming over http in vehicular environments,” in *Proceedings of the 4th Workshop on Mobile Video*. ACM, 2012, pp. 37–42.
- [156] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. Cambridge Univ Press, 1998, vol. 1, no. 1.
- [157] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2001.
- [158] J. Hamilton, *Time series analysis*. Cambridge Univ Press, 1994, vol. 2.

- [159] E. Goldoni, G. Rossi, and P. Gamba, "Improving available bandwidth estimation using averaging filtering techniques," *Rapport technique, Università degli Studi di Pavia, Laboratorio Reti*, 2008.
- [160] G. Barnard, "Control charts and stochastic processes," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 239–271, 1959.
- [161] L. Burgstahler and M. Neubauer, "New modifications of the exponential moving average algorithm for bandwidth estimation," in *Proc. of the 15th ITC Specialist Seminar*, 2002.
- [162] M. Kim and B. Noble, "Mobile network estimation," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 298–309.
- [163] Z. Sun, D. He, L. Liang, and H. Cruickshank, "Internet qos and traffic modelling," in *Software, IEE Proceedings-*, vol. 151, no. 5. IET, 2004, pp. 248–255.
- [164] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *Networking, IEEE/ACM Transactions on*, vol. 5, no. 6, pp. 835–846, 1997.
- [165] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of ethernet traffic," in *ACM SIGCOMM Computer Communication Review*, vol. 23, no. 4. ACM, 1993, pp. 183–193.
- [166] M. Zukerman, T. Neame, and R. Addie, "Internet traffic modeling and future technology implications," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1. IEEE, 2003, pp. 587–596.
- [167] V. Paxson and S. Floyd, "Wide area traffic: the failure of poisson modeling," *IEEE/ACM Transactions on Networking (ToN)*, vol. 3, no. 3, pp. 226–244, 1995.





# List of Publications

## Journal articles

- Menkovski, V. & Liotta, A. (2012). Adaptive psychometric scaling for video quality assessment. *Signal Processing: Image Communication*, Available online 11 February 2012, ISSN 0923-5965.
- Menkovski, V., Exarchakos, G., Cuadra-Snchez, A. & Liotta, A. (2011). A quality of experience management module. *International Journal on Advances in Intelligent Systems*, 4(1-2), 13-19.
- Menkovski, V., Exarchakos, G., Liotta, A. & Cuadra Sanchez, A. (2010). Quality of experience models for multimedia streaming. *International Journal of Mobile Computing and Multimedia Communications*, 2(4), 1-20.
- Menkovski, V., Exarchakos, G. & Liotta, A. (2011). The value of relative quality in video delivery. *Journal of Mobile Multimedia*, 7(3), 151-162.

## Book chapters

- Bertone, F., Menkovski, V. & Liotta, A. (2012). Adaptive P2P streaming. In M. Fleury & N. Qadri (Eds.), *Streaming media with peer-to-peer networks : wireless perspectives*. Idea Group Publishing.
- Menkovski, V. & Liotta, A. (2012). QoE for mobile streaming. In Dian Tjondronegoro (Ed.), *Mobile Multimedia*. InTech Publishing.

## Reports

- Menkovski, V., Exarchakos, G. & Liotta, A. (2011). The value of relative quality in video delivery. Eindhoven: Technische Universiteit Eindhoven.

## Proceedings and Conference Contributions

- Menkovski, V. & Liotta, A. (2013). Intelligent control for adaptive video streaming. In Proceedings of the 2013 International conference on consumer electronics. (ICCE'13, Las Vegas, USA, January 11-14, 2013). IEEE (Accepted)
- Antonio Liotta, Luca Druda, Vlado Menkovski and Georgios Exarchakos (2012). Quality of Experience Management for Video Streams: the case of Skype Quality of Experience Management for Video Streams: the case of Skype. In Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia (MoMM'12)
- Menkovski, V., Exarchakos, G. & Liotta, A. (2012). Tackling the sheer scale of subjective QoE. In L. Atzoni, J. Delgado & D.D. Giusto (Eds.), Mobile Multimedia Communications (7th International ICST Conference, MOBIMEDIA 2011, Cagliari, Italy, September 5-7, 2011, Revised Selected Papers) Vol. 79. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (pp. 1-15). Berlin: Springer.
- Okyere-Benya, J., Aldiabat, M.A.H.B., Menkovski, V., Exarchakos, G. & Liotta, A. (2012). Video quality degradation on IPTV networks. In Proceedings of the 2012 International Conference on Computing, Networking and Communications (ICNC'12, Maui HI, USA, January 30-February 2, 2012) (pp. 702-707). IEEE.
- Exarchakos, G., Menkovski, V. & Liotta, A. (2011). Can Skype be used beyond video calling? In 9th International Conference on Advances in Mobile Computing and Multimedia (MoMM'11, Ho Chi Minh City, Vietnam, December 5-7, 2011) (pp. 155-161). New York NY: ACM.
- Menkovski, V., Exarchakos, G. & Liotta, A. (2011). Adaptive testing for video quality assessment. In M.J. Damsio, G. Cardoso, C. Quico & D. Geerts (Eds.), Adjunct Proceedings of EuroTV 2011 (Lisbon, Portugal, June 29-July 1, 2011) (pp. 128-131). Lisbon: COFAC/Universidade Lusfona de Humanidades e Tecnologias.
- Menkovski, V., Exarchakos, G., Liotta, A. & Cuadra Sanchez, A. (2010). Estimations and remedies for quality of experience in multimedia streaming. In Proceedings Third International Conference on Advances in Human-Oriented and Personalized Mechanisms, Technologies and Services (CENTRIC 2010, August 22-27, 2010, Nice,

France) (pp. 11-15). Piscataway: IEEE.

- Menkovski, V., Exarchakos, G. & Liotta, A. (2010). Machine learning approach for quality of experience aware networks. In Proceedings 2nd International Conference on Intelligent Networking and Collaborative Systems (INCOS'10, Thessaloniki, Greece, November 24-26, 2010) (pp. 461-466). IEEE.
- Menkovski, V., Exarchakos, G., Liotta, A. & Cuadra Sanchez, A. (2010). Measuring quality of experience on a commercial mobile TV platform. In Proceedings of the 2nd International Conference on Advances in Multimedia (MMEDIA, Athens, Greece, June 13-19, 2010) (pp. 33-38). IEEE.
- Menkovski, V., Exarchakos, G. & Liotta, A. (2010). Online learning for QoE management. In Informal proceedings of the 19th Annual Machine Learning Conference of Belgium and The Netherlands (Benelearn'10, Leuven, Belgium, May 27-28, 2010) (pp. 1-6).
- Menkovski, V., Exarchakos, G. & Liotta, A. (2010). Online QoE prediction. In Proceedings of the 2nd International Workshop on Quality of Multimedia Experience (QoMEX), June 21-23, Trondheim, Norway (pp. 118-123). Piscataway: IEEE.
- Menkovski, V., Oredope, A., Liotta, A. & Cuadra Sanchez, A. (2009). Optimized online learning for QoE prediction. In T. Calders, K. Tuyls & M. Pechenizkiy (Eds.), Proceedings 21st Benelux Conference on Artificial Intelligence (BNAIC'09, Eindhoven, The Netherlands, October 29-30, 2009) (pp. 169-176).
- Menkovski, V., Oredope, A., Liotta, A. & Cuadra Sanchez, A. (2009). Predicting quality of experience in multimedia streaming. In Proceedings 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM2009, Kuala Lumpur, Malaysia, December 14-16, 2009) (pp. 1-8). ACM.
- Oredope, A., Exarchakos, G., Menkovski, V. & Liotta, A. (2009). An analysis of mobile signalling in converged networks. In Proceedings of the 9th IEEE Malaysia International Conference on Communications (MICC 2009, Kuala Lumpur, Malaysia, December 15-17, 2009) (pp. 629-634). Piscataway: IEEE Service Center.

## Patents

- Cuadra-Snchez, A., Mar Cutanda Rodrguez, M. del, Liotta, A. & Menkovski, V. (05-14-2010). Methodology for calculating perception of the user experience of the quality of monitored integrated telecommunications operator services. no WO2011141586.

# Awards

- Best paper award, 2012: Antonio Liotta, Luca Druda, Vlado Menkovski and Georgios Exarchakos (2012). Quality of Experience Management for Video Streams: the case of Skype Quality of Experience Management for Video Streams: the case of Skype. In Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia (MoMM'12)
- COST Training School Grant. COST Training School on 3D Media, UX and Computational Architectures (3DMUX 2012) August 2012.
- Best Student Award. 2012 Edition of the Erasmus Intensive Programme "Multimedia and the Future Internet: Moving Social and Mobile"
- Best Group Project award - 'The future of TV'. 2012 Edition of the Erasmus Intensive Programme "Multimedia and the Future Internet: Moving Social and Mobile"
- Erasmus Training School Grant. Erasmus Intensive Programme "Multimedia and the Future Internet: Moving Social and Mobile".
- Best paper award, 2011: Georgios Exarchakos, Vlado Menkovski, Antonio Liotta, Can Skype be used beyond video calling? MoMM'11. Vietnam. November 2011. (awarded by International Journal of Pervasive Computing and Communications, Emerald Group Publishing Limited)
- Best paper award, 2010: V. Menkovski, G. Exarchakos, A. Liotta, A. Cuadra Sanchez, Estimations and Remedies for Quality of Experience in Multimedia Streaming. CENTRIC'10. France. August 2010.
- Best paper award, 2010: V. Menkovski, G. Exarchakos, A. Liotta, A. Cuadra Sanchez, Measuring Quality of Experience on a Commercial Mobile TV Platform. MMEDIA'10. Greece. June 2010.



# Curriculum Vitæ

Vlado Menkovski was born on 14th of February 1979 in Skopje, Macedonia (part of former Yugoslavia in 1979). He studied Electrical Engineering at Ss. Cyril and Methodius University in Skopje, Macedonia and obtained the degree Graduated Engineer (5 years) in 2006. He continued his studies in Information Networking and obtained a Master of Science degree from Carnegie Mellon University, Pittsburgh, USA in 2008.

During the period of 2002 to 2005 he worked as a Software Engineer in Netcetera, Skopje, Macedonia. Between 2005 and 2006 he co-founded and worked as a CTO in the start-up Synapse, Skopje, Macedonia. In the second half of 2007 until early 2008 he worked as a Research Assistant in AIT, Athens, Greece. Next, between mid-2008 and mid-2009 he worked as a Research Engineer in Innovaworks, Skopje, Macedonia.

In 2009 he started the PhD programme at the Electrical Engineering department of Eindhoven University of Technology, Eindhoven, The Netherlands. The results of the work during the PhD programme are presented in this thesis.





