eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

PhD thesis

Measuring Quality of Video of Internet Protocol Television (IPTV)

Author:

Iñigo Sedano Pérez

Co-directors:

Gorka Prieto (UPV/EHU)

Maria Kihl (University of Lund)

PhD program:

Information technologies and communications in mobile networks

Bilbao, 2020

Acknowledgments

This thesis required a persistent effort during many years and was possible with the support of my family. My deceased father Javier helped me greatly and pushed me towards the goal, together with my brother Ibon, my mother Elena and all my friends.

I am very grateful to Dr. Gorka Prieto (thesis director from University of the Basque Country) and Dr. Maria Kihl (thesis director from Lund University) by their patience and dedication. Dr. Kjell Brunnström from RISE Research Institutes of Sweden AB in Sweden contributed giving me expert advice and hosting me for a period of one year and half in their institution (RISE Acreo). I would like also to thank Dr. Marcus Barkowsky for his help during my short stage at IRCCyN (University of Nantes).

Finally, Tecnalia Research & Innovation, the company where I currently work, supported me with time and funding.

To all of you, thank you very much!!

Index

Index of figures

Index of tables

Chapter 1
Introduction

1.1 Context

1.1.1 Interest in video quality metrics

The development of high-speed access networks has enabled a variety of video delivery alternatives over the Internet, for example IPTV. A major problem is that the Quality of Experience (QoE) of video can be severely affected even by a low packet loss rate. Consequently, it is important to allocate necessary network resources to minimize the loss of video information. For doing that it is necessary to monitor and estimate the QoE delivered to the user. Accordingly, the development of video quality metrics is receiving large attention in the research community [1][2][3].

For the purpose of clarifying some concepts here we explain the meaning of metric. An objective video quality metric tries to approximate the subjective perception of the user evaluating parameters related to a video [4]. The objective metrics can be classified according to the need of information from the original undistorted video. There are three different types of objective metrics: no-reference metrics, reduced-reference metrics and full-reference metrics. The no-reference metrics are those metrics in which only the distorted video is used to compute the quality of the video. In the reduced-reference metric, the original undistorted video itself is not available, but some parameters related to it are available. In the full-reference video both the original undistorted video and the degraded video are available to the objective metric that computes the video quality. The strict mathematical definition of metrics is not valid for video quality measures but this has been the usual term in literature.

The subjective tests involving a panel of observers constitute the most accurate mean of measuring the quality of a video. However, it requires careful preparation of viewing conditions, it is very time consuming as many people are involved and cannot be applied in real-time monitoring of video in IPTV networks. This fact has driven many researchers to develop objective metrics. At the beginning the focus was on full-reference metrics and shifted gradually to reduced-reference metrics and finally to no-reference metrics [5][6].

1.1.2 Stay in Sweden and collaboration with the Video Quality Experts Group

The author of this thesis, Iñigo Sedano stayed in Sweden in a research institute called RISE Acreo (Research Institutes of Sweden - Acreo) [7] since November 2009 to April 2011. RISE Acreo is a Swedish research institute within electronics, optics and communication technologies. The funding during the visit was provided by an Etortek grant from *Fundación Centros Tecnológicos Iñaki Goenaga* [8]. The purpose of the grant was to get specialized in video quality monitoring by getting knowledge from specialized research entities from other countries. During that period in Sweden, the thesis directors (Maria Kihl from University of Lund [9], together with Gorka Prieto from University of the Basque Country [10]) monitored the thesis activities. In fact, physical meetings took place each two weeks with Maria Kihl. In addition, Kjell Brunnström from Acreo research institute (based on Stockholm), as an expert integrated in the Video Quality Experts Group (VQEG [11]) gave advice during all the period.

The author also collaborated personally with some people of the Joint Effort Group (JEG [12]) of the Video Quality Experts Group (VQEG), in particular with Marcus Barkowsky from University of Nantes [13](France) and Kjell Brunnström from Acreo. The author of this thesis helped the VQEG to implement a model [14][15] for evaluating quality of video based on the Quantization Parameter taking into account compression distortions. In order to collaborate more closely with Marcus, the author also stayed in Nantes (France) at L'Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN [16]) of University of Nantes for one month in June 2010, and implemented the model mentioned during that month.

The close contact with Acreo also led to the publication of a study of user behaviour in a real IPTV network from Sweden (Swedish municipal network) [17]. Currently Iñigo Sedano works in Tecnalia Research & Innovation [18] technology centre in Spain, Bilbao.

1.1.3 Joint Effort Group of VQEG description

Since 1997, the VQEG has been working towards validating competitive objective video quality metrics for different resolutions up to High Definition (HD) [19][20][21] which has resulted in a number of international standards such as ITU-T J.144 (Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference) [22], ITU-T J.246 (Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference) [23] and ITU-T J.247 (Objective perceptual multimedia video quality measurement in the presence of a full reference) [24]. In parallel, VQEG input has led to the adjustment and revisions of some ITU standards on subjective assessment, e.g. ITU-T BT.500 (Methodology for the subjective assessment of the quality of television pictures) [25] and ITU-T P.910 (Subjective video quality assessment methods for multimedia applications) [26].

The strength of VQEG is based on the wide distribution of domains of the participants which ranges from network providers, helping in the selection of typical transmission conditions, to the manufacturers of objective models. This is complemented by the members of the Independent Laboratory Group (ILG) consisting of several academics, research institutions and even private companies. This mixture allows to identify the current requirements of the industry as well as the most promising solutions.

In order to validate new objective video quality metrics, submitted and developed by different proponents individually, several projects were set up which define the overall scope and validity of the candidate metrics.

Example projects include Hybrid Perceptual/Bitstream, Multimedia Phase II and 3DTV. Within each project, a systematic approach is used to validate the submitted objective quality metrics. This includes the formulation of a test plan defining the exact procedure for performing the validation. Each test plan contains the definition of source sequences, typical degradations by coding and transmission in the context of the scope of application, obtaining several subjective test databases in different cultural environments and performing statistical analysis on the results.

In order to guarantee a correct and fair validation workflow, the proponents of models are identified and separated from the ILGs. As a contrast to the other ad-hoc working groups of VQEG, all partners in the Joint Effort Group (JEG) contribute in a collaborative effort to develop the most suitable, most effective and highest quality video quality assessment strategies. The JEG group is organizing frequent telephone conference calls and face-to-face meetings in the context of the VQEG, which are held at least twice a year.

The first goal of the JEG group is to develop an objective video quality model by using only the information available at the receiver side i.e. no-reference, e.g. inside a set-top box. The bit stream is captured at the network layer and the decoded video signal is stored. The bit stream may be provided to the model in a parsed form which simplifies model development [14]. In order to evaluate and train the objective quality metrics, subjective databases are necessary. A toolchain was created by the JEG in order to ease this training database creation process and up to now, several subjectively evaluated video databases are available for testing. As the JEG is an open collaborative approach, the databases are publicly available so that algorithms can be tested and integrated easily into the JEG-Hybrid model.

The VQEG Board is composed of those individuals who have accepted a role to help coordinate an area of work in VQEG. In Table 1 the board is shown including names, positions and organizations.

Table 1. VQEG board

Group	Name and Position	Organization
VQEG	Co-Chair Kjell Brunnström	RISE Research Institute of Sweden AB
	Co-Chair Margaret Pinson	NTIA/ITS
Independent Lab Group (ILG)	Chair Margaret Pinson	NTIA/ITS
	Vice Chair Phil Corriveau	INTEL
5G Key Performance Indicators (5GKPI)	Chair Pablo Perez	Nokia
	Vice Chair Kjell Brunnström	RISE Research Institute of Sweden AB
Audiovisual HD (AVHD)	Chair Shahid Satti	Opticom
	Vice Chair Quan Huynh-Thu	Canon(CiSRA)
	Vice Chair Silvio Borer	Rhode & Schwarz
Computer Graphics Imagery (CGI)	Chair Saman Zadtootaghaj	TU Berlin
	Vice Chair Nabajeet Barman	Kingston University
Human Factors for Visual Experiences (HFVE)	Chair Maria Martini	Kingston University
	Vice Chair (vacant)	
Immersive Media Group	Chair Jesus Gutierrez	University of Nantes
	Vice Chair Zhenzhong Chen	Wuhan University
	Vice Chair Phil Corriveau	INTEL
	Vice Chair Pablo Perez	Nokia
JEG Hybrid	Chair Marcus Barkowsky	Deggendorf Institute of Technology (DIT)
	Vice Chair Glenn Van Wallendael	Ghent University - imec - IDLab
	Vice Chair Enrico Masala	Politecnico di Torino
Joint Effort Group	Kjell Brunnström	RISE Research Institute of Sweden AB
	Patrick Le Callet	University of Nantes
No Reference Metrics (NORM)	Chair Jeremy Bennington	Spirent
	Vice Chair Phil Corriveau	INTEL
	Vice Chair Margaret Pinson	NTIA/ITS
PsyPhyQA	Chair Sebastian Bosse	Heinrich Hertz Institute
	Vice Chair Ulrich Engelke	CSIRO
	Vice Chair Naeem	University of the West of

	Ramzan	Scotland
Quality Assessment for Computer Vision Applications (QACoViA)	Chair Mikolaj Leszczuk	AGH University
	Chair Pablo Perez	Nokia
Statistical Analysis Methods	Chair Lucjan Janowski	AGH University
	Vice Chair Ioannis Katsavounidis	Facebook
	Vice Chair Zhi Li	Netflix
	Vice Chair Patrick Le Callet	University of Nantes
Tools and Subjective Labs Setup Co-Chair	Chair Glenn Van Wallendael	Ghent University - imec - IDLab
	Vice Chair Bert Vankeirsbilck	Ghent University-iMinds-IBCN

1.2 Objectives

1.2.1 Main objectives

The broadcasting TV is nowadays very popular and moves a very significant amount of money that comes mainly from film productions, advertising, TV suppliers and Internet providers. The services can only be successful as long as the quality is guaranteed [27]. It is very important to ensure the maximum possible quality of experience. In the video services there are a number of factors that can affect the quality of experience: the resolution, the absence of image errors, the previous experiences of the users regarding the quality, the fulfillment of expectations, the possibility to pause and rewind, the absence of advertisements, the quality of the display, the distance between user and display and even the factors related to the sound that accompanies the video [28].

It is important to note that we talk about quality of experience, which is the subjective perception of the user, in contrast with traditional metrics such as packet loss, delay and jitter. Measuring quality of experience in IPTV represents an innovation compared to the current solutions that are applied to IPTV networks, because in most networks only packet loss, delay and jitter are measured [29][30][31].

The main objective of the thesis is to provide a solution to monitor the quality of experience in an IPTV network. [32]

1.2.2 Secondary objectives

List of secondary objectives:

- SO1: Select the best full-reference metric among those publicly available. Evaluate its applicability to IPTV networks in terms of performance.

- SO2: Propose a solution that combines both full-reference metrics and no-reference metrics for IPTV monitoring.

- SO3: Develop and validate a no-reference bitstream metric suitable for IPTV networks.

Here we offer a description of the purpose of those secondary objectives:

- The full-reference metrics are the most accurate. Therefore, we consider that their application to an IPTV network would be very beneficial. However, its applicability must be evaluated. The selection of the best full-reference metric is one of the secondary goals of this thesis, as a previous step to the evaluation of the applicability of full-reference metrics to IPTV networks (S01).

- In our studies it was discovered that full-reference metrics could only be applied to a subset of situations to monitor. The objective is to solve this problem and in the thesis the solution of combining both full-reference metrics and no-reference metrics for IPTV monitoring is studied (S02).

- Finally, in order to make our own contribution regarding the no-reference metric to be used in IPTV monitoring, we set the goal of developing and validating a no-reference bitstream metric (S03).

Chapter 2
State of the art

2.1 Basics

2.1.1 Introduction

In this section, we introduce the basic concepts of subjective and objective assessment of videos and the evolution of the interest of the research community from algorithms evaluating compression distortions to transmission distortions. In addition, the basics of objective algorithms developments are explained.

The video quality assessment can be done following two different approaches. From one side, it is possible to evaluate the quality by presenting the videos to a sufficient number of users and asking them to give a score for each of them. This approach is called subjective evaluation. On the other hand, algorithms that make an analysis of the images of the videos were also developed and depending on the results of the analysis a score is given to the video. This second approach is called evaluation by objective methods. The objective methods are also usually called metrics.

When a video is compressed usually there is a loss in quality. Many objective methods have been developed to measure the difference between the compressed video and the video prior to compression in terms of quality perceived by the user [33]. In that field a number of researchers made contributions and some of these methods were evaluated by VQEG (Video Quality Experts Group), being established a significant and sufficient degree of correlation between the output of the proposed methods and the real quality of experience.

More recently the research has shifted the focus to the study of the quality of the videos subject to transmission distortions [1]. A typical transmission distortion is caused by packet losses in the transmission. To give the reader an idea, it would be for example a black area in an image or an area of the image that the decoder tried to reconstruct but failed on doing it and cannot be properly recognized by the user.

The development of objective algorithms involves the extraction of parameters from the videos and the recollection of the scores given by users and combining that information in order to generate a model. Commonly accepted procedures for subjective evaluation [25][26] of videos involve satisfying certain requirements regarding the room in which the evaluation takes place (for example the illumination), the brightness of the display and the distance between user and display. In addition to that, the subjective scores collected must be treated with specific statistical methods in order to prepare for objective model development. The statistical methods to be followed ensure that data collected is valid.

New forms of objective models development that do not require subjective evaluations are being proposed in the current thesis. The author proposal consists in using already evaluated very accurate full-reference metrics, such as the VQM metric [34], build training databases which can be used to develop no-reference metrics.

When the development of the objective model is finished, the algorithm must be validated using a complete different set of videos, that were not used during the development, and that must include also the subjective scores given by a number of users. Please refer to ITU-T Rec P.1401 [35] to know more about the evaluation methods e.g. correlation coefficients involved in the evaluation of video quality metrics.

One common problem found is to develop a model for a very specific scope and validate it using a different scope. In that case, the results of the validation are usually not very good. An example of such problem would happen when a model is trained taking into account only compression distortions but then it is used in the presence of transmission distortions, which represent a different scope [36].

The typical process involved in the development of new objective models is summarized in Figure 1.

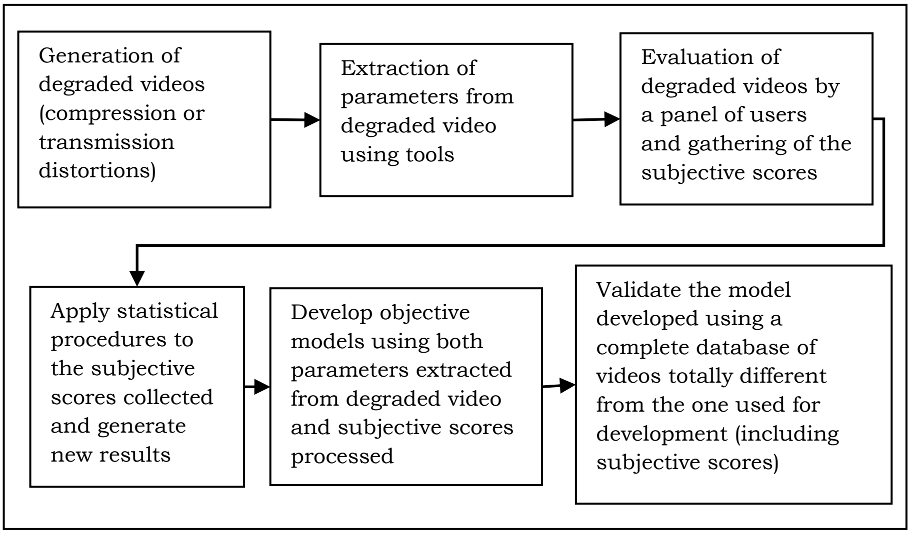


Figure 1. Typical steps involved in the research of new objective models in the field of video quality assessment

2.1.2 Subjective evaluation

The subjective evaluation of a set of degraded videos is accomplished using a panel of observers. In this section, the conditions that must be met regarding the environment, how the videos are presented and rated, and how the results are processed are reviewed.

The main documents that must be taken into account are namely the ITU-T Rec. P.910 [26], ITU-T Rec P.913 [37] and the ITU-R Rec. BT.500 [25]. Also in the VQEG website [11], relevant documentation about the parameters that should be met can be found.

There are different recommendations for the viewing conditions for the case of laboratory environment:

- Ratio of luminance of inactive screen to peak luminance: ≤ 0.02.

- Ratio of the luminance of the screen, when displaying only black level in a completely dark room, to that corresponding to peak white: ≈ 0.01.

- Display brightness and contrast (see Recommendations ITU-R BT.814 [38] and ITU-R BT.815 [39]).

- Maximum observation angle relative to the normal 30º (this number applies to CRT displays).

- Ratio of luminance of background behind picture monitor to peak luminance of picture: ≈ 0.15.

- Other room illumination: low.

The general viewing conditions for subjective assessments in home environment are the following:

- Ratio of luminance of inactive screen to peak luminance: ≤ 0.02.

- Display brightness and contrast (see Recommendation ITU-R BT.815 [39]).

- Maximum observation angle relative to the normal (this number applies to CRT displays): 30°.

- Screen size for a 4/3 format ratio: This screen size should satisfy rules of preferred viewing distance..

- Screen size for a 16/9 format ratio: This screen size should satisfy preferred viewing distance rules Rec. ITU-R BT.500 [25].

- Monitor processing: Without digital processing.

- Peak luminance: 200 cd/m2.

- Environmental illuminance on the screen (Incident light from the environment falling on the screen, should be measured perpendicularly to the screen): 200 lux.

The viewing distance and the screen sizes are to be selected in order to satisfy the PVD (preferred viewing distance). In Figure 2 the recommended ratio of viewing distance to picture height is shown.

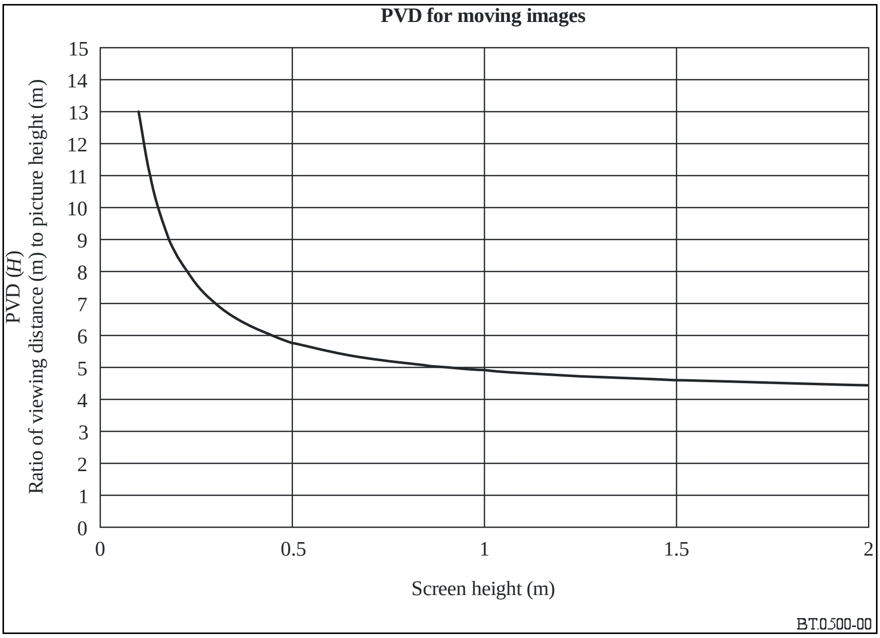


Figure 2. Ratio of viewing distance to picture height to be satisfied [25]

Before making a subjective evaluation the degraded videos must be generated covering the range of qualities from pristine (excellent) to bad. If they are distorted due to compression, an encoder/decoder must be used. Usually the evaluations are done with short videos, for example 10 seconds duration. The videos should represent a variety of contents and account for different characteristics regarding motion, contrasts, pan, scene cuts, colours, illumination changes, zooms.

If the degraded videos are due to transmission distortions, real network losses are simulated. The tool usually takes the encoded file, performs the simulation and generates a new encoded file. Then the videos must be decoded prior to presentation to the user.

In order to present the videos to the user and ask for ratings, specific software tools have been developed. Acreo Swedish ICT provides a software called AcrVQWin [40]. The software can be used to run subjective experiments for video quality in Windows environment, using the Absolute Category Rating (ACR) method [26]. It implements the test procedure in the Video Quality Experts Group (VQEG) MultiMedia phase I testplan. This tool was used by all laboratories performing test in VQEG MultiMedia phase I [20].

2.1.3 Classification of objective metrics

Now the three different main categories of objective metrics are reviewed in more detail. These are based on the amount of information that is available from the original undistorted video in the objective metric that is developed. Both the full-reference and the no-reference metrics have been used in the development of this thesis.

In Figure 3 the no-reference quality estimation scenario is shown. In this case no information regarding the undistorted video is available for the metric.

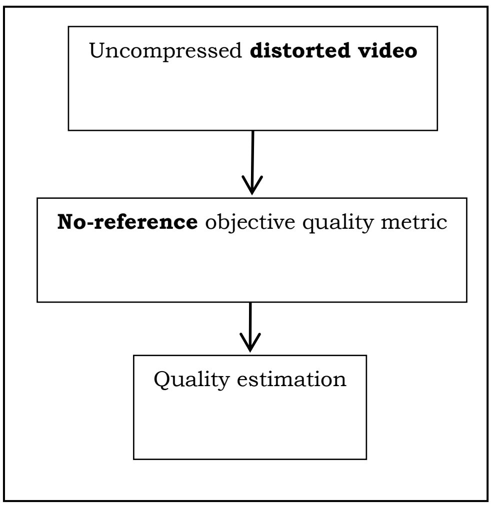


Figure 3. Quality estimation scenario using no-reference objective metrics

In Figure 4 the reduced-reference quality estimation scenario is shown. In this case, additionally to the uncompressed distorted video there are certain parameters extracted from the original undistorted video which are available to the metric.

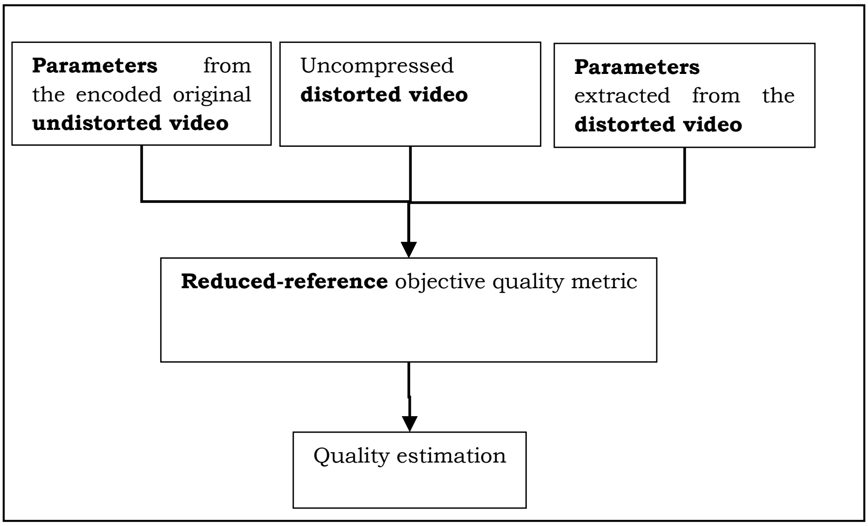


Figure 4. Quality estimation scenario using reduced-reference objective metrics

Finally, in the Figure 5 the full-reference quality estimation scenario is shown. Here in addition to the decompressed distorted video the complete uncompressed original undistorted video is available to the metric. This represents the case where more information is available to the metric, in comparison with the no-reference or reduced-reference cases. This is the reason why the full-reference metrics are considered to be the most accurate.

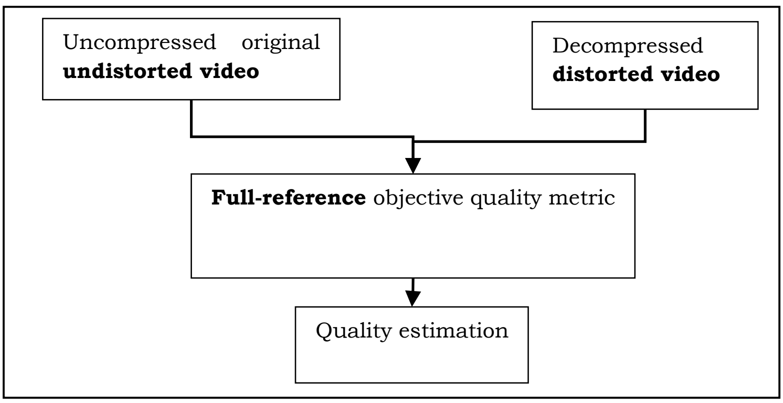


Figure 5. Quality estimation scenario using full-reference objective metrics

Apart from this main classification, objective metrics can also be classified depending on the type of input data as:

- Media-layer models. They use only the speech or video signal to compute the Quality of Experience (QoE)

- Parametric packet-layer models. They inspect only the packet header such as IP-packets: RTP (Real-time Transport Protocol) or UDP (User Datagram Protocol). The main document that must be taken into account is namely the ITU-T Rec. P.1203 [41].

- Parametric planning models. They make use of quality planning parameters for networks and terminals in the prediction of QoE. The main document that must be taken into account is namely the ITU-T Rec. G.1070 [42].

- Bitstream-layer models. They analyze the encoded bit-stream itself and extract the information needed for prediction but they don't fully decode the video.

- Hybrid models. They use both video pixel information and bit-stream information (full decoding of video).

Another way of classifying the objective methods is related to their usability in adaptive streaming solutions as out-of-service methods and in-service methods. The difference between them is that the out-of-service methods cannot be used in real-time while the in-service methods are suitable for that purpose.

2.1.4 Types of features and main related indicators

The main objective of the thesis is to provide a solution to monitor the quality of experience in an IPTV network. In this section, the main degradations that can be measured by the algorithms are introduced. First we will list and explain the spatial degradations and later on we will detail the temporal ones.

Classification of spatial degradations

A spatial degradation is a degradation in part of one or various images of the video. The time factor is not involved in this type of degradation.

Among the most common found degradations, we can find the following: blockiness, blurriness, geometric distortions, deinterlacing artifacts, spatial error concealment and channel switches [17].

Blockiness

Blockiness or block artifacts are usually caused by the use of lossy compression. The transform of NxN pixel blocks (4x4 or 8x8) and the quantization of the DCT coefficients for each block separately causes noise shaping that leads to coding artifacts in the form of discontinuities for coded block boundary. Color intensity changes are localized in uniform areas of the image by the removal of the least significant coefficients of DCT.

Blurriness

Blur is mostly caused by the removal of DCT coefficients of high frequency or by introduction of loop-filters to counteract on blockiness, which both lead to low-pass filtering. This effect can be seen as a loss of detail in the image, reducing sharp edges and texture of objects.

Geometric distortions

Geometric distortions may be caused by various types of image adaptations such as re-scaling due to aspect ratio conversion.

Deinterlacing artifacts

Related to the television broadcast, the interlaced format is still broadly used. Associated with it, there are different perceptual artifacts like the inversion of the top-field and the bottom-field, and the de-interlacing algorithms used in current display technologies.

Spatial error concealment

When there are packet losses, information from the pictures is lost. To correct it, the decoder usually replaces the missing content with previously transmitted content or by in-painting of surrounding regions. However usually the information reconstructed does not have coherence in respect to surrounding perceptual information. This artifact usually affects various images as the missing blocks of one image are used for further image prediction.

Channel switches

The cannel switches, both voluntary and involuntary introduce content mixtures similar to error concealment artifacts and pauses in transmission.

Classification of temporal degradations

A temporal degradation is a degradation where the time factor is involved. Typical examples are too low frame rate, frame freezing or frame skipping.

Frame rate

The frame rate affects directly the quality perceived by the user. A slower frame rate is qualified as worse quality than a quicker frame rate. For the video sequence to be fluid for the user, a minimum frame rate must be met, but it seems from that point on, a higher frame rate is still perceived by the viewer. There are at least two possible effects. The first one is the flicker fusion frequency, which is in principal above 50-60 Hz and the other is motion blur i.e. fast moving objects becomes blurry.

Frame freezing and frame skipping

When the video cannot be decoded at a given time, usually the playback pauses and then resumes skipping a few frames. In other occasions, the video freezes and resumes without skipping frames. This type of distortion may appear both in TCP and UDP transmission.

The perceptual impact of frame freeze and frame skip is highly content dependent and viewers prefer a scenario in which a single but long freeze event occurs to a scenario in which frequent short freezes occur when the total freezing time is comparable.

2.1.5 Objective algorithms development

In this section we introduce the usual steps that are to be followed to develop an objective metric and also we detail the many factors that can influence the video quality assessment process.

The development of an objective algorithm involves many steps that must be executed prior to the algorithm design. Also the designed algorithm must be properly validated, using a completely different subjective database including the scope for which the algorithm was designed.

In Figure 6 an overall view of the whole process is shown.

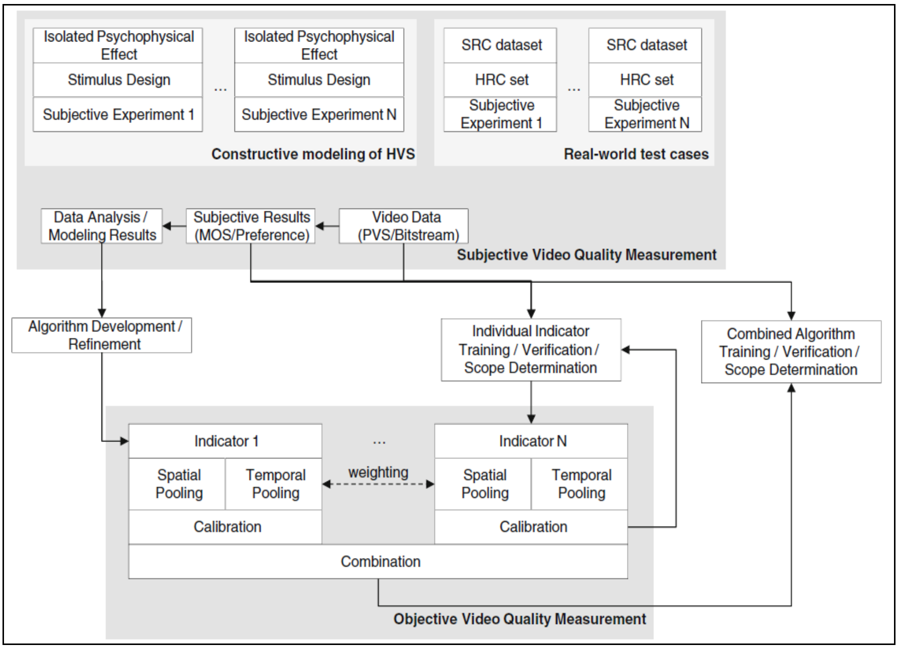


Figure 6. Overall view of the process of development of an objective algorithm [36]

The development of an objective algorithm consists in a series of steps that must be properly executed in order to guarantee the validity of the results. As a first step, different psychophysical effects must be isolated and analyzed by means of separate subjective tests. The goal of this step is to identify what psychophysical effects have significant impact on the perception of the user. Once they have been identified, it is time to shift to real-world test cases. In the real-world test cases different degraded videos datasets are prepared, and exposed to the users that give scores to the videos. Both the subjective results together with the video data (Processed Video Sequence /Bitstream) is feed to the algorithm modelling part. The built model must necessarily take into account the temporal dimension in order to be complete. Once a model has been developed, it will only be considered valid if validated against a subjective database consisting of the scope selected and not used at all in the development of the algorithm.

In the ITU Rec P.1401 [35] the methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models are specified.

In the Recommendation it is discussed also how to make use of MOS (Mean Opinion Score) from different experiments, because usually it happens that they cannot be compared easily. Normally, only the MOS values from the same test type can be compared. Also, the same scale (for example Absolute Category Rating) must be used. Even if the same participants are involved each experiment is different.

Subjects are also influenced by the short-term history of the samples they have previously scored. Depending on whether the previous sample was of poor or good quality, the user will tend to rate the next video higher or lower than it should.

Another effect found is that people tend not to use the entire set of scores offered in an experiment, causing that in an experiment that contains mainly low quality samples, people tend to score them higher, and vice versa.

In addition, individual distortions for samples presented less often are scored lower, as compared to experiments where samples are presented more often and people become more familiar with them.

Last, there is a difference in general cultural behaviour of the individuals that has influence in the different interpretation of category labels as well as expectations. It is considered a long-term dependency.

All the differences that have been pointed cannot be avoided but can be minimized by providing informative instructions, well-balanced test designs, a sufficient number of participants and a mixed presentation order. However it still remains very complicated to directly compare MOS values from different experiments.

The processing of the MOS scores has the goal of scaling and normalizing the values obtained to avoid the difficulties mentioned.

2.1.6 Conclusion

The research done by a significant number of researchers during many years led to the outcome of the detections of many factors than can influence the video quality assessment process and the establishment of procedures that must be followed in the evaluation of metrics and subjective testing. Furthermore, a number of objective metrics were developed, and some of them standardized. There are some documents which constitute a good foundation to research in Quality of Experience assessment methods. These documents are the documents issued by the VQEG (Video Quality Experts Group) [11] as well as the ITU Recommendations, namely the ITUT-R BT.500-14 [25], ITU-T P.910 [26] and the ITU-T P.1401 [35].

2.2 Objective metrics

2.2.1 Review of existing objective metrics

In this section we review the state of the art regarding objective metrics for video quality assessment.

In [43] a prediction model based on computing the number of losses on different type of frames (I, B or P) is presented. The model is trained with the full-reference model VQM [34] (this particular model is named Video Quality Metric). In this thesis a similar approach is followed to develop a no-reference model.

In [43] they show the idea of using a database to map parameters or features to objective quality scores, instead of using a fitting curve.

In [44] the authors propose in the field of reduced-reference metrics, the use of convolutional neural networks to allow a continuous time scoring of the video. In particular, they propose the use of time delay neural network. The paper focuses mainly on the temporal pooling from the human vision system point of view. Among the features extracted used for testing of the system proposed in the paper are the following: frequency content measures: GHV and GHVP, temporal content measure (power of frame difference) and blocking measure. The paper shows that the neural network could be used for temporal dimension across the video of a model.

A no-reference low-complexity QoE measurement algorithm for H.264 video transmission systems is depicted in [45]. In [45] a no-reference model considering compression and transmission distortions is developed and compared using VQM [34] (Video Quality Metric) and SSIM [46] (Structural Similarity Index) metrics. Regarding the part of the algorithm related to encoding, the following parameters are taken into account: average length of the slice motion vectors, length standard deviation of the motion vectors, length entropy of the motion vectors, entropy in different directions of the motion vectors and mean residue energy of the image.

Related to the part of the algorithm that considers the transmission distortions basically the algorithm assigns a different number depending on if each block of the image has been correctly received and based on that information and on the motion vectors it computes an average of the error quantity in the frame. Finally, an exponential formula is used to combine the scores of the compression and transmission distortions.

In [47] a no-reference model based on neural network feed with bitstream parameters is presented. The paper builds on previous work and neural network techniques with support vector machine in order to reduce the computational complexity of the neural network. The parameters used are Quantization Parameter, the ratio of intra blocks out of the total blocks in an inter frame and many parameters related to motion vectors. The proposed method is evaluated against subjective MOS (Mean Opinion Score), PEVQ (Perceptual Evaluation of Video Quality), PSNR, SSIM and MS-SSIM (Multi-scale Structural Similarity Index).

A no-reference video quality assessment model based on Laplacian pyramids is presented in [48]. The authors present a method to predict the quality of videos subject to compression distortions based on measuring the distortion. Each frame of the distorted video sequence is first decomposed to an N-subband Laplacian pyramid, then their intra-subband and inter-subband statistical features are fully exploited. Three intra-subband features and three inter-subband features are used in the prediction. The metric is evaluated on the Live database and compared with MS-SSIM [5], V-VIF (Visual information fidelity) and MOVIE [49] (Motion-Based Video Integrity Evaluation). The approach seems rather complex.

In [50] the authors propose a reduced-reference model based on PSQA (Pseudo-Subjective Quality Assessment) that takes into account parametric information: resolution of the video sample, the amount of movement of the viewed sample, the percentage of packets being lost during the transmission of the video sample, the mean loss burst size during the transmission of the video sample and the sum of TI [51] (scene cut Temporal Perceptual Information) and SI (Spatial Perceptual Information) calculated for the original sample. The function is implemented by means of a neural network.

In [52] they develop a no-reference model in which they first identify the flat regions in the images of the video and then evaluate the blocking artifacts. The fact of first discarding any other region improves the accuracy of the estimation. The flat-region is considered to be the smooth part of the image, namely the magnitude of gradient is smallest than in other regions (edge-region and texture-region).

In the document in [53] and in VQEG phase I [11] they review the most used methods for subjective evaluation as well as the following full-reference objective metrics, that correspond to those models validated and documented by VQEG in March 2000: PSNR, CPqD, Tektronix, NHK/Mitsubishi Electric Corp, KDD, École Polytechnique Fédérale de Lausanne, NASA, KPN/Swisscom CT and NTIA. The document also includes a comparison between HD and SD.

A cognitive no-reference model for mobile streaming services is presented in [54]. In the model, the weights are calculated by means of an artificial neural network that takes as input three parameters: the bitrate, the level of motion and the scene complexity. The quality is computed using a neural network but in two steps.

In [55] two video quality models are introduced. The first one is full-reference and measures the structural distortion and the second is no-reference. The second one is intended for quality estimation of compressed MPEG (Moving Picture Experts Group) video stream. In the no-reference model proposed using the inverse quantized DCT coefficients and the quantization scale the quantization error is estimated. Next, they evaluate the blocking effect and extract the motion vectors. The estimation from the quantization and blocking are combined and a final adjustment is done using the motion vectors.

A review of the HVS (Human Visual System) based image quality assessment models is presented in [56]. First, they concentrate on bionics methods and later on they concentrate on engineering methods. The author proposes that both approaches could be complementary. Two factors are pointed as relevant that could be considered in the development of a metric: the visual attention mechanism and the perception of the chromatic part of the image.

In the end of degree project [57] the authors review the different steps of the Human Visual System (optical, retina and brain processing) with detailed view of each component involved. The properties of the human vision are explored: sensitivity to the luminance, sensitivity to contrast, sensitivity to frequency, masking, eye movements and visual attention. The different types of distortions are reviewed. The article points that the contrast sensitivity function can be used to measure the visibility of a distortion. Also it is commented that in the research community most efforts were put on the evaluation of quality of images and not so much on the quality of the videos. Many metrics take as a fundamental point of start metrics that were designed for image quality assessment.

Finally in [14] an hybrid model taking into account the quantization parameter is shown. The hybrid model was designed by Marcus Barkowsky and implemented by the author of this thesis, as part of the VQEG work. The formula is shown in (1).

$$EstimatedMOS = -0.172 * meanSequenceQP + 9.249 \qquad (1)$$

The formula is a result from Marcus Barkowsky derived from a set of subjective tests with varying conditions regarding both the distortions applied and the input videos characteristics.

In [14] the VQEG tool for creation of PVS (Processed Video Sequences) is described. A processed video sequence is a video encoded, optionally impaired (consequence of the transmission process) and decoded.

The VQEG tool makes use of an XML format that captures all the available information from the encoded video bit stream, the decoded video sequence and the network level. The whole process of Processed Video Sequence generation is repeated for each of the source sequences. In the tool developed by VQEG, the process has been automated and can be configured by a graphical user interface.

During the generation of Processed Video Sequence, all the information collected from the encoded video bit stream, the decoded video signal and the network level, which is very extensive, is stored in a XML file of approximately 1 gigabyte for a video sequence of 10 seconds. This information can serve as an input to the hybrid no-reference H.264/AVC objective video quality metric. This is shown in Figure 7.

In particular the work done by the author of this thesis was the processing of the XML generated by the VQEG JEG (Joint Effort Group) tool and computing the value of MOS based on the QP. The program done by the author of this thesis for processing the XML and computing the model is shown in [14]. The implementation was specifically designed to parse the XML file in very short time. The author contributed to the JEG of VQEG with this implementation.

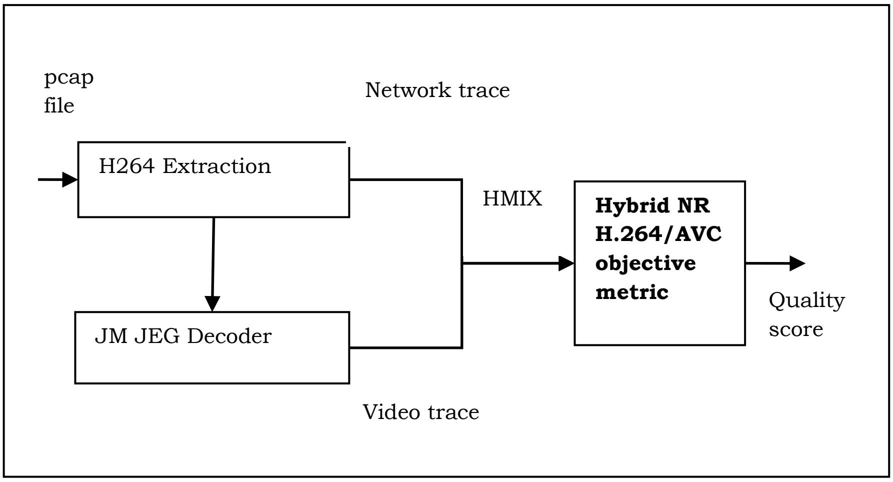


Figure 7. The combination of the network and the video trace into an HMIX file (XML format) can be used by an objective metric to compute a quality score, together with the Processed Video Sequence (decoded video)

2.3 Subjective databases publicly available

In this section, the subjective databases publicly available that are of interest for this thesis are described. The subjective databases represent great contributions due to the amount of effort and usefulness of those resources. We used in total four different publicly available video databases. One of them (IRCCyN University of Nantes subjective database [16]) was used for application of the implemented model (the one that depends on the QP) within the VQEG JEG framework (see previous section for details). The other three of them (EPFL-PoliMI (Ecole Polytechnique Fédérale de Lausanne and Politecnico di Milano [58], IRCCyN University of Nantes HDTV video database [16] and LIVE Wireless video database [59]) were used to evaluate many full-reference methods and select the best metric. The databases contained videos encoded in H.264 and with transmission distortions. The transmission mechanism in all cases was RTP (Real-time Transport Protocol) over UDP (User Datagram Protocol) and the transmission distortions were caused by packet losses. For example, one of the cases that match these conditions is an IPTV network.

Table 2 shows the different parameters of the three databases used to evaluate the full-reference methods.

Table 2. Parameters of the three databases for the study of full-reference methods

	EPFL-PoliMI	HDTV video database	LIVE Wireless video database
Number of sequences	Total 78, 6 different source sequences	Total 45, 9 different source sequences in compressed and uncompressed formats	Total 170, 10 different source sequences
Resolution	4CIF (704x576)	1920x1080	768x480
Duration	10 seconds and 8 seconds	10 seconds	10 seconds
Reference	Compressed (with high quality)	Compressed (with high quality) and uncompressed	Compressed (with high quality)
Compression parameter	Fixed QP between 28 and 32	Fixed QP value 26	Reference video: Fixed QP value 18 Degraded videos: bitrates 500 kbps, 1 Mbps, 1.5 Mbps and 2 Mbps
I-frame period	Not available	24 frames	Reference video: 14 frames Degraded videos: 96
Frame rate	25 and 30 fps	59,94 (interlaced) fps	30 fps
Transmission distortions	PLR (Packet loss rate) 0.1%, 0.4%, 1%, 3%, 5%, 10%. Two different channel realizations for each PLR.	PLR 0.7% (from 42% to 56% of the way), 4.2% (from 21% to 64% of the way), 4.2% (from 42% to 56% of the way).	PLR 0.5%, 2%, 5% and 17%
Encoder/decoder	H.264/AVC JM reference software	Not available	H.264/AVC JM reference software

Number of subjects	21 at PoliMI lab and 19 at EPFL lab	24	31
Processing of subjective scores	Difference scores → Z-scores (with outliers detection) →re-scaling to range [0,5] → DMOS	Difference scores → Z-scores (with outliers detection) →re-scaling to range [0,5] → DMOS both for compressed and uncompressed reference	Difference scores → DMOS →re-scaling to range [0,5]

2.3.1 IRCCyN University of Nantes subjective database

The IRCCyN University of Nantes subjective database is available for download without any cost [16].

In total 10 source reference sequences (SRC) were processed with 16 degradations. The SRC, listed in Table 2, were selected to spread a large variety of different content in Full-HD 1920x1080p25 format. The duration of each sequence was 10 seconds. The degradations are due to encoding process and for some HRCs (Hypothetical Reference Circuits) also due to packet losses. Two different encoders were used, the JM reference encoder and the x264.

Table 3 shows the parameters for the different degradations:

Table 3. Hypothetical Reference Circuits

HRC	**Remarks**	**Encoder**	**R/QP**	**GOP**	**Packet loss**	**Decoding**
0	Reference					
1		x264	16/-	IB7P64		JM
2		JM	-/32	IBBP32		JM
3		JM	-/38	IBBP32		JM
4		x264	8/-	IB3P16		JM
5		x264	4/-	IB7P64		JM
6		x264	1/-	IB7P64		JM

7		x264	-/32	IB3P16		JM
8	FPS↓2	x264	8/-	IB3P16		JM
9	Res↓2	X264	8/-	IB3P16		JM
10	Enc. JM IBBP32 Dec. JM	JM	-/44	IBBP32		JM
11		JM	-/32	IBBP32	Gilbert weak	JM
12		JM	-/32	IBBP32	Gilbert strong	JM
13		JM	-/32	IBBP32	Gilbert strong	ffmpeg
14		x264	8/-	IB3P16		JM
15		JM	-/32	IB3P16	Gilbert weak	JM
16		JM	-/32	IBBP32	Random strong	JM

In most of the cases for decoding the JM Reference decoder was used. The frame copy concealment strategy was used with the JM decoder, while the motion copy strategy was used with ffmpeg. Either, a fixed bitrate (R, in MBit/s) was used or the quantization parameter (QP) was chosen as a fixed value, resulting in a variable bitrate but an approximately constant quality over the duration of the video sequence. Several different GOP (Group of Pictures) structures were configured, where the notation indicates the number of repetitions of the frame types and the GOP size is provided at the end.

All sequences were streamed using RTP encapsulation without further multiplexing using the Sirannon software. The packet trace was captured with tcpdump. In some of the HRCs, packet losses were introduced by simulating the packet losses on Sirannon with either a random loss model or a Gilbert network model channel.

Regarding the video quality scores, an Absolute Category Rating with Hidden Reference (ACR-HR) test on a 5 point scale as described in ITU-T P.910 [26] was performed in order to assess the video quality of each processed video sequence. The viewing environment corresponded to ITU-R BT.500 [25]. A TV-Logic LVM-401W 40" display was used to display the sequences and calibrated to match ITU-R BT.500 and VQEG guidelines for TFT displays. The viewing distance was set to three times the height of the screen, which is 150 cm.

The total number of observers was 27 (14 males and 13 females), aged from 19 to 48 years old, who viewed the 160 processed video sequences (PVS).

There was a training set of five sequences before the evaluation. At the end of each sequence, a grey screen was displayed, and the observer was asked to evaluate the video quality with a score ranging from 1 (worst quality) to 5 (best quality). According to observers screening criteria from both ITU-BT.500 and VQEG Multimedia Test Plan, none of the observers was rejected.

The mean value and the standard deviation for each HRC was calculated.

2.3.2 EPFL-PoliMI (Ecole Polytechnique Fédérale de Lausanne and Politecnico di Milano)

The first database that we used to evaluate the full-reference metrics was the freely available EPFL-PoliMI (Ecole Polytechnique Fédérale de Lausanne and Politecnico di Milano) video quality assessment database [60][61][58]. That database was specifically designed for the evaluation of transmission distortions.

The database contains 78 video sequences at 4CIF spatial resolution (704 × 576 pixels). The distorted videos were created from five 10 seconds long and one 8 seconds long uncompressed video sequences in planar I420 raw progressive format [62]. The reference videos were lightly compressed to ensure high video quality in the absence of packet losses. The Quantization was set to values in between 28 and 32. The encoder and decoder used was the H.264/AVC reference software in High Profile configuration. For coding efficiency reasons B pictures and Context-adaptive binary arithmetic coding (CABAC) were enabled.

A fixed number of slices per frame was configured. A slice consists of a full row of macroblocks.

In order to compute the DMOS (Differential Mean Opinion Score) values the compressed videos in the absence of packet losses were used as the reference. Note that this is not the most common case found in the literature. Usually the video used as reference is the one prior to the process of compression. Three of the reference videos had a frame rate of 25 frames per seconds (fps). This was accomplished by cropping HD resolution video sequences down to 4CIF [62] resolution and reducing the frame rate from 50 to 25 fps. The other three videos have a frame rate of 30 fps. Different packet loss rates were applied to the videos in order to simulate the transmission distortions. The packet loss was generated using a two-state Gilbert's model with an average burst length of three packets and two different channel realizations were selected for each PLR. Forty people participated in the subjective tests. The ITU continuous scale in the range [0–5] was used for the subjective evaluation [26]. Regarding the forty subjects that participated, 21 from them took part in the evaluation at the PoliMI lab and 19 took part in the evaluation at the EPFL lab. More details about the subjective evaluation can be found in [58][60][61].

The generation of the degraded videos and the subjective tests were done by EPFL and PoliMI entities.

Regarding the processing of the subjective scores the author of this thesis decided to merge the data from the two labs and calculate DMOS (Difference Mean Opinion Scores) and confidence intervals. This processing is detailed in Annex A.

2.3.3 IRCCyN University of Nantes HDTV video database

We used the HDTV video database made freely available by Barkowsky et al. [63] The video database is made of nine different source video sequences. Among all the conditions we selected three different conditions corresponding only to transmission distortions, referred to as HRC 5, 6 and 7. These HRCs are coded with high quality (QP26) and contain simulated transmission errors that have the appearance of blurriness and motion artifacts. The errors were inserted in the middle of the video sequence. In HRC5, from 42% to 56% of the way through the 14s sequence bitstream (before removing the beginning and end of the sequence), 0,7% of packets were randomly lost. HRC6 contained 4,2% of packets randomly lost from 21% to 64% of the way through the bitstream. HRC7 contained 4,2% of packets randomly lost from 42% to 56% of the way through the bitstream. The encoder always used two interlaced slice groups of two macroblock lines. For error recovery, an intra image was forced every 24 frames and the ratio of intra macroblock refresh was 5%. The video resolution was 1,920 × 1,080 pixels at 59.94 fields-per-second in interlaced format. The sequences have a duration of 10 s. In total, 24 naive observers viewed the content. The Absolute Category Rating with Hidden Reference (ACR-HR) conforming to ITU-T P.910 with a five-point rating scale was used. The subjects viewed the content at a distance of 1.5 m corresponding to three times the picture height.

The DMOS values were calculated both for the scenario with compressed reference (QP26, HRC1) and with the uncompressed reference (HRC0). Two outliers were found in the case of compressed reference and no outliers in the case of uncompressed reference.

2.3.4 LIVE Wireless video database

The LIVE Wireless video database contains ten source sequences, each 10 s long at a rate of 30 frames per second. The source videos are in RAW uncompressed progressive scan YUV420 format with a resolution of 768 × 480. However, the videos used as reference were already compressed with high quality (average PSNR > 45 dB). For the reference sequences, the Quantization Parameter was set to 18 and the I-frame period to 14. One-hundred sixty distorted videos were created (4 bitrates × 4 packet loss rates = 16 distorted videos per reference sequence). The simulated wireless transmission errors were inserted to the H.264 compressed videos, which were generated with the JM reference software (Version 13.1). The source videos were encoded using different bitrates: 500 kbps, 1 Mbps, 1.5 Mbps, and 2 Mbps with three different slice groups and an I-frame period of 96. The RD Optimization was enabled, and the baseline profile was used for encoding and hence did not include B-frames. The packet size was set to between 100 and 300 bytes. The Flexible Macroblock Ordering (FMO) mode was set as ‘dispersed’. Packet loss rates of 0%, 5%, 2%, 5%, and 17% were simulated using bit-error patterns captured from different real or emulated mobile radio channels. The JM reference software was used to decode the compressed video stream. For the subjective test, the Single Stimulus Continuous Quality Evaluation with hidden reference was used. A total of thirty-one subjects participated in the study.

The difference scores were calculated by subtracting the score that the subject assigned to the distorted sequence to the score that the subject assigned to the reference sequence. One subject was rejected. The scores from the remaining subjects were then averaged to form a Differential Mean Opinion Score (DMOS) for each sequence. No Z-scores were used. Finally, the DMOS values were scaled to the range [0–5]. The LIVE Wireless video database is no longer publicly available because of the uniformity and simplicity of the content. However, we use this database because our study involves various video databases.

Chapter 3
Selection and application of full-reference metric

3.1 Comparison between full-reference metrics

In this section we compare several full-reference publicly available metrics by evaluating them on three different subjective databases. The databases were described in the section 2.3.2 EPFL-PoliMI (Ecole Polytechnique Fédérale de Lausanne and Politecnico di Milano), section 2.3.3 IRCCyN University of Nantes HDTV video database and section 2.3.4 LIVE Wireless video database. We selected as the specific scope for application of the model to be developed the transmission distortions caused by packet losses in a communication similar to RTP over UDP and the compression distortions were not considered at all. Therefore, the reference used in the evaluation was compressed (the compressed video that is uncompressed for entering the full-reference metric, not the original uncompressed video that has not gone through the compression process). Using as the reference video the compressed one is a novel technique introduced by the author that was properly evaluated as it will be shown later. We found that it allowed greater accuracy compared to the most common used of feeding the full-reference metric with the uncompressed original video.

3.1.1 Objective algorithms

We have evaluated and compared several well-known objective video quality algorithms using the videos and subjective results in the three databases. The objective algorithms are described below. The default values of the metrics were used for all the metrics. No registration problems, i.e., a misalignment between the reference and degraded videos due to the loss of entire frames, occurred in the dataset.

Peak Signal-to-Noise Ratio

PSNR is computed using the mean of the MSE vector (contains the Mean Square Error of each frame). The MSE is computed per frame. The implementation used is based on the 'PSNR of YUV videos' program (yuvpsnr.m) by Dima Pröfrock available in the MATLAB Central file repository [64]. Only the luminance values were considered.

Structural SIMilarity

SSIM [46] is computed for each frame. After that an average value is produced. The implementation used is an improved version of the original version in which the scale parameter of SSIM is estimated. The implementation, named ssim.m, can be downloaded in the author's implementation home page [65]. Only the luminance values were considered.

Multi-scale SSIM

MS-SSIM [66] is computed for each frame. Afterwards, an average value is produced. The implementation used was downloaded from the Laboratory for Image & Video Engineering (LIVE) at the University Of Texas at Austin [59]. Only the luminance values were considered.

Video Quality Metric

For VQM, we used the software version 2.2 for Linux that was downloaded from the author's implementation home page [34]. We used the parameters in Table 4.

Table 4. VQM parameters used

Parameter	**Value**
Parsing type	None
Alignment	Spatial
Valid, gain and temporal calibration	automatic
Temporal algorithm	sequence
Temporal valid uncertainty	false

Alignment uncertainty	15
Calibration frequency	15
Video model	general model

The files were converted from planar 4:2:0 to the format required by VQM (Big-YUV file format, 4:2:2) using ffmpeg.

Visual Signal-to-Noise Ratio

VSNR [67] is computed using the total signal and noise values of the sequence. We modified the authors' implementation available at [68] to extract the signal and noise values in order to sum them separately. Only the luminance values were considered. The VSNR was obtained dividing the total amount of signal by the total amount of noise.

MOtion-based Video Integrity Evaluation

MOVIE [49] includes three different versions: the Spatial MOVIE index, the Temporal MOVIE index and the MOVIE index. The MOVIE Index version 1.0 for Linux was used and can be downloaded from [59]. The optional parameters framestart, frameend, or frameint were not used. Only EPFL-PoliMI was analyzed with MOVIE due to the huge amount of time required for the execution of the MOVIE metric.

3.1.2 Methodology

The typical full-reference scenario is shown in Figure 8. The original uncompressed video is compared to the uncompressed video that contains the compression and transmission distortions. In this paper, we also consider the scenario shown in Figure 9 corresponds to compressed reference. The reference videos are lightly compressed to ensure high video quality in the absence of packet losses. The references are thus similar in quality to the uncompressed original. Therefore, in the compressed reference scenario, the video is first compressed before being used in the evaluation. The decompressed video with compression distortions is compared to the decompressed video with compression and transmission distortions.

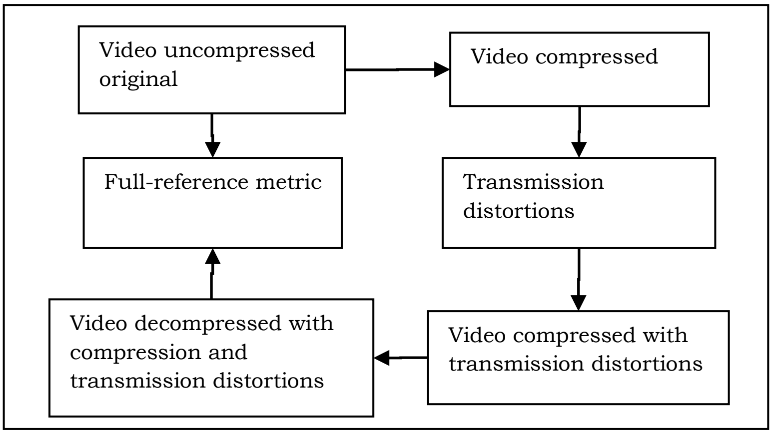


Figure 8. Uncompressed reference used by full-reference metric

In the EPFL-PoliMI and the LIVE Wireless database the reference used is compressed with high quality. In the HDTV video database the reference used is compressed and uncompressed, both cases are considered.

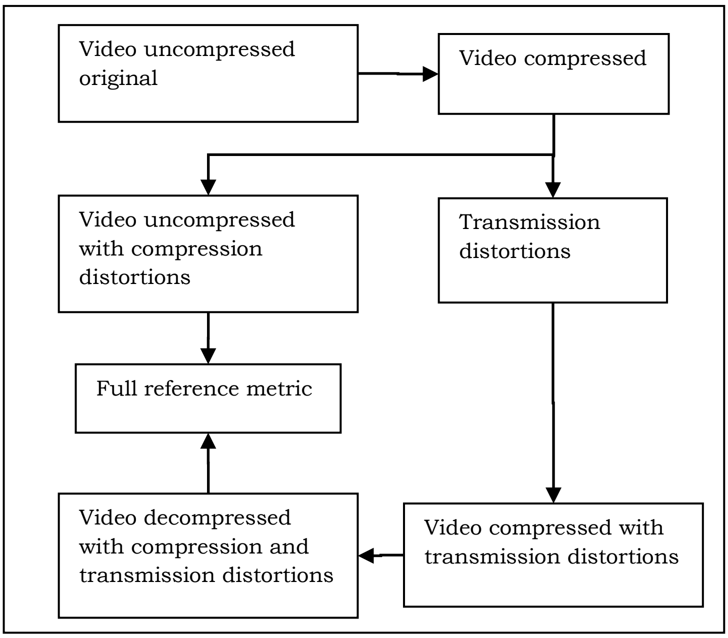


Figure 9. Compressed reference used by full-reference metric

In order to test the performance of the objective algorithms, we computed the Spearman Rank Order Correlation Coefficient (SROCC), the Pearson correlation coefficient, the RMSE, and the Outlier Ratio (OR) [20]. The Spearman coefficient assesses how well the relationship between two variables can be described using a monotonic function. The Pearson coefficient measures the linear relationship between a model's performance and the subjective data. The RMSE provides a measure of the prediction accuracy. Finally, the consistency attribute of the objective metric is evaluated by the Outlier Ratio. The Pearson, RMSE, and Outlier Ratio were computed after a non-linear regression. In the analysis of the EPFL-PoliMI video database, the regression was performed using a monotonic cubic polynomial function with four parameters. The function is constrained to be monotonic:

$$DMOSp = a \cdot x^3 + b \cdot x^2 + c \cdot x + d \qquad (2)$$

In equation (2), the DMOSp is the predicted value. The four parameters were obtained using the MATLAB function 'nlinfit'. In the analysis of the other two databases, the monotonic logistic function with four parameter shown in (3) was used. In this case the parameters were also calculated using MATLAB function 'nlinfit' (function must not be necessarily linear in order to use nlinfit).

$$DMOSp = \frac{\beta_1 - \beta_2}{1 + \exp\left(-\frac{x - \beta_3}{|\beta_4|}\right)} + \beta_2 \qquad (3)$$

Those functions were used because they are recommended by VQEG for that specific purpose [35][69].

In each of the databases, we used the function providing the best fitting. The performance of the metrics is compared by means of a statistical significance analysis based on the Pearson, RMSE, and Outlier Ratio coefficients [34].

3.1.3 Results

In this section, we present the results of the statistical analysis. Also, in several figures, the scatter plots of the VQM objective metric scores vs. DMOS for the different databases are shown. We show the plots of the VQM objective metric because the VQM metric performs very well in all the video databases. The fitting function is also plotted.

EPFL-PoliMI

In Table 5, the values of the coefficients corresponding to all the metrics for the EPFL-PoliMI video database are shown.

Table 5. EPFL-POLIMI video database results

	Pearson	Spearman	RMSE	Outlier Ratio
PSNR	0.958	0.961	0.219	0.625
SSIM	0.959	0.969	0.217	0.597
MS-SSIM	0.964	0.978	0.204	0.597
VSNR	0.974	0.973	0.173	0.472
VQM	0.961	0.960	0.210	0.541
MOVIE	0.965	0.962	0.202	0.625
SPATIAL MOVIE	0.981	0.978	0.148	0.458
TEMPORAL MOVIE	0.924	0.914	0.294	0.611

The values for the Pearson correlation coefficient ranged from 0.92 (for TEMPORAL MOVIE) to 0.98 (for SPATIAL MOVIE). The values for the Spearman rank order correlation coefficient were confined within 0.91 (TEMPORAL MOVIE) and 0.98 (SPATIAL MOVIE). Looking also at the RMSE, we can see that the TEMPORAL MOVIE performed significantly worse than the other methods. In general, the magnitude of the coefficients was high and the differences between them were small. The statistical significance analysis based on Pearson and RMSE confirms that at 95% confidence level MS-SSIM, VSNR, VQM, MOVIE, and SPATIAL MOVIE performed better than TEMPORAL MOVIE, being SPATIAL MOVIE the best performing metric.

In Figure 10 the scatter plot of VQM is shown including the fitting function. The horizontal axis corresponds to the values of the VQM metric. The vertical axis corresponds to the DMOS values. A lower DMOS means higher video quality. The fitting function is plotted in red color. In the scatter plot, we can see that the correlation between VQM and DMOS is not linear and that the correlation is very high.

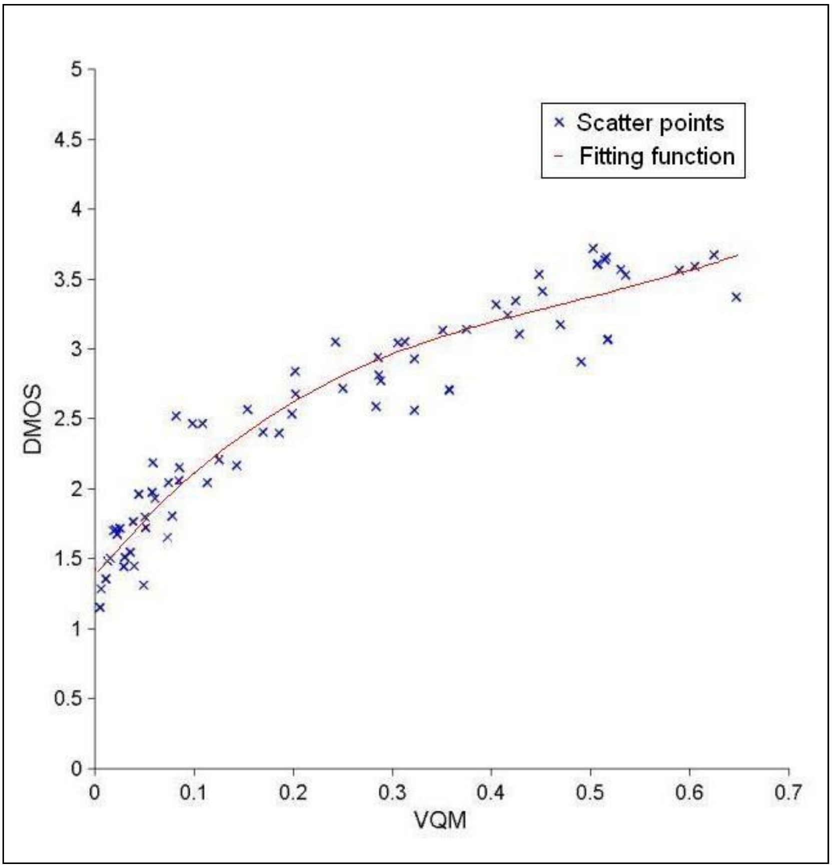


Figure 10. Scatter plot VQM for EPFL-PoliMI

The offset in the figure is explained by the fact that only a subset of DMOS values between 1 and 4 are interesting for the analysis. Although in the figure VQM has some points near to 0 they are in fact never 0 meaning that the quality is never perfect.

HDTV video database

In Table 6, the values of the coefficients corresponding to all the metrics for the HDTV video database when the reference is lightly compressed can be observed.

Table 6. HDTV video database compressed reference results

	Pearson	Spearman	RMSE	Outlier Ratio
PSNR	0.817	0.804	0.346	0.296
SSIM	0.871	0.856	0.295	0.370
MS-SSIM	0.891	0.884	0.273	0.296
VSNR	0.837	0.774	0.328	0.444
VQM	0.887	0.860	0.277	0.333

It can be seen in Table 6 that the values for the Pearson correlation coefficient were distributed within 0.82 (for PSNR) and 0.89 (for MS-SSIM). The values for the Spearman rank order correlation coefficient were confined within 0.80 (for PSNR) and 0.88 (for MS-SSIM). The general magnitude of the coefficients was high. The statistical significance analysis based on Pearson and RMSE shows that at 95% confidence level, there were no significant differences between the studied metrics.

In Figure 11, the scatter plot of VQM using compressed reference is shown including the fitting function. In the scatter plot, we can see that the correlation between VQM and DMOS is not linear and that the correlation is high.

In Table 7, the values of the coefficients corresponding to all the metrics for the HDTV video database when the reference is uncompressed are presented. The values for the Pearson correlation coefficient ranged from 0.63 (for VSNR) to 0.84 (for VQM). The values for the Spearman rank order correlation coefficient had the lowest value at 0.51 (VSNR) and the highest at 0.78 (VQM). The general magnitude of the coefficients was low. The statistical significance analysis based on RMSE shows that at 95% confidence level, VQM performed better than VSNR.

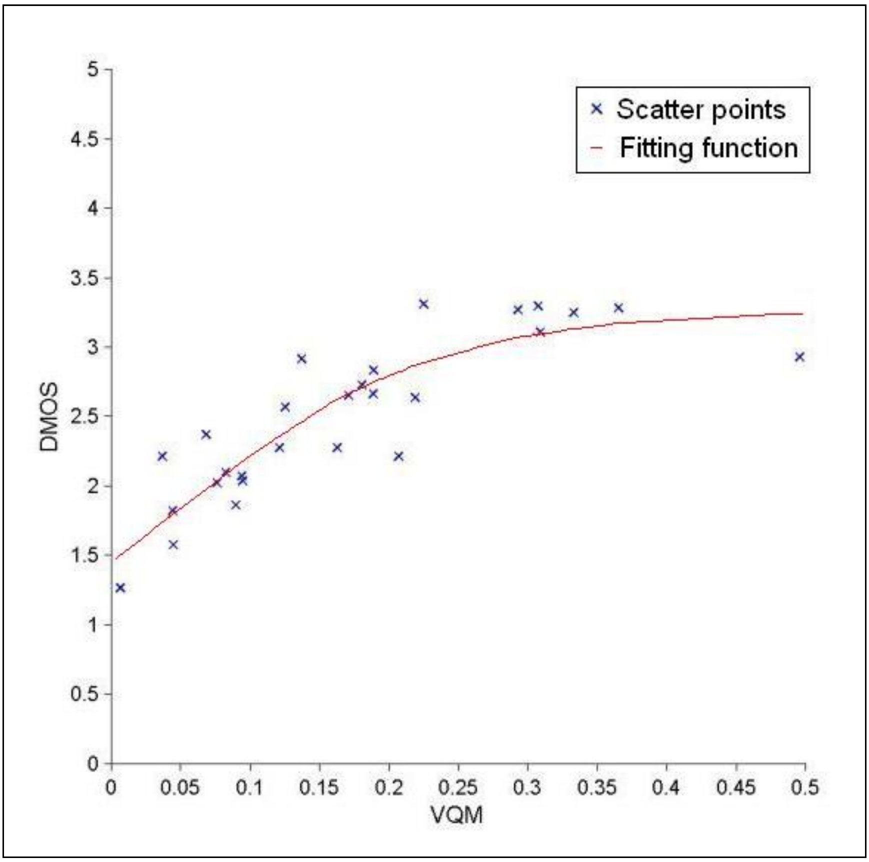


Figure 11. Scatter plot VQM for HDTV video database compressed reference

Table 7. HDTV video database uncompressed reference results

	Pearson	Spearman	RMSE	Outlier Ratio
PSNR	**0.661**	**0.600**	**0.422**	**0.555**
SSIM	**0.720**	**0.653**	**0.391**	**0.518**
MS-SSIM	**0.727**	**0.664**	**0.386**	**0.518**
VSNR	**0.629**	**0.511**	**0.438**	**0.592**
VQM	**0.840**	**0.782**	**0.305**	**0.370**

In Figure 12, the scatter plot of VQM using uncompressed reference is shown including the fitting function. The correlation between VQM and DMOS is high and not linear.

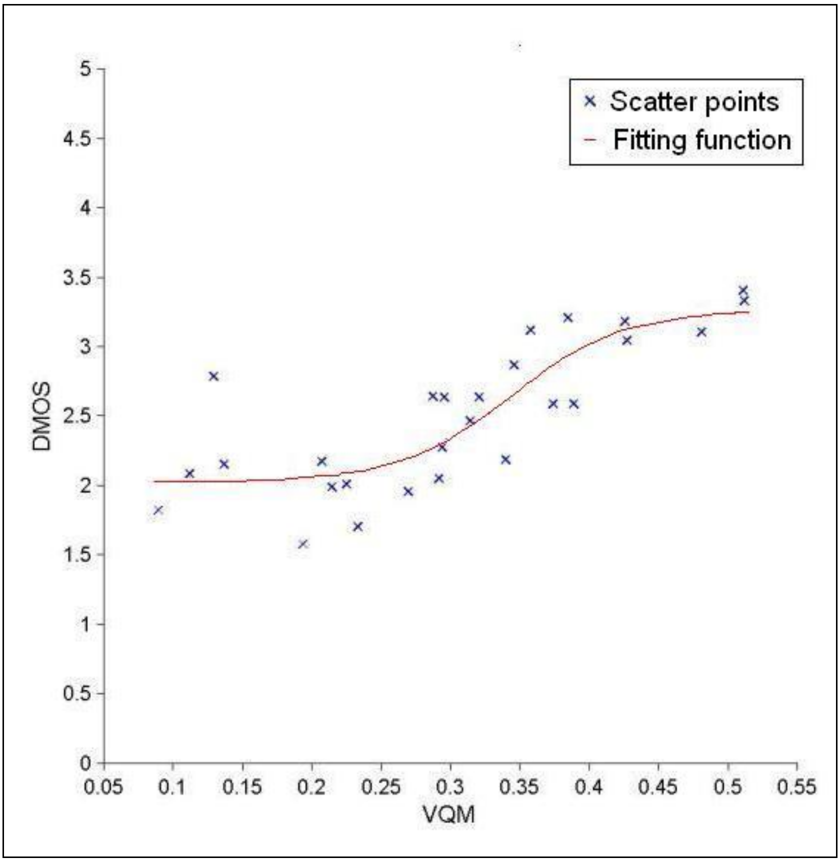


Figure 12. Scatter plot VQM for HDTV video database uncompressed reference

The use of a fitting function such as the one in the Figure 12 is allowed in the video quality community.

Live Wireless database

The coefficients corresponding to the LIVE Wireless database are shown in Table 8.

Table 8. LIVE Wireless video database results

	Pearson	Spearman	RMSE
PSNR	0.959	0.960	0.365

SSIM	**0.954**	**0.954**	**0.386**
MS-SSIM	**0.96**	**0.963**	**0.364**
VSNR	**0.949**	**0.946**	**0.409**
VQM	**0.974**	**0.974**	**0.294**

The values for the Pearson correlation coefficient are distributed within 0.93 (for VSNR) and 0.97 (for VQM). The values for the Spearman rank order correlation coefficient are confined within 0.95 (VSNR) and 0.97 (VQM). The general magnitude of the coefficients is very high and the differences between them are small. The statistical significance analysis based on Pearson and RMSE shows that at 95% confidence level VQM performed better than all the other metrics.

In Figure 13 the scatter plot of VQM is shown including the fitting function. In this case, the correlation between VQM and DMOS is approximately linear and very high.

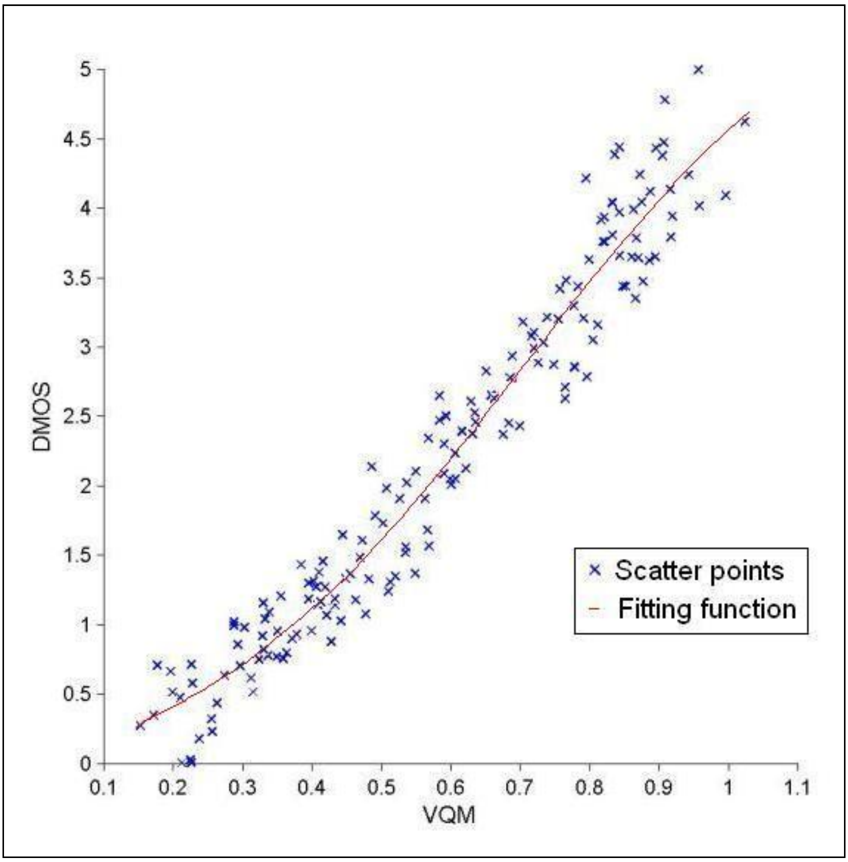


Figure 13. Scatter plot VQM for LIVE Wireless

The curve has a different form when compared to curve of the other two databases because the subjective tests performed for each database are different and the subjective tests involve different people and different videos as well. Therefore is a high variability in the DMOS values for each different database.

3.1.4 Discussion

The goal was to select the best metric that fitted the specific scope we have previously defined. Just to remind it, the specific scope was transmission distortions caused by packet losses in a communication similar to RTP over UDP and the compression were not considered at all.

Our results show that VQM has a very good performance in all the databases, being the best metric among the studied in the HDTV video database (uncompressed reference) and in the LIVE Wireless video database. In the EPFL-PoliMI video database, SPATIAL MOVIE performed better than the other metrics. On the other hand, the performance of TEMPORAL MOVIE was lower than the other metrics, at least for the EPFLPoliMI video database.

The performance of MOVIE, SPATIAL MOVIE, and TEMPORAL MOVIE was not evaluated in HDTV video databases and in the LIVE Wireless video database because the execution of the metric requires a very significant amount of time (many days) in comparison with the other metrics. This fact decreases the usability of these metrics considerably.

In the results from the HDTV video database, we can appreciate that the accuracy in the prediction can be increased if the reference is compressed, compared to the case where the reference is uncompressed.

3.2 Application to reconstruct incompletely decoded videos

3.2.1 Problem to solve

Most of the full-reference objective metrics compare the videos frame by frame and later on an overall quality index is computed. Many full-reference objective metrics require that the original and the degraded video contain the same number of frames. PSNR, SSIM [46] [65] and MS-SSIM [66] are some examples.

Most codecs are not able to decode the video properly when the video is subject to packet losses and produce incomplete video files with lower number of frames than the original video. We present a simple method to reconstruct the degraded videos so that it has the same length as the original.

The solution the author proposes is to insert information regarding the frame numbers into the original uncompressed video and to use it later on to identify the missing frames and reconstruct the degraded video.

3.2.2 Overview of the reconstruction process

In Figure 14 we show what happens when a typical full-reference metric (that compares frame by frame) is used and some frames have been lost (in this example frame number 4 and 7 are lost).

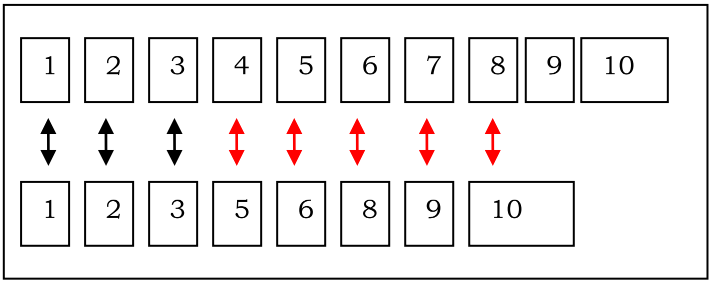


Figure 14. Error in quality metrics due to shift caused by missing frames (original video sequence and degraded video)

As we can see in the final part of the comparison, the frames 8 and 10 are being compared and they are quite different actually. This introduces a significant error. The author proposes to first reconstruct the missing frames before applying the full-reference metrics.

The method for reconstruction proposed by the author is repeat the previous frame to the missing one.

In the Figure 15 we can see an example of how the comparison would be after applying the reconstruction of the missing frames (number 4 and 7).

In this case in the final part of the comparison we are correctly comparing the number 10 of the original video with the number 10 of the distorted and reconstructed video. The error is minimized as compared to the previous described scenario.

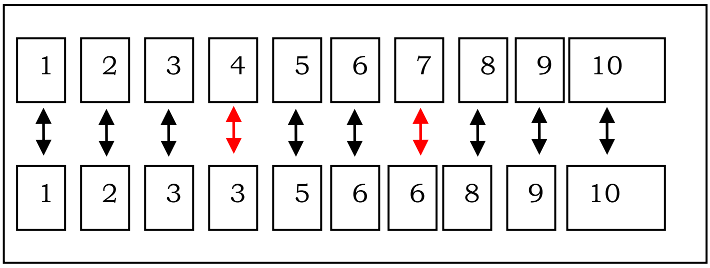


Figure 15. Error in quality metrics is minimized using the reconstructed video (original video sequence and reconstructed video)

Other methods of reconstruction could be to insert the image corresponding to the missing one extracting it from the original uncompressed sequence. Other possibilities would be to repeat the next frame to the missing one or to insert a white or black frame. The author believes that the method chosen is realistic because many decoders produce frame freezing when many packets are lost.

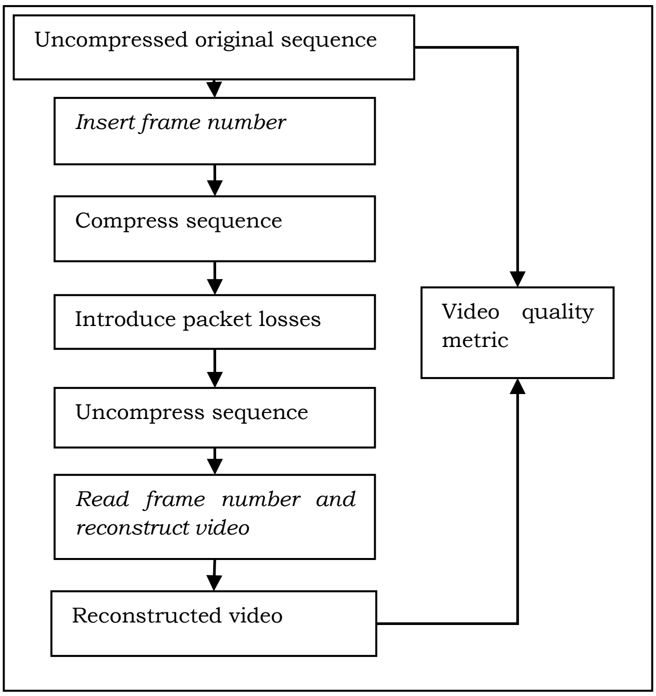


Figure 16. The frame number is inserted in the frames and used in the reconstruction process

In order to reconstruct the distorted video, it is necessary to first detect which frames are missing. The frame number is inserted into the original uncompressed sequence in different position of each image. The video is then compressed, packet losses are introduced and the video is uncompressed. The result is an incomplete video file due to decoder problems. Finally the frame numbers are extracted from the frames, and with that information the video is reconstructed. After all this steps the video would be prepared to be used by a full-reference video quality metric.

In Figure 16 the complete process is shown.

3.2.3 Insertion of number of image

In this subsection the details of the insertion of the number in the image are described. The method proposed has been chosen for its simplicity and robustness to the packet losses impact on the images.

The original uncompressed video is in planar format YUV 4:2:0. YUV formats fall into two distinct groups, the packed formats where Y, U (Cb) and V (Cr) samples are packed together into macropixels which are stored in a single array, and the planar formats where each component is stored as a separate array, the final image being a fusing of the three separate planes. The planar YUV 4:2:0 is an 8 bit Y plane followed by 8 bit 2x2 subsampled U and V planes. An NxN Y plane is followed by (N/2)x(N/2) U and V planes. The YUV 4:2:0 subsampling is detailed in Table 9.

Table 9. YUV 4:2:0 subsampling

	Horizontal	Vertical
Y sample period	1	1
U sample period	2	2
V sample period	2	2

The frame number is inserted into the Y plane. The frames are numbered starting from 1. The frame number is represented by 16 bits, being the most significant to the left. Each of the bits is mapped to one pixel. If the bit is 0 the luminance byte takes the value 0, and if the bit is 1 the luminance byte takes the value 255. The bitwise XOR of the frame number is also calculated and inserted into the luminance information of the video in the same way. Whenever we mention XOR, we are referring to bitwise inversion, that is changing ‘1’s by ‘0’s and ‘0’s by ‘1’s.

The frame number is inserted four times in different positions of the image. The XOR of the frame number is also inserted in four positions of the image. The XOR is used to verify that the frame number read is correct. The reason of inserting four times the frame number and the XOR is to maximize the probability that the number is correctly read despite the degradation of the image.

The Figure 17 shows how the frame number and xor is inserted in the image.

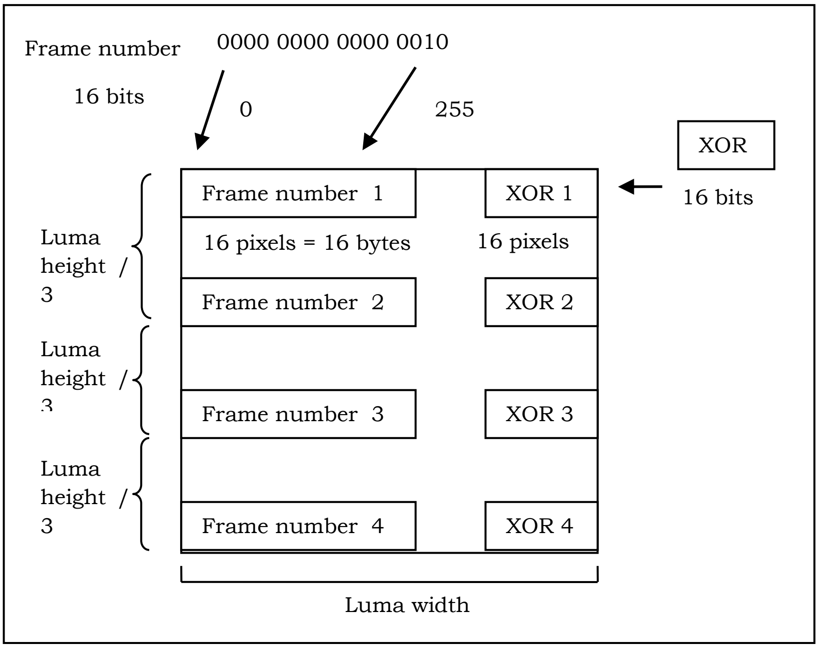


Figure 17. The frame number is inserted four times and the same XOR is inserted four times in the image

3.2.4 Reading of number of image and reconstruction of the file

The reconstruction program receives as input parameters the YUV input file, the YUV output file, the frame resolution and the total number of images. The frame number and XOR are read from four different positions. In the process of reading the numbers and XORs a bit is 0 if the pixel value is lower than 127 and 1 if the pixel value is higher than 127. The number and XOR combinations verified are summarized in Table 10.

Table 10. Combinations verified in reconstruction process algorithm

Number	XOR
1	2, 3 and 4
2	1, 3 and 4
3	1, 2 and 4
4	1, 2 and 3

The number 1 is considered valid if the combination frame number 1 XOR 2 is correct, or the combination frame number 1 XOR 3 is correct or the combination frame number 1 XOR 4 is correct and the number is equal or lower than the total number of images in the video. The same idea applies to the rest of the numbers.

Once the whole video has been read if there are more frames to be inserted the last frame inserted is repeated until completing the total number of frames. On the other side, if the first frame is missing the first processed frame is repeated. The complete algorithm is shown in Figure 19.

3.2.5 Reliability

Both in the development phase and when evaluating reliability, a real encoder, decoder and random packet loss generator were used. The whole process of evaluation is shown in the Figure 18.

The frame number was inserted into the original uncompressed sequence. The sequence was then encoded with the encoder x264 [70] with a Quantization Parameter between 26 and 44. The maximum interval between IDR frames was set to 12, 36, 60 and 84. Random packet losses were inserted using a packet loss simulator [71]. The video was then decoded using ffmpeg, producing a video with missing frames in some parts. The video was finally reconstructed using the frame numbers. The verification of the frame numbers and XOR combinations in different parts of the image decreases the probability of reading erroneous numbers.

During the development phase we tried different algorithms for computing validation numbers (similar to XORs) and also different positions for the frame numbers and XORs in the image. The number of frame numbers and number of XORs were varied until an optimal performance was reached. For each condition we generated new encoded video, inserted packet losses and decoded the video. The result was a degraded video where some parts of the images were changed due to the transmission distortion, causing some numbers or XORs to be unreadable. We manually checked each image in order to see if the algorithm was valid. We varied as well the percentage of packet losses to see the reliability of the method. The final method described was found to be reliable for packet loss rates of around 10% (ffmpeg decoder used). The reliability increases with lower packet loss rates.

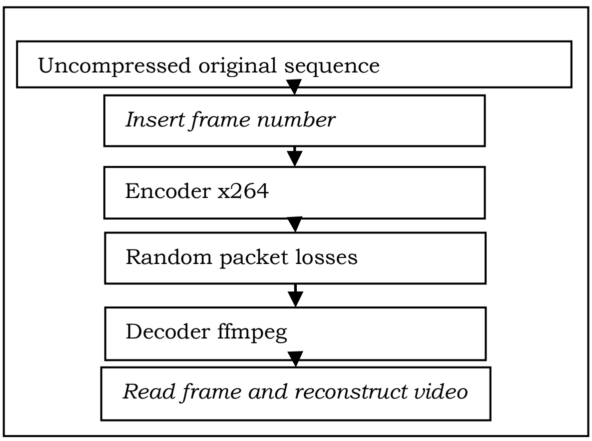


Figure 18. Framework for the evaluation of the reliability of the method proposed

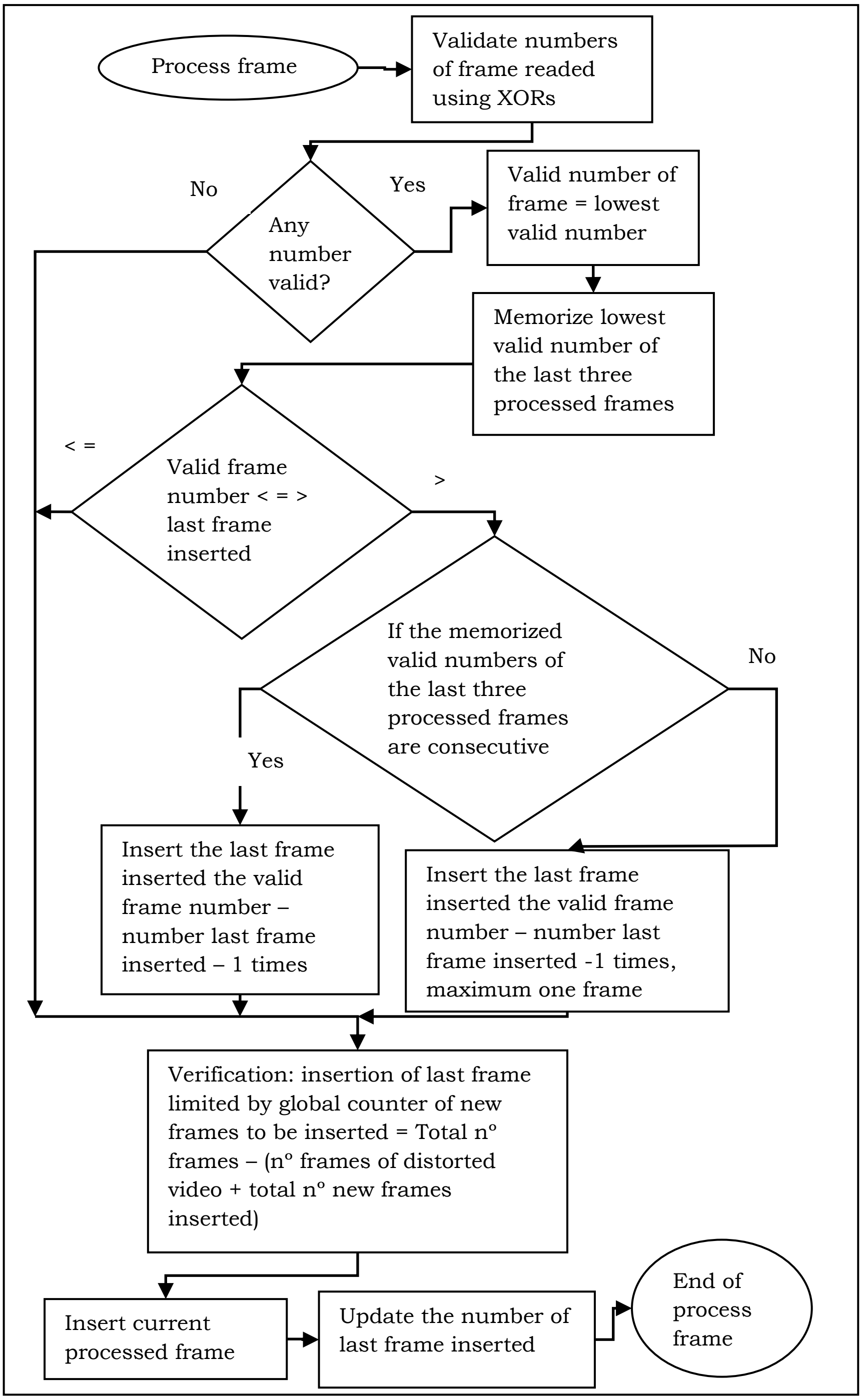


Figure 19. Reconstruction tool algorithm

3.2.6 Comparison with other methods

The method by S. Wolf [72] is a full-reference method for estimating variable frame delays. The best matching original (or input) video is determined for each processed (or output) video frame. The algorithm involves the computation of the Mean Square Errors (MSEs) between each processed frame and the set of original frames that are within a user-specified temporal search window. The algorithm only utilizes the luminance images of the video clips. If we compare both methods, we appreciate that our method is better suited for the case of still or nearly still video or video with repetitive motion. In both methods the position of the frames is identified. However, in our method we include the reconstruction of the distorted video, providing the possibility of directly applying video quality metrics that require videos of the same length, like PSNR or SSIM. Both methods require access to the original uncompressed reference. However, in our method the access to the original uncompressed reference is necessary prior to the processing of the video (compression, introduction of packet losses and decoding) in order to introduce the frame number in the video. The method described by S. Wolf implies more complexity than our method.

3.2.7 Conclusion

The method proposed solves the problem of applying full-reference metrics to degraded videos that have some missing frames. The method is based on the insertion of the number of image in the original uncompressed video and later reading of that value in the reconstruction process. The method has been evaluated using the x264 encoder and ffmpeg decoder, together with a random packet loss generator. This tool was developed as a consequence of a real need found while developing this PhD thesis.

3.3 Application of full-reference metric in IPTV

During the development of this thesis we noticed a recommendation from ITU that we think could be a good solution of assessing video quality in an IPTV network. We studied the recommendation and decided to evaluate its applicability in real IPTV networks. The applicability of this recommendation is basically dependant on the performance of running the processes described on it necessary for video quality assessment.

3.3.1 Full-reference metric applied on transmitter

The scenario considered is shown in the ITU BT.1789 recommendation “A method to reconstruct received video using transmission error information for packet video transmission” [73]. In that recommendation there are two different cases, one of them refers to computing the video quality metric on the transmitter and the other one refers to computing the video quality metric on the receiver. In this section we will focus on the case where the video quality metric is computed on the transmitter.

The scenario, directly extracted from the recommendation, is shown in Figure 20 and Figure 21:

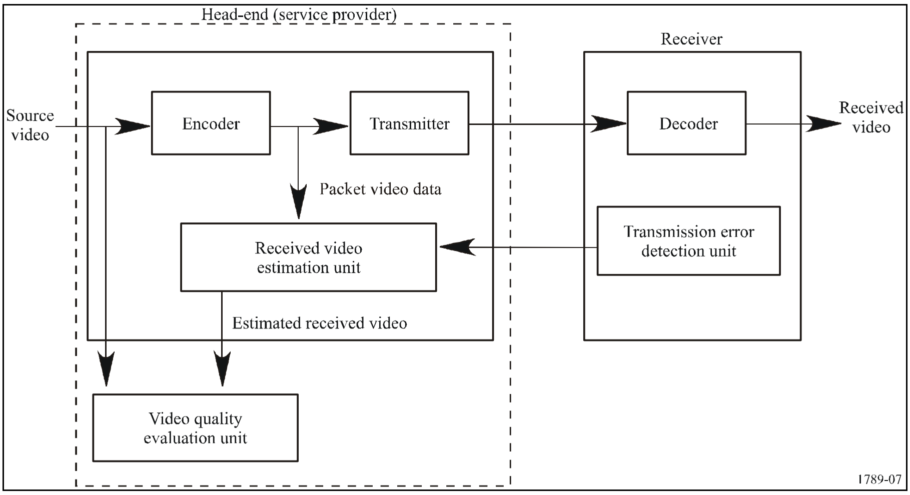


Figure 20. Method for a head-end to monitor video quality of a receiver using transmission error information (ITU-R BT.1789)

As it can be seen in the figure the service provider (transmitter) encodes and transmits a video to the receiver. The receiver, even before decoding the video, is capable of knowing which packets have been lost, which is easily known by taking into account the RTP sequence numbers. The numbers of lost packets are identified in the receiver and sent to the transmitter. The transmitter receives the numbers of the RTP packets lost and using a packet loss simulator (software tool) that also uses as input the encoded video, it can generate a degraded encoded video that will be identical to the one that was received in the receiver. That process is done in the block named “received video estimation unit”. After introducing the packet losses with the packet loss simulator, the video is decoded (in the transmitter) and fed into the video quality estimation unit. The video quality estimation unit is a full-reference video quality metric that needs also the input of the original source video. That way, the transmitter knows the quality of the received video, only exchanging between receiver and transmitter the numbers of the packets lost.

In our approach we have considered the delays can be mapped into packet losses, because if it happens that the decoder does not have the information available on time, it can be considered missing information, given that in an IPTV network there is real-time delivery requirement.

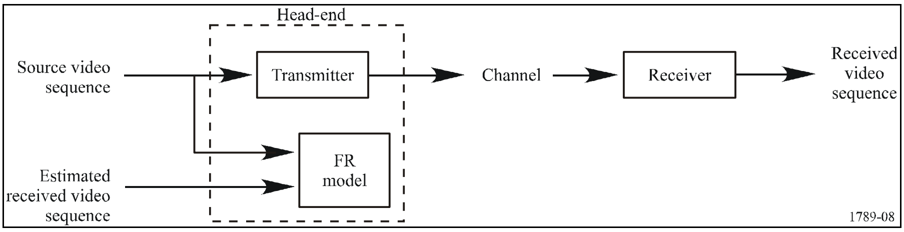


Figure 21. A block diagram for the head-end (service provider) computing the video quality of the received video using the estimated received video (FR model)

Taking as a starting point the recommendation we develop the blocks in the figure and picture how a real implementation would look like. We consider more simple case of one transmitter and one receiver and the more complex case of one transmitter and multiple receivers.

Application of recommendation in case of one transmitter and one receiver

The Figure 22 shows how the solution would look like in case we only consider one transmitter and one receiver.

We have considered H.264 format for encoding and decoding, RTP transmission for video delivery and the VQM metric as full-reference metric. We have selected VQM because it is the one that achieved better performance among the ones studied in this thesis, both for the case of transmission distortions and for the case of compression distortions.

In the transmitter after the packet loss simulator it is interesting to note that the same decoder than the one used in the receiver must be considered, because it is imperative to reproduce the conditions of the receiver. These conditions include the use of the same error concealment technique.

Application of recommendation in case of one transmitter and multiple receivers

Next we show the practical application of the recommendation for the case of one transmitter and multiple receivers (n receivers), instead of just one transmitter and one receiver. This case is shown in Figure 23. It is the common case found in an IPTV network. The source video is encoded just one time because multicast delivery is used to reach the multiple receivers.

The difference with the case of 1 transmitter and 1 receiver is that in the transmitter multiple packet loss simulations corresponding to videos of multiple receivers must be executed simultaneously. This in turn leads to multiple videos having to be decoded simultaneously and multiple simultaneous computations of the full-reference metric. As it will be shown later these multiple processes (one for each receiver video to be analysed) must not be necessarily run at the same time, but may instead be run with some starting delay to obtain better performance.

3.3.2 Performance evaluation

Executing an accurate full-reference metric such as VQM [34] can take minutes in an average computer for just one user. It is unfeasible to analyze all the videos received by users in an IPTV network for example consisting of 10.000 users using a single computer running the VQM metric. Therefore it would be required to use other type of systems or lightweight no-reference metrics [74] in addition to the full-reference metric mentioned. Lightweight no-reference metrics can be used for discarding potential situations to evaluate because they are accurate enough for that task, and where more accuracy is required the full-reference metric VQM could be used. The main work in this section is focused on determining how many situations/users can be analyzed using the VQM metric in a computer with good performance.

The first part of the evaluation consists on evaluating the resource consumption (in terms of CPU, RAM memory and disk transfer capacity) and time needed for execution of the packet loss simulator, decoder and full-reference VQM metric separately. This is done running only one instance (thereby one process) or multiple instances (multiple processes) of the component being analysed.

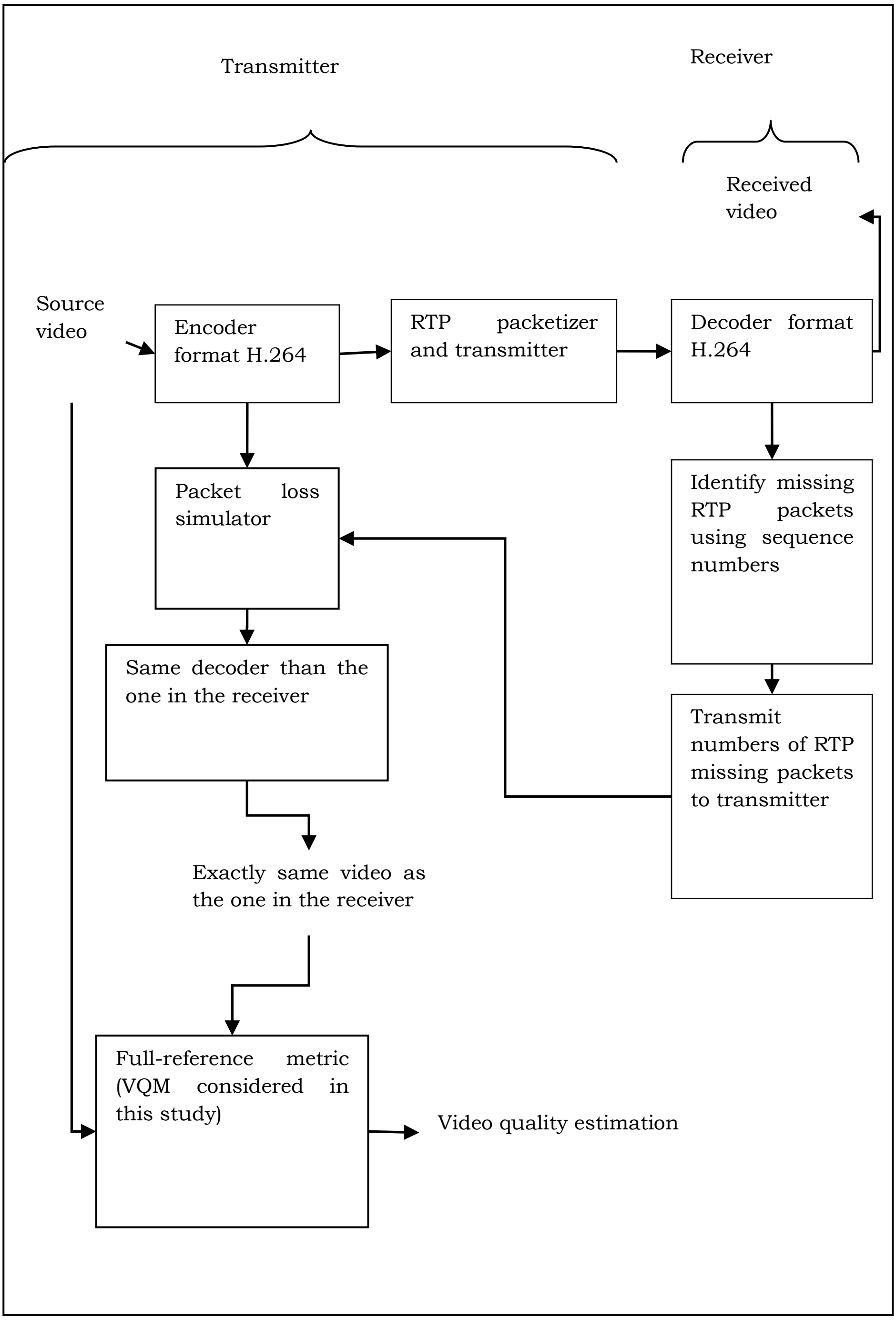


Figure 22. Practical application of the recommendation ITU-R BT.1789 for one transmitter and one receiver

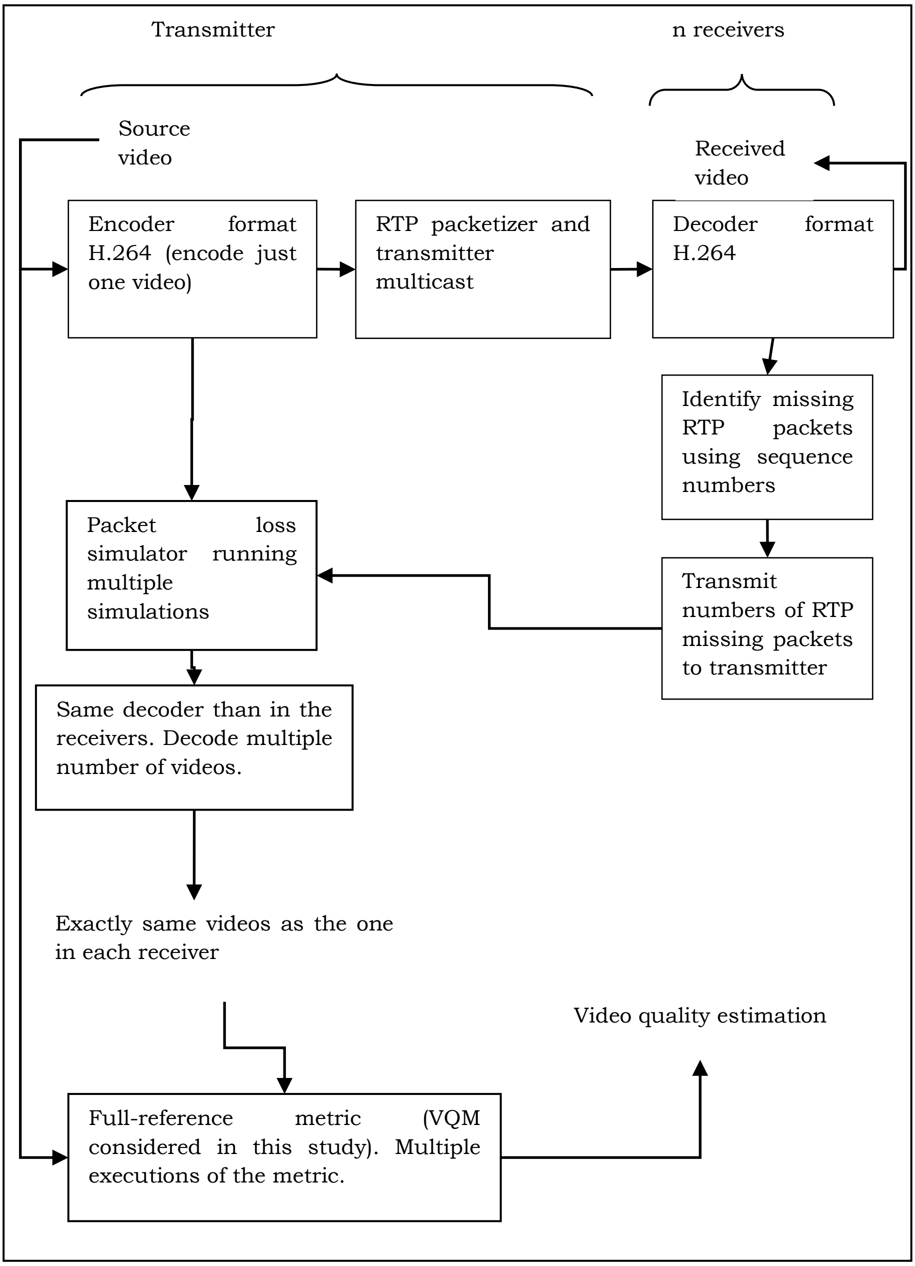


Figure 23. Practical application of the recommendation ITU-R BT.1789 for one transmitter and multiple receivers

The second part of the evaluation consists on analysing the performance of running multiple processes, each one composed of three parts that are executed consecutively:

- One packet loss simulation
- Decoding of the generated file by the simulation
- Computation of the full-reference metric VQM on the decoded degraded video together with the original video as reference.

The starting time of these processes can be configured with a specific delay between the start of one process and the next one. The objective is to measure the time needed for running multiple processes with different starting delays between them.

This second part of the evaluation, as it has been described, was done for different computers. In particular they are done for machines that are reflected in Table 11.

Table 11. Different machines considered in the study

Machine	**Processor**	**RAM Memory**	**Storage disk**
1	i7	32 GB	SSD
2	i5	32 GB	SSD
3	i7	32 GB	NO SSD
4	i7	16 GB	SSD

We had to use different tools in order to reproduce the scenarios. The scripts used to create the different processes and start the simulator, decoder and metric were programmed in Python language.

The simulator is a tool that takes as input an H.264 file and a pattern of zeros and ones representing which slices must be lost and generates as output another H.264 file that doesn't contain the slices to be lost [71]. The generation of the pattern of zeros and ones is done using another python program in which we specify the packet loss percentage.

The decoder used was the ffmpeg that now supports the presence of lost packets and conceals the errors as best as possible. In all the studied cases it generated a decoded video of the expected length (same length as the original video). This fact is important because otherwise if the videos don't have the same length it would be problematic to apply the full-reference metric in a correct way.

Finally, the full-reference metric applied was VQM. For that purpose we used a version with support of 64 bits and we skipped the process of calibration. We used the general model. The operating system used in all cases was Windows 10.

The video had a length of 10 seconds and a resolution of 1920x1280. The video has a medium degree of movement.

The last part of the evaluation is intended to replicate the exact mode of operation that would be used in a real scenario. We think that the normal way of operation would be to have a maximum number of simultaneous processes for optimal operation. When one of the running processes would end, a new one would start running, thus having always the same number of processes running if there is enough work to be done, or lower number than the maximum if there is not so much work to be done. These evaluations are denoted by P+20 or P+60 depending on the number of processes started (will be explained in the results section). In this case we start a number of processes with a variable delay between them and when the maximum number of simultaneous processes is reached, we stop launching new processes, and we only launch new processes when processes are stopped. For each process stopped, a new one is launched.

The evaluation of the resources used by packet loss simulator, decoder and VQM metric with multiple processes is described in Annex B.

Measurements of time for the packet loss simulator, decoder and VQM metric executed as in the ITU-R BT.1789

In this chapter instead of evaluating the packet loss simulator, decoder and VQM metric separately we integrate in each process the packet loss simulation of a video file, the decoding of the video with the packet losses introduced and the execution of the VQM metric to assess the quality of the video. This way, we have a scenario similar to the one that would be used in a real deployment of IPTV video quality monitoring. We make the measurements for four different machines:

- i7, 32 GB RAM and SSD disk.
- i5, 32 GB RAM and SSD disk
- i7, 32 GB RAM and HD disk (no SSD).
- i7, 16 GB RAM and SSD disk.

We selected these configurations in order to be able to see the impact of varying the processor, RAM memory or disk, which make us capable of determining which is the most critical component.

In the Annex B, we present the results of this test using the four different machines. We start the number of processes indicated with a specific time delay between start of processes and there is no limit about the maximum simultaneous processes running in the simulations (see Annex B).

Measurements of time for the packet loss simulator, decoder and VQM metric executed as in the ITU-R BT.1789 for i7, 32 GB RAM, SSD disk with defined maximum number of simultaneous processes, mode P+20

In this subsection we specify a limit about the maximum simultaneous processes running in the simulations. In those cases there is a number of videos to be processed but at a given time only a maximum number of simultaneous processes are executed, and when one of those finishes, another starts its execution. We think that the case denoted by P+60, which will be shown later, is the closest to a real deployment of the solution.

We use the best of the machines considered (i7, 32 GB RAM, SSD disk) and we put a limit on the number of simultaneous processes. Once the maximum number of simultaneous processes is reached, when one process finishes another one is started. P+20 means that P is the number of simultaneous processes started and 20 are the additional processes that have to be executed when other process finish. For example 60+20 with a delay of 0 seconds means that 60 simultaneous processes are run and when one of those processes is finished a new one is started, until running a number of 20 additional processes. We show the case of P+20 in Table 12, Table 13 and Figure 24.

Table 12. Timings ITU-R BT1789 in seconds - disk SSD - i7 – RAM memory 32 GB - P+20 (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES				
TOTAL NUMBER OF PROCESSES	0"	30"	60"	120"	180"
3 + 20	2889,72	2851,38	2878,90	2927,28	2988,80
5 + 20	2868,46	2865,98	2906,80	3027,88	3246,46
10 +20	3425,74	3405,78	3419,35	3574,13	4072,60
20 + 20	4576,64	4533,56	4558,15	4814,48	5850,12
40 + 20	6928,32	6790,45	6800,25	7193,02	9458,55
60 + 20	9391,04	9054,64	9075,23	9591,88	13048,9
80 + 20	11609,4	11320,7	11341,0	11994,6	16676,6
Average per process	116,451	114,030	114,469	120,456	154,587

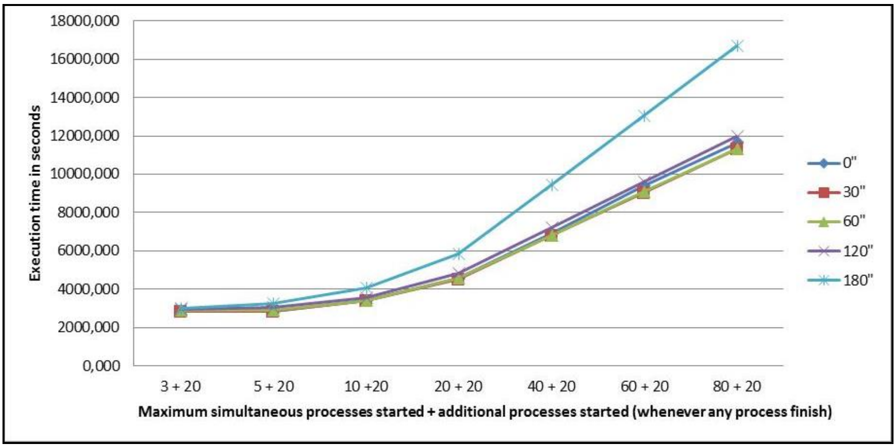


Figure 24. Timings ITU-R BT1789 in seconds - disk SSD - i7 - RAM memory 32 GB - P+20 (packet losses percentage 1%)

Table 13. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 – RAM memory 32 GB - P+20 (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES				
TOTAL NUMBER OF PROCESSES	0"	30"	60"	120"	180"
3 + 20	125,640	123,973	125,170	127,273	129,948
5 + 20	114,739	114,639	116,272	121,115	129,859
10 +20	114,192	113,526	113,978	119,138	135,754
20 + 20	114,416	113,339	113,954	120,362	146,253
40 + 20	115,472	113,174	113,338	119,884	157,643
60 + 20	117,388	113,183	113,440	119,899	163,112
80 + 20	116,094	113,208	113,410	119,946	166,767
Average per process	116,849	115,006	115,652	121,088	147,048

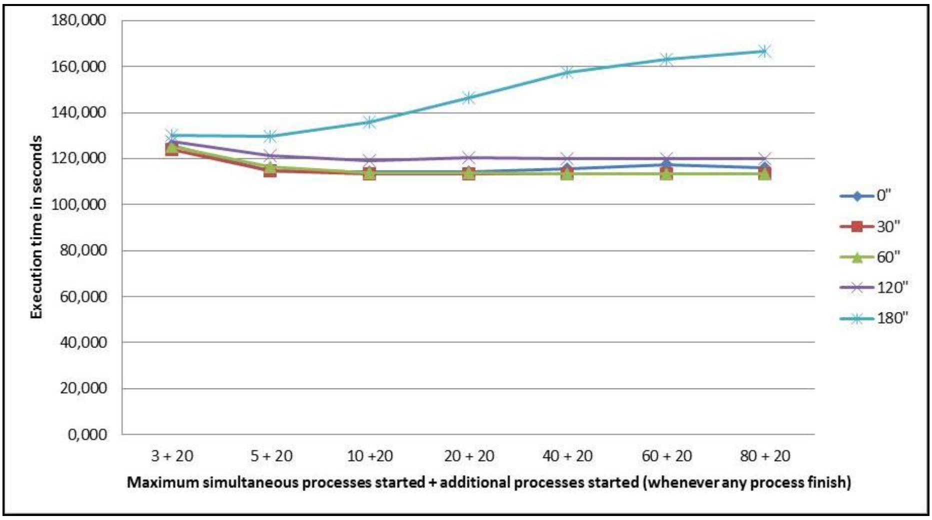


Figure 25. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 - RAM memory 32 GB - P+20 (packet losses percentage 1%)

The Figure 25 is very interesting because it shows that the best case is having 40 processes with a delay of 30 seconds between start of processes. We show in Figure 26 the case in a detailed view using the tool developed.

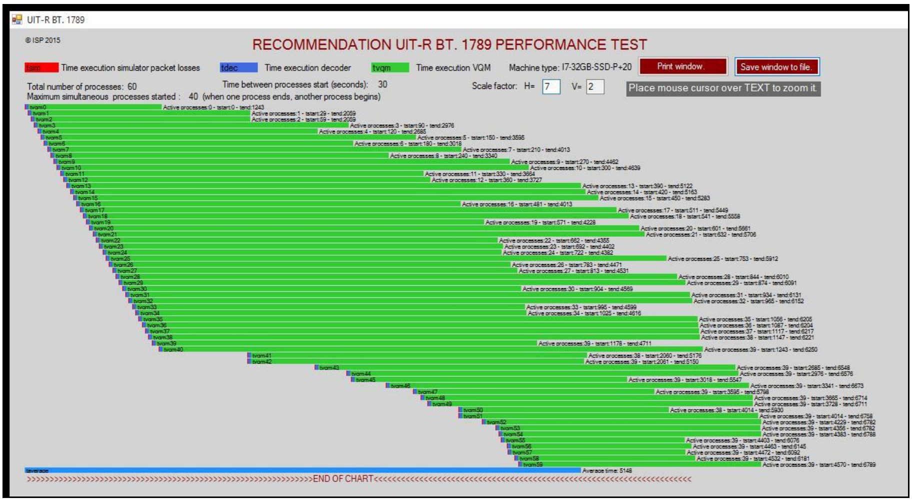


Figure 26. Best case ITU-R BT1789 - disk SSD – i7 - RAM memory 32 GB (packet losses percentage 1%), P+20

Although this case is interesting, we think the case shown in the next section, P+60 is more interesting, because it reflects how the system works having a continuous load of work.

<u>Measurements of time for the packet loss simulator, decoder and VQM metric executed as in the ITU-R BT.1789 for i7, 32 GB RAM, SSD disk with defined maximum number of simultaneous processes, mode P+60</u>

This case shows the same than the previous section (see it for more explanation) but with two differences: the time between start of processes is always 0 seconds, and the number of additional processes is 60, reflecting a high load of the system. The case P+60 is shown in Table 14, Table 15 and Figure 27.

Table 14. Timings ITU-R BT1789 in seconds - disk SSD - i7 – RAM memory 32 GB - P+60 (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES	Average per process
TOTAL NUMBER OF PROCESSES	0"	
3+60	7832,544	124,326
5+60	7445,071	114,540
10+60	7898,188	112,831
20+60	9071,808	113,398

40+60	11442,417	114,424
Average per process	115,582	

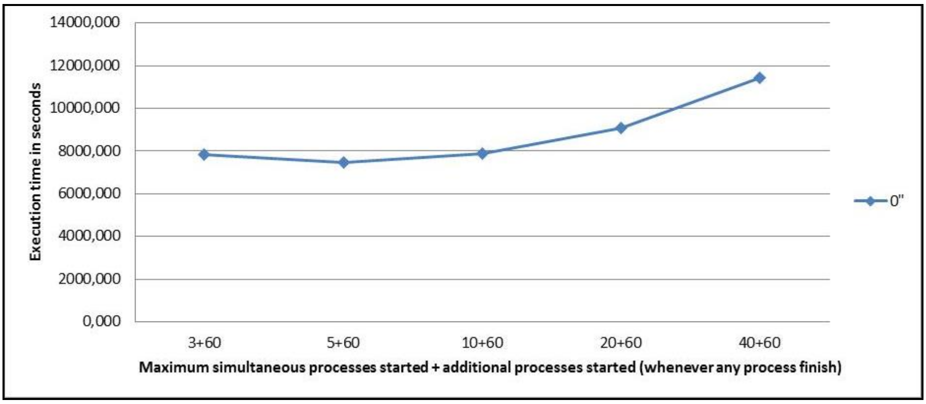


Figure 27. Timings ITU-R BT1789 in seconds - disk SSD - i7 - RAM memory 32 GB - P+60 (packet losses percentage 1%)

Table 15. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 - RAM memory 32 GB - P+60 (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES	Average per process
TOTAL NUMBER OF PROCESSES	0"	
3+60	124,326	124,326
5+60	114,540	114,540
10+60	112,831	112,831
20+60	113,398	113,398
40+60	114,424	114,424
Average per process	115,904	

The conclusion is that in order to run the ITU-T recommendation with the VQM metric, the ideal is to have always 10 simultaneous processes being executed. The time per process would be of 112,831 seconds.

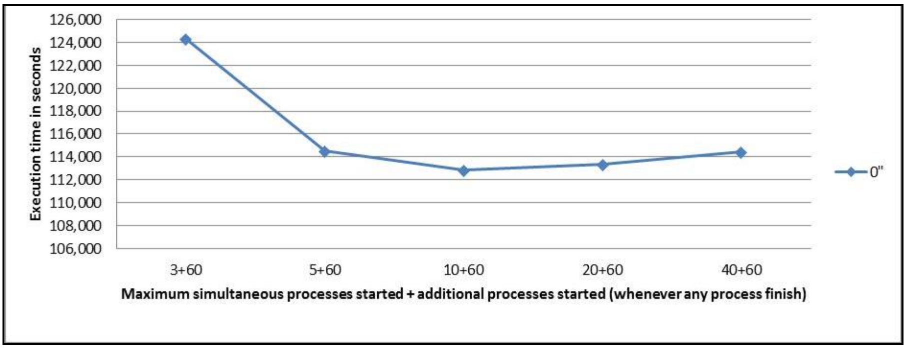


Figure 28. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 - RAM memory 32 GB - P+60 (packet losses percentage 1%)

We show the best case in Figure 29.

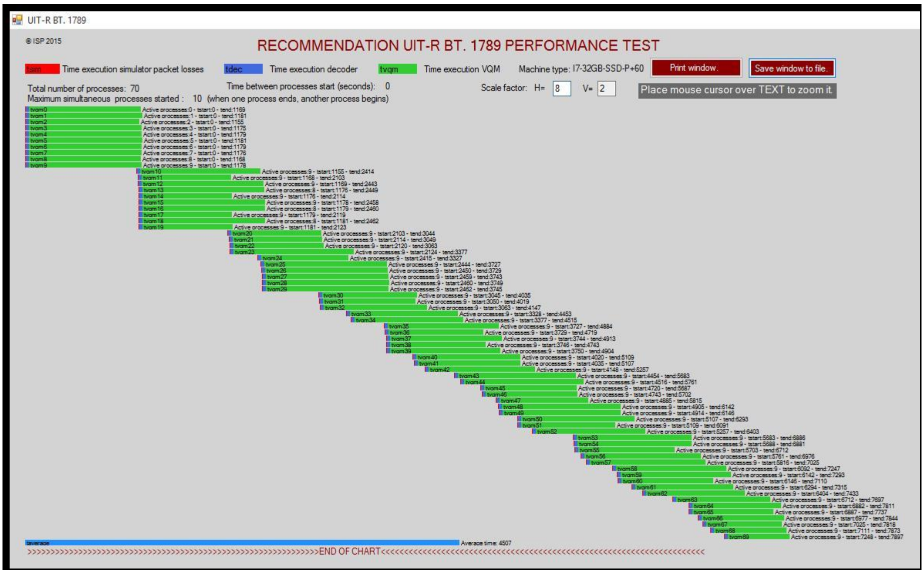


Figure 29. Best case ITU-R BT1789 - disk SSD – i7 - RAM memory 32 GB (packet losses percentage 1%), P+60

We can see that using this approach we arrive to a rather stable situation, where the number of simultaneous processes is stable, having therefore a stable situation regarding the consumption of CPU, RAM memory and disk transfer capacity. We select this case as the one that should be used in a real deployment.

Regarding the time needed for execution per process 112,831 seconds, it means that the quality of the users cannot be monitored continuously in an IPTV network using only a full-reference metric. The solution is to use for example the packet loss rate information or any other no-reference or reduced-reference metric and only analyse the interesting situations for the IPTV provider (where there is doubt and interest) with the full-reference metric. The solution is scalable because the total number of users can be divided in groups and each group of users monitored in a different server. As future lines it would be interesting to explore how many users could be covered using other type of solutions, such as cloud computing, the use of GPUs or porting the code to other language different than MATLAB.

3.3.3 Tool developed to visualize results

We developed a tool with the purpose of understanding better the performance evaluation results shown in the previous section.

This tool, when executed, shows different options regarding which results are to be displayed.

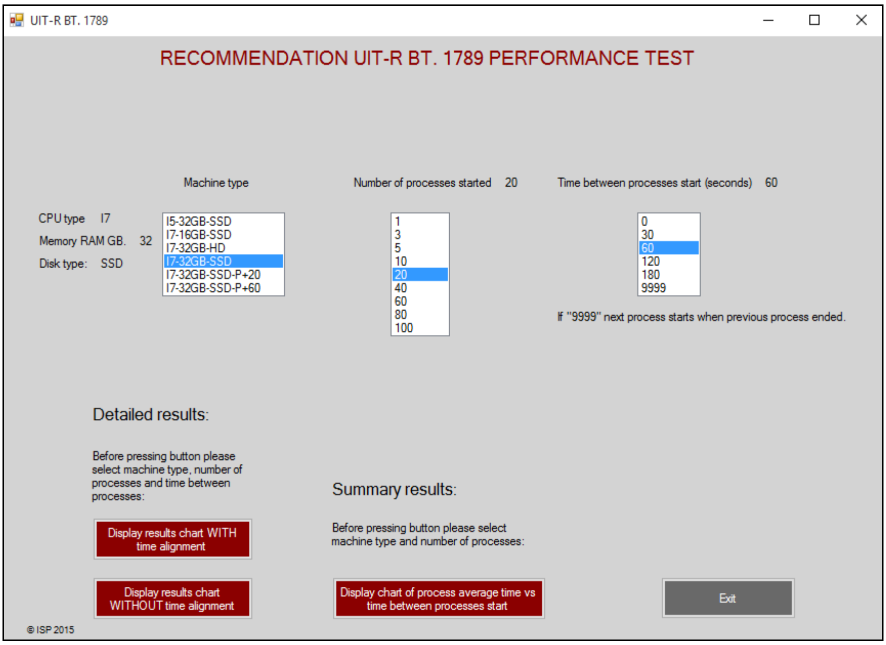


Figure 30. Developed tool to visualize results from the analysis of the recommendation UIT-R BT.1789 (main entrance screen)

In the screen shown in Figure 30 we select which results we want to see, and then we have three buttons. The first one is “display results chart with time alignment”. It means the starting time of the process is reflected in the graph. An example is shown in the Figure 31.

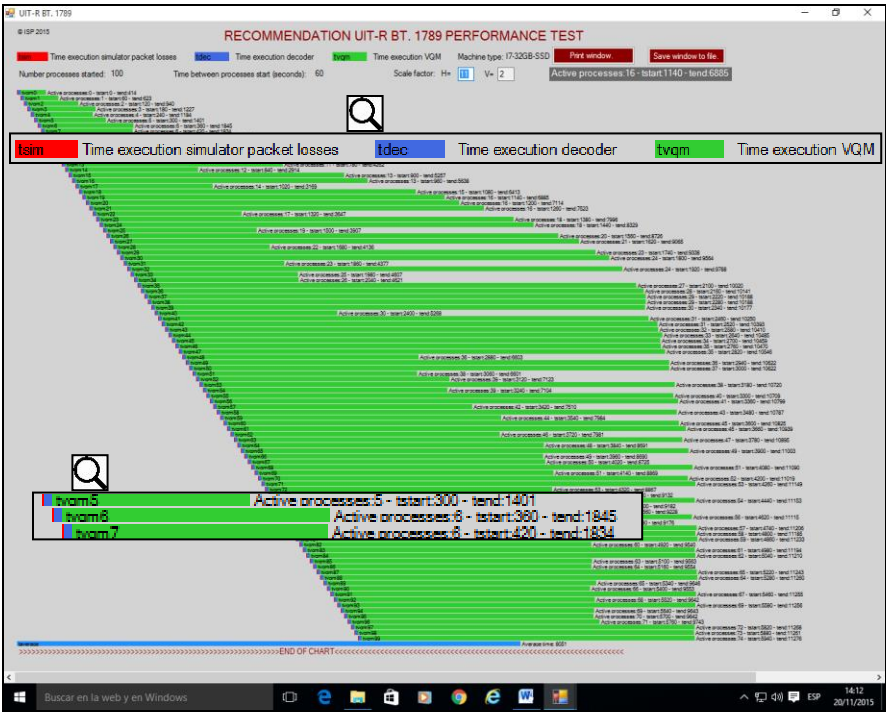


Figure 31. Developed tool to visualize results from the analysis of the recommendation UIT-R BT.1789 (detailed view of execution of processes in one simulation with time alignment)

In the Figure 31, each bar corresponds to the time spent by each process. The execution time of the packet loss simulator is represented in red color, the execution time of the decoder is plotted in blue and the execution time of the metric VQM is shown in green color. Additionally the number of active processes is shown together with the starting and ending time of each process. This type of graph enables the user to have a more clear picture of the evolution of the different tasks.

If we press the button “display results chart without time alignment”, the starting time of the process is not reflected in the figure. We show an example in the Figure 32.

Finally, if we press the button “display chart of process average time vs time between processes start”, the correspondent graph showing that relation is shown. An example is shown in Figure 33.

There is one additional button to send the window to a printer and another additional button to save the window to a file.

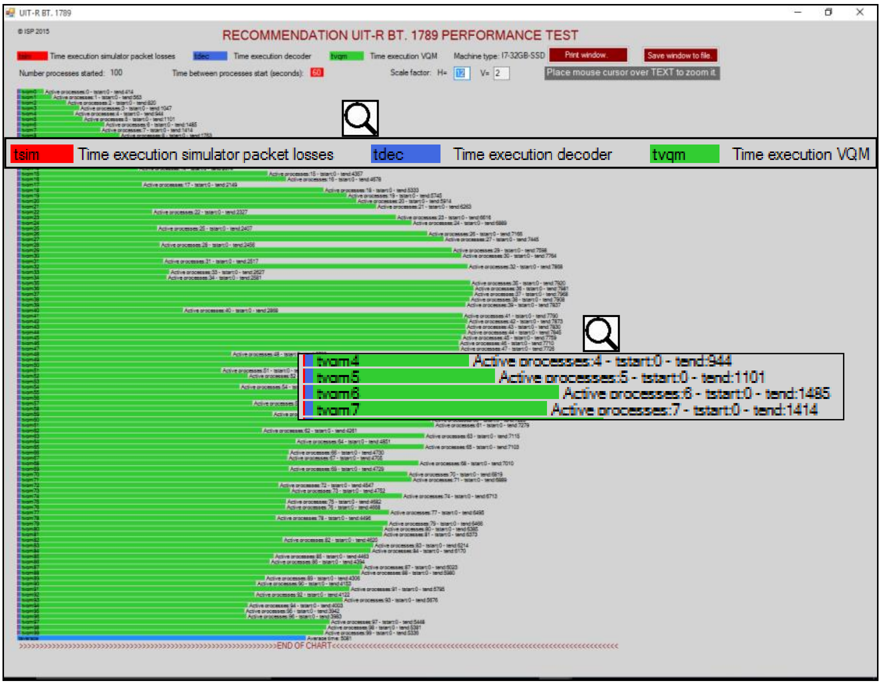


Figure 32. Developed tool to visualize results from the analysis of the recommendation UIT-R BT.1789 (detailed view of execution of processes in one simulation without time alignment)

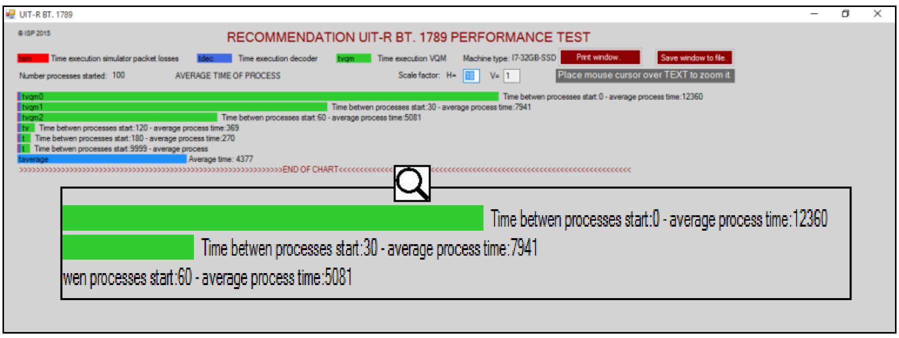


Figure 33. Developed tool to visualize results from the analysis of the recommendation UIT-R BT.1789 (detailed view of a simulation with the average process time vs time between processes start)

In the Figure 33, the objective was to show the relation between the average process time and the time between processes start. As we can see when the time between processes start is increased the average process time is decreased, because the system is less overloaded. The average process time is divided in three parts. The average execution time of the packet loss simulator is plotted in red, the execution time of the decoder is represented in blue, and the execution time of the metric VQM is shown in green color.

3.3.4 Full-reference metric applied in receiver

The alternative to monitoring the quality of IPTV networks applying full-reference metrics on the transmitter is to apply full-reference metrics on the receiver. The solution is shown in the Figure 34.

In this solution the receiver detects which packets have been lost and transmits that information to the transmitter. The transmitter sends only the lost packets to the receiver. The receiver is then able to reconstruct the original undistorted video, and can apply a full-reference metric such as VQM to estimate the quality of the video received. Finally, the video quality estimation is sent to an IPTV network monitoring server which could be located in the transmitter or elsewhere.

In this solution the application of the video quality metric, which as we have seen in the previous section is the most consuming process of video quality monitoring in an IPTV network, is done in the receiver. This enables the continuous simultaneous monitoring of all the users. However, it has the drawback that additional bandwidth from IPTV provider to user is needed in order to transmit the lost packets. This solution depicted is only a proposal we considered during the development of the thesis. It was discarded because of the drawback already mentioned: the high bandwidth needed for the transmission of the lost packets. This is shown in Figure 34.

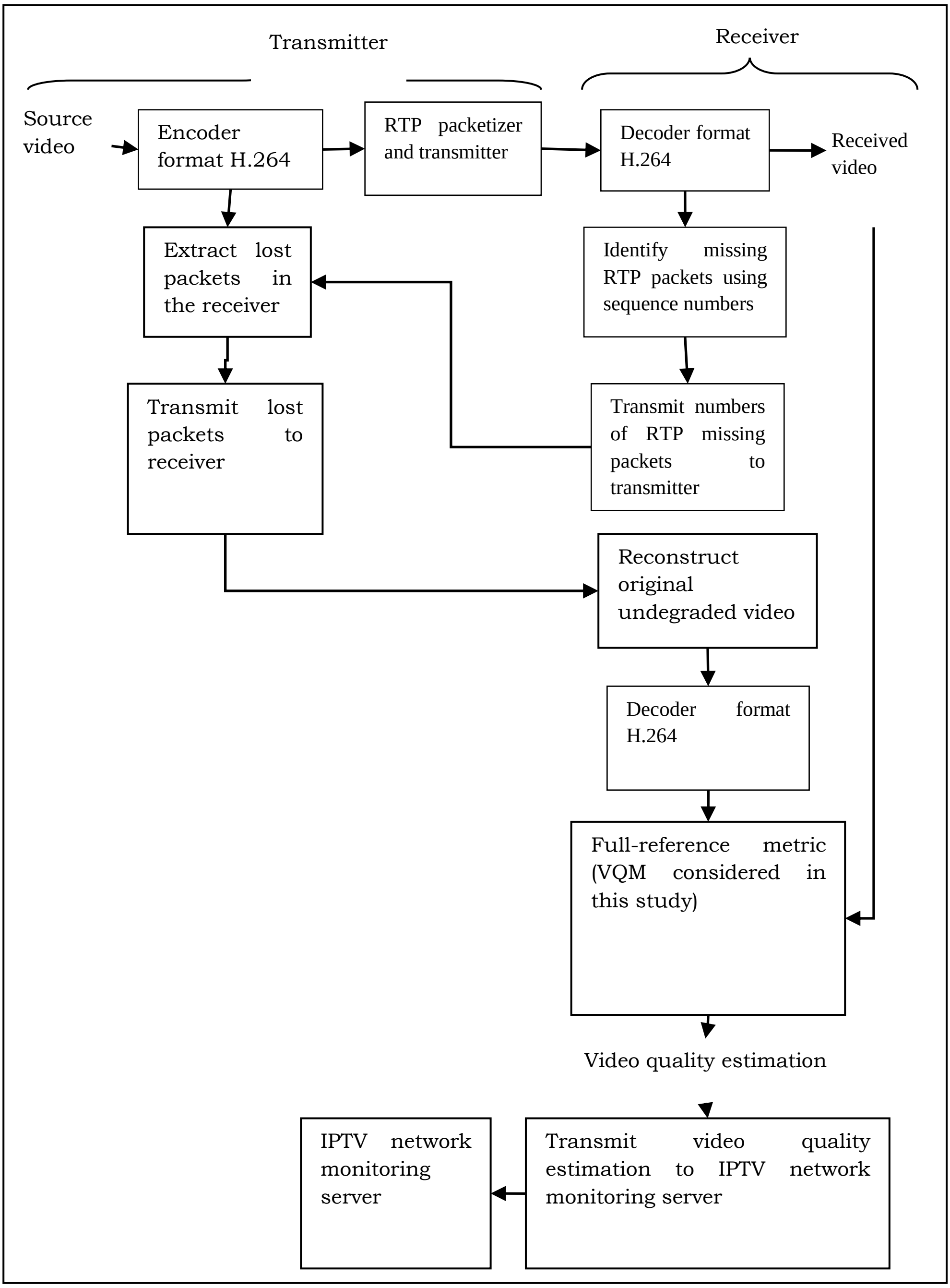


Figure 34. Practical application of assessing IPTV network quality applying full-reference metrics on the receiver

Chapter 4
Development of no-reference metric

In this section we propose a strategy to develop no-reference models, we build a training database and develop a no-reference model (includes validation).

Figure 35 illustrates the proposed strategy to develop no-reference models. This strategy has the purpose of speeding up the development of no-reference models.

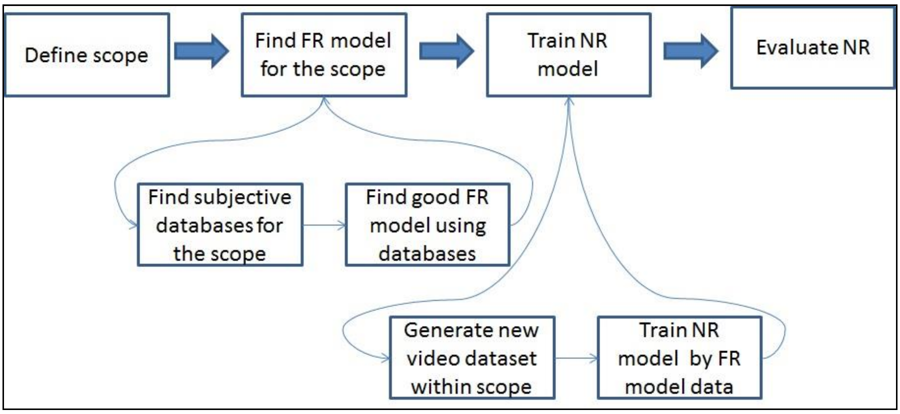


Figure 35. Methodology proposed to develop no-reference models

The first step of the strategy proposed is to define the scope of the model to be developed. If the scope is too wide, the accuracy of the model will be usually lower. Therefore, it is recommended to select the scope as narrow as possible taking into account where the model is going to be applied. In our example, the scope was the transmission distortions caused by packet losses in a communication over UDP. Also, in our example the video taken as reference is compressed, instead of using an original uncompressed video. This is due to the fact that the model to be developed is intended to be used for evaluating transmission distortions only, and not compression distortions.

Once the scope has been defined, the next step is to find a full-reference model for the scope. This step can include the search of subjective databases for the scope and evaluating several full-reference metrics in those databases in order to find the best full-reference model. In our example we performed all these steps, we evaluated several full-reference metrics on three different databases until we got to the conclusion that the metric VQM was the best for our purpose. These steps have been already described in section 3.1.

Once the full-reference metric for the scope has been identified, we use it to develop and validate a no-reference model. In order to do that, we generate two different video dataset with different source videos associated to each dataset. The first video dataset generated is used for development of the model and the second one is used for validation. To develop the datasets, it is necessary to vary all the parameters that we want to include in our final no-reference model. Finally, after evaluating the model with the second dataset, we proceed to make a final proposal of model taking into account both datasets together. This is shown in Figure 36.

One drawback of our strategy is that we will never achieve greater accuracy than the one the full-reference selected metric has. But taking into account that we are developing a no-reference metric and that the full-reference metrics achieve always greater accuracy than no-reference metrics (due to having more information available) we think that the approach of using a full-reference metric for developing a no-reference metric is valid.

Finally it is possible to keep the scope and incorporate new parameters in the model. For that purpose it is necessary to generate a new dataset varying all the parameters, including the new ones, evaluate the new dataset with the full-reference metric and then redesign the model. Comparing the modelling phase of the previous and newer no-reference model one can know if greater accuracy is achieved (observing error in fitting curve). This fact can also be confirmed by evaluating the newer no-reference method with a completely independent video database containing subjective scores and comparing the results with the previous no-reference model.

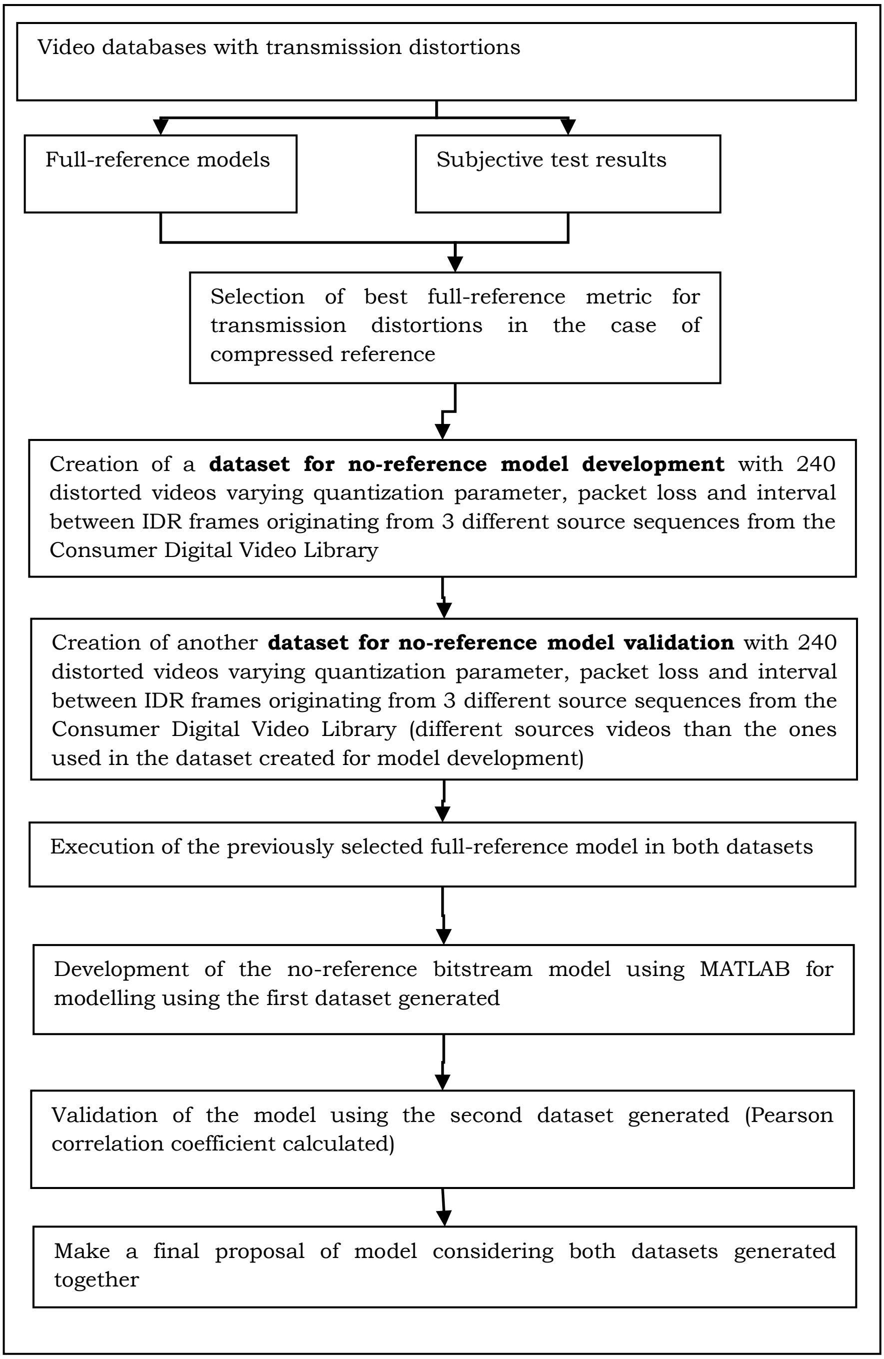


Figure 36. Practical application of the strategy proposed for development of no-reference models

In the following sections we explain:

- The creation of the dataset for no-reference model development.
- The creation of the dataset for no-reference model validation.
- Execution of the VQM full-reference metric in both datasets.
- The development of the no-reference model bitstream model using the first dataset.
- The validation of the no-reference model using the second dataset.
- The development of the final no-reference model considering both datasets.

We will present a lightweight no-reference bitstream method that uses the packet loss rate and the interval between instantaneous decoder refresh frames (IDR frames) to estimate the video quality. IDR frames are 'delimiters' in the stream. After receiving an IDR frame, frames prior to the IDR frame can no longer be used for prediction. As well as this, IDR frames are also I-frames, so they do not reference data from any other frame. This means they can be used as seek points in a video. The no-reference bitstream model was fitted using several videos from the Consumer Digital Video Library (CDVL) database [75] and the VQM metric. Then, it was validated with other different videos from the Consumer Digital Video Library (CDVL) database and the VQM metric. The VQM metric has been used to train the no-reference bitstream model regarding only the transmission distortions, and no compression distortions such as QP have been taken into consideration because it has been shown that VQM is very accurate when only transmission distortions are considered using a compressed reference. The case where VQM is used to measure a combination of compression and transmission distortions (for example, different QP and packet loss rate with uncompressed reference) is not evaluated.

4.1 Construction of the training and validation databases

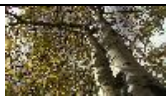
4.1.1 Creation of the training and validation databases

Six sequences with resolution 1,920 × 1,080 pixels were downloaded from the Consumer Digital Video Library (CDVL) database [75], with different characteristics. In five of the videos, the final part was removed to generate videos of a total length of 17s at 30 fps. One of the sequences had a total length of 14s at 25 fps. The SRC, listed in Table 6, were selected to spread a large variety of different content in progressive Full-HD 1,920 × 1,080 format. The videos were converted from YUV packed 4:2:2 to YUV planar 4:2:0. The videos were compressed with the Quantization Parameter set to 26, 32, 38 and 44. In order to make sure the no-reference model is valid for the different compression qualities but not dependent on the compression qualities (QP is not in the model) the QP has been set to 26, 32, 38 and 44.

The six sequences were divided in two groups of three sequences. The first group is used as training database for model development while the second one is used for validation purposes. After validation, a final proposal of model is done considering all six source sequences. The list of SRCs used in model development and validation is shown in Table 16.

Table 16. List of SRCs used in model development and validation

SRC	Thumbnail	Description	Name of the sequence in CDVL
1-model development		Woman smoking and people on a street, high contrast in the rock	NTIA outdoor mall with tulips (3e)
2-model development		Kayaking, scene changes, fast moving water	NTIA Red Kayak

3-model development		Trees, leaves, short and numerous movements in most of the image, scene changes	NTIA Aspen Trees in Fall Color, Slow Scene Cuts
4-model validation		Mountain with snow and moving fog in a sunny day, high brightness, scene changes	NTIA Snow Mountain
5-model validation		Global view of a city, buildings, scene changes, rather static	NTIA Denver Skyscrapers
6-model validation		Two people speaking in at the table and showing an electronic device	NTIA Front End (Part of a Longer Talk)

4.1.2 Execution of VQM metric in the training and validation databases

We selected the VQM metric to develop a no-reference bitstream model because of the very good performance shown in section 3.1. However, the performance of VQM in the case of compressed reference has been only tested in the case of compressed reference of high quality, which may not correspond to a QP value of 44. This causes a small degree of uncertainty in the obtained results because the scenario in which the compressed reference has low quality remains to be verified. The parameter keyint in the x264 encoder, corresponding to the interval between IDR frames, was set to 12, 36, 60 and 84. The maximum slice size was set to 1400 bytes. Those values were selected because they are very common in a video coding context.

We consider that the keyint parameter is important since the distortion due to a packet loss propagates until the next IDR frame. Thus a higher value implies more error propagation and lower video quality. Finally the packet loss rate was set to 0.1%, 1%, 3%, 5%, and 10%. In total, 6 × 4 × 4 × 5 = 480 distorted videos were evaluated using the VQM metric. The videos were encoded with the x264 encoder [70], random packet losses were inserted using a packet loss simulator [71] and the videos were decoded with the ffmpeg decoder. The ffmpeg decoder produces incomplete video files when random packet losses are inserted.

To be able to apply the VQM metric, the videos were reconstructed so that they have the same length as the original. The reconstruction was done in two steps. First, the frame numbers were inserted into the luminance information of the uncompressed original sequence. After decoding the videos, the frame numbers were read and used to identify the missing frames and reconstruct the decoded video. The reconstruction method is explained in detail in [76] as well as in this thesis.

The framework is described in Figure 37. As it can be seen in the figure, the VQM metric was applied (after conversion to packed 4:2:2 format) between the compressed reference (video compressed and then decompressed) and the reconstructed video. We used the same version of VQM than in the previous sections. The same parameters as in the previous sections were used for VQM.

4.2 Development of the model

Our objective was to develop a lightweight model to predict the quality of the video as a function of two parameters: packet loss rate in percentage, denoted p, and interval between IDR frames in number of frames, denoted I.

Once the VQM metric was applied to all the distorted videos in both training and evaluation datasets, we proceeded to plot the relationship between the VQM measurements and the interval between IDR frames grouping by different packet loss rates, in order to see what model could fit better to the data. For doing this plot we considered the training dataset.

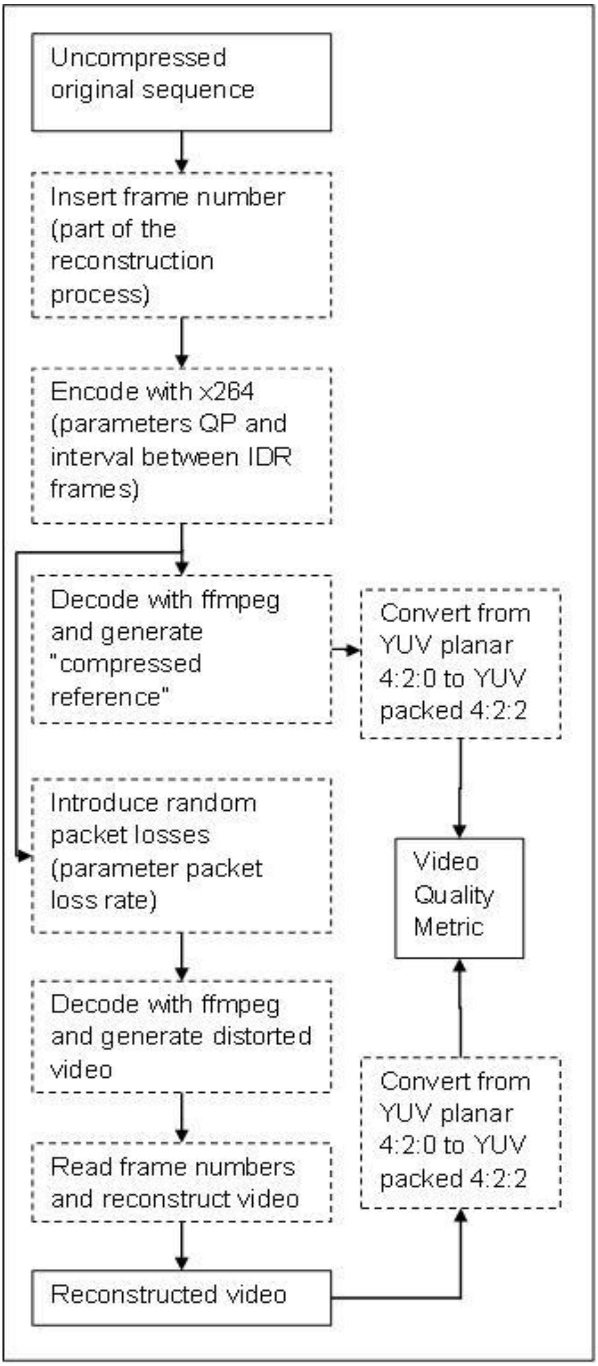


Figure 37. Framework for model development and validation

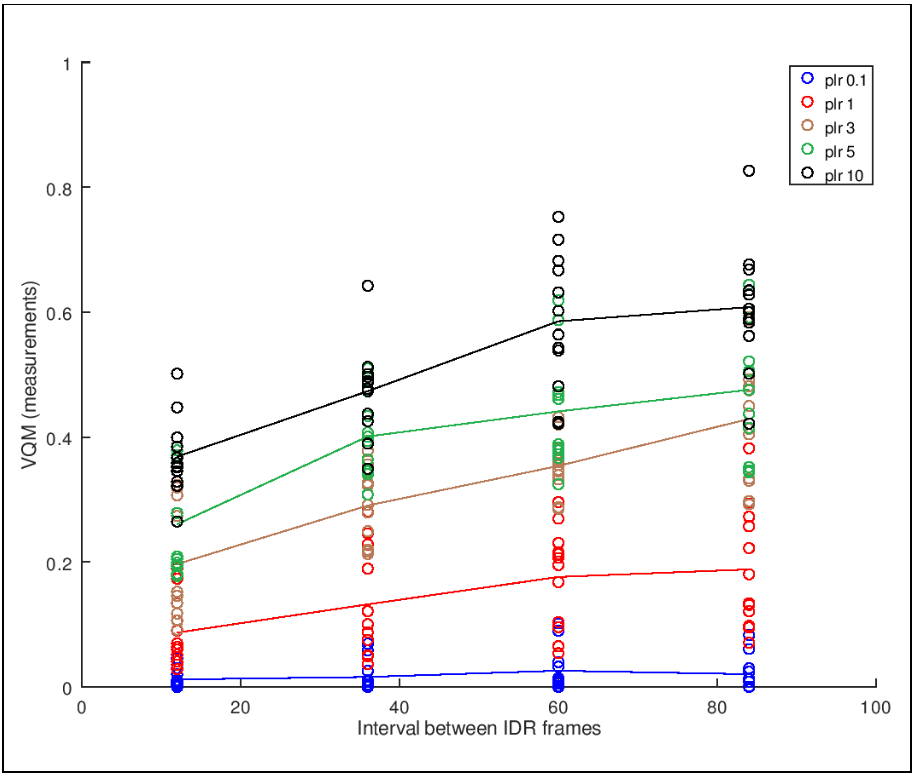


Figure 38. Training database VQM measurements as a function of interval between IDR frames and grouped by packet loss rates.

As it can be appreciated in Figure 38, the maximum value of VQM increases with an increase in the interval between IDR frames. A higher value of VQM implies worse quality of video.

Then we proceed to visualize the relationship between VQM measurements and packet loss rate grouped by interval between IDR frames. This is shown in Figure 39.

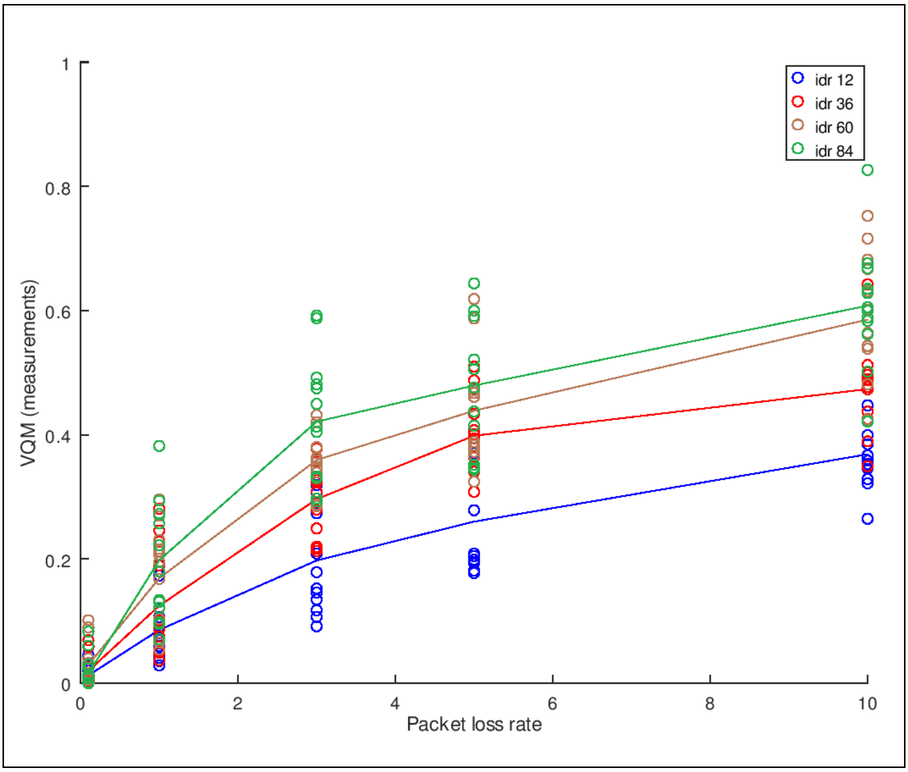


Figure 39. Training database VQM measurements as a function of packet loss rate and grouped by interval between IDR frames.

Analyzing these two figures, we come to the conclusion that a polynomic fitting of third order could be a good approximation in order to develop the model.

The packet loss rate shows a more direct relationship with VQM than the interval between IDR frames. A non-linear model of the following type could be appropriate:

$$VQMpred = b_0 + b_1 * I^3 + b_2 * I^2 + b_3 * I + b_4 * p^3 + b_5 * p^2 + b_6 * p \quad (4)$$

Where p is the packet loss rate in percentage and I is the interval between IDR frames in number of frames.

The MATLAB function polyfitn was used to calculate the coefficients of (4).

With the non-linear fit, we obtained the following no reference bitstream model for the predicted quality, f(I,p), shown in (5).

$$f(I,p) = -1{,}36e\text{-}01 - 3{,}11e\text{-}08 * I^3 - 1{,}73e\text{-}05 * I^2 + 4{,}12e\text{-}03 * I + 1{,}14e\text{-}03 * p^3 - 2{,}26e\text{-}02 * p^2 + 1{,}63e\text{-}01 * p \quad (5)$$

In order to see how good the model fits the data, we have plotted in Figure 40 the model showing VQM as a function of interval between IDR frames for different packet loss rates together with the median values obtained from the measurements.

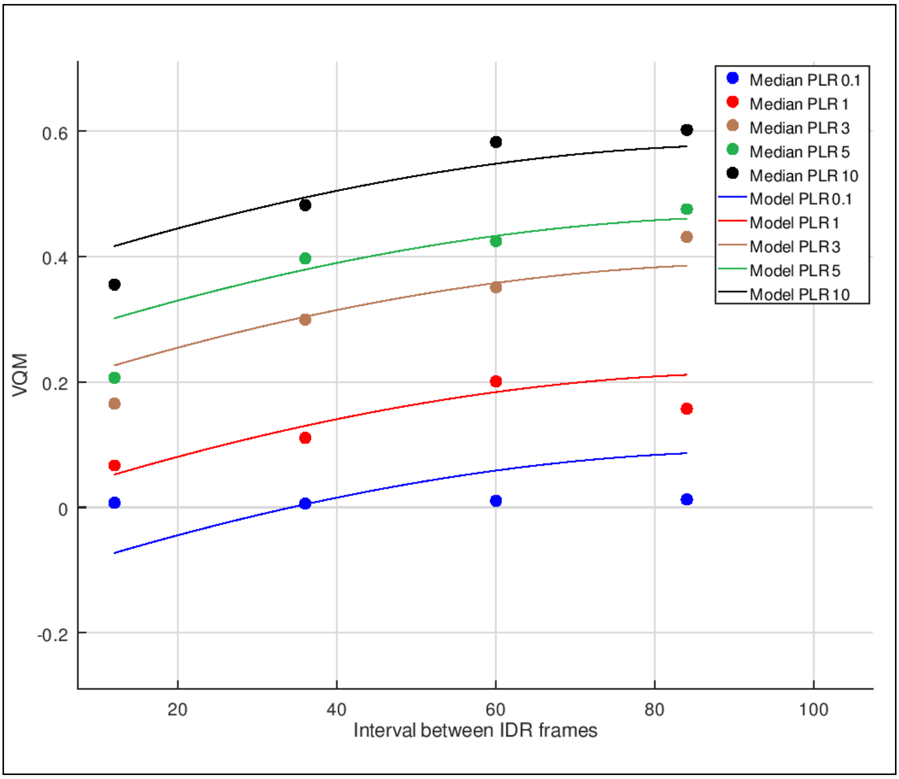


Figure 40. Model developed and medians from measurements using the training dataset. VQM as a function of interval between IDR frames.

We also want to check if the model developed fits well when considering the other perspective, that is the VQM as a function of packet loss rate for different interval between IDR frames. In the graph we also plot the medians from the measurements. Different curves have been generated corresponding to different values of the parameter interval between IDR frames (from 12 to 84), as it can be seen in Figure 41. All of this is done using the training dataset.

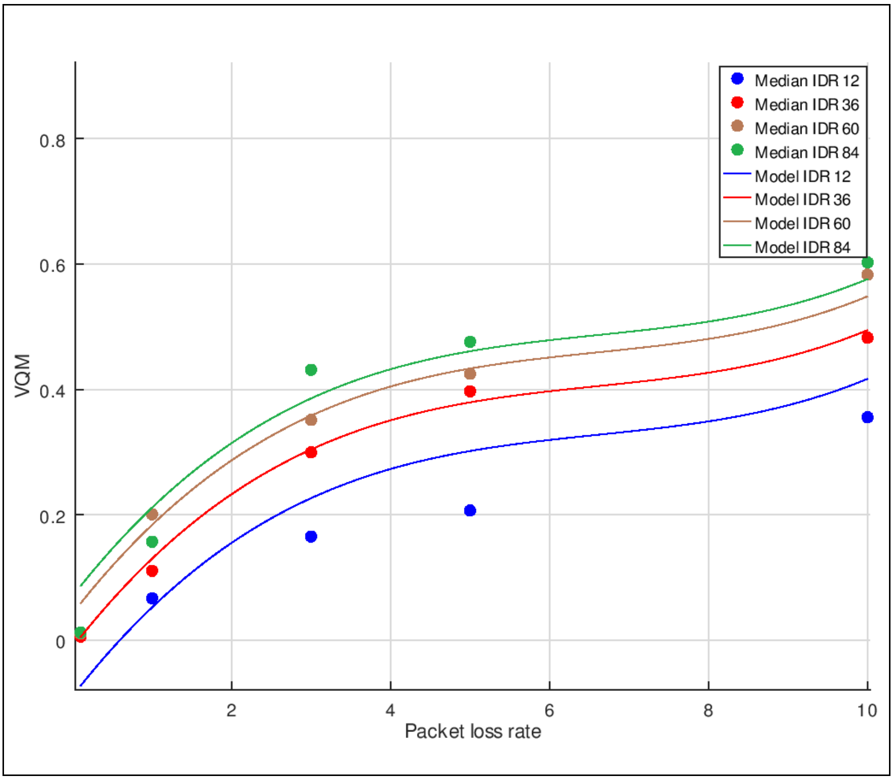


Figure 41. Model developed and medians from measurements using the training dataset. VQM as a function of packet loss rate.

The developed no-reference metric can be used as a base for further developments of no-reference metrics extending its scope and number of parameters involved.

4.3 Validation of the model

We applied the model generated to the dataset created for validation (completely independent video sources sequences from the ones used for generation of the model). Therefore, we obtained the predicted values by the model for the validation dataset.

Previously we had also applied the full-reference metric VQM to the validation dataset, obtaining the VQM measurements.

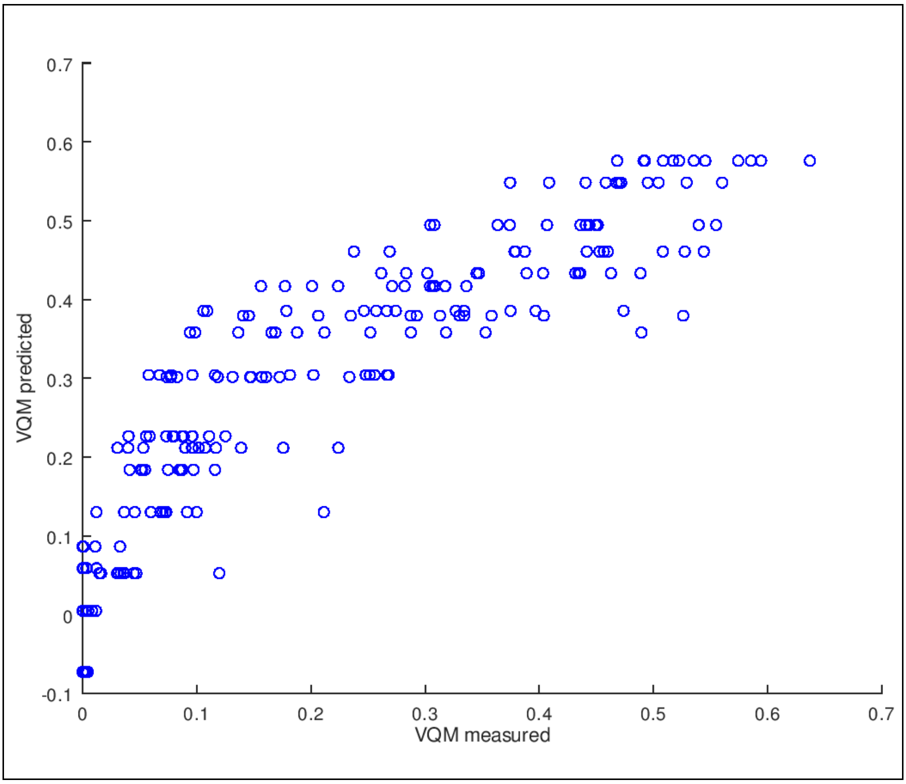


Figure 42. Scatter plot representing VQM predictions by the model (VQM predicted) against VQM obtained directly applying the full-reference metric VQM (VQM measured) in the validation dataset.

In Figure 42 both the VQM values predicted by the model developed and the ones obtained by applying the full-reference metric VQM are plotted. In order to know the exact of correlation we computed the Pearson correlation coefficient using MATLAB between the VQM predicted values and the VQM measured values, obtaining a value of **0.89** for the **Pearson** correlation coefficient.

We believe that the model can be improved by adding new parameters and improving the fitting function used. The important fact is that these results validate the methodology followed in order to develop a no-reference bitstream model.

4.4 Final model proposal

Finally, we merge both dataset, the one that was used for training and the other that was used for validation and using all the data we fit the model again and make a final proposal.

$$f(I,p) = -1{,}56e\text{-}01 + 2{,}93e\text{-}07 * I^3 - 6{,}46e\text{-}05 * I^2 + 6{,}04e\text{-}03 * I + 4{,}65e\text{-}04 * p^3 - 1{,}16e\text{-}02 * p^2 + 1{,}16e\text{-}01 * p \quad (6)$$

4.5 Conclusions about the methodology proposed and the model developed

We have proposed a strategy for developing new no-reference objective video quality metrics by using well performing full-reference video objective quality metrics to reduce the development time. The starting point is to define a relatively narrow scope. Find a FR model to create a big training database by varying the parameters that will be present. Finally train the NR model on this database.

This strategy is illustrated on the scope of transmission distortions in the case of compressed reference. As a first step, we have evaluated six publicly available full-reference metrics using three freely available video databases. The main objective of the evaluation was to compare the performance of the metrics when transmission distortions in the form of packet loss were introduced. The results show that VQM performs very well in all the video databases, being the best metric among the studied in the HDTV video database (uncompressed reference) and in the LIVE Wireless video database. In the EPFL-PoliMI database, SPATIAL MOVIE performed best and TEMPORAL MOVIE performed worst. When transmission distortions are evaluated, using the compressed video as the reference provides greater accuracy than using the uncompressed original video as the reference, at least for the studied metrics.

Furthermore, to demonstrate the suggested strategy of model development, we present a no-reference bitstream model trained and optimized using full-reference model evaluation. The objective of the model is to accurately enough predict the video quality when transmission distortions are introduced. We fit the model using videos from the Consumer Digital Video Library (CDVL) database and the VQM metric. Then, the model is validated using different videos from the Consumer Digital Video Library than the ones used for the development of the model. Once the model has been developed and evaluated successfully (pearson correlation 0.89), we make a final proposal of the model using all the sequences (sequences used for training and evaluation are merged together in order to make a final model proposal).

Chapter 5 Conclusions and future lines

5.1 Summary and contributions

We summarize in this section the results presented in this thesis and we highlight the most relevant contributions.

In *Chapter 2 State of the art* the major contributions are generation of documentation about basics, objective metrics (contribution of application to extract estimated MOS processing XML file from VQEG tools) and publicly available subjective databases.

In *Chapter 3 Selection and application of full-reference metric* the following achievements are to be highlighted:

- Comparison between full-reference publicly available metrics.
- Development of an application to reconstruct incomplete decoded videos.
- Analysis of the practical application of the recommendation ITU-R BT.1789. This included the replication of the scenario depicted on the recommendation, the realization of performance tests (memory, CPU, disk transfer) in different machines and the development of a tool to display results graphically.

In *Chapter 4 Development of no-reference metric* we built training and validation databases. In this chapter another achievement was the development and validation of no-reference bitstream real-time video quality model based on the full-reference metric VQM.

We can extract the conclusion that **the IPTV monitoring based on full-reference metrics on the transmitter in the presence of**

packet losses can be done only when combined with other type of evaluation (such us no-reference bitstream, hybrid or even reduced-reference), **in such a way that only videos which quality is subject to doubt and interest are evaluated with the full-reference video quality metric**. This conclusion is derived from the fact that the best time per process (in the case of 70 processes) is 112,831 seconds in a machine with the following characteristics: i7, 32 GB RAM, SSD disk. Probably the metric VQM can be optimized to run faster (or even real-time) and also the technology will evolve quickly in the future, reducing drastically the time needed per process. In addition to that, it is possible to use more than one server to compute the quality of the receivers. But for the time being, using the full-reference metric on the transmitter to monitor IPTV quality of the receivers needs the combination of another evaluation mechanism, such as for example a no-reference bitstream, hybrid or reduced-reference method.

From the study of the applicability of full-reference metrics on the receiver to compute the quality of the receivers in an IPTV network we can deduce that it can be applied already nowadays, because it has the advantage of de-centralized processing of video quality (it is done on the receivers). However it has the drawback that it requires additional bandwidth in the network and the use of dedicated unicast connections.

There is still much work to do in order to develop an accurate no-reference video quality metric with wide scope of application. However, we hope our contributions will enable a better understanding of the problems involved and will provide some solutions to speed up the development process. In the thesis we provide **tools** (quick processing of huge XML file generated within the VQEG tools; tool to reconstruct incomplete decoded videos for use in objective quality metrics), **implementations** (a no-reference models for transmission distortions) and **methodologies** (strategy to develop no-reference models based on full-reference models).

One of the conclusions is that **it is possible to use full-reference metrics to speed up the development of no-reference metrics**. In this thesis we explain how this process can be achieved.

We also provided valuable knowledge about performance of current full-reference metrics in the case of compressed reference and transmission distortions. The result of the analysis showed that **VQM was the best metric among the studied** for such scope.

The second conclusion that can be extracted from the analysis is that **using a compressed reference provides more accuracy than using an uncompressed reference** when transmission distortions are studied (and not compression distortions). But this second conclusion should be confirmed by further studies.

5.2 Dissemination

The **dissemination of results** was done by publications in **international peer-reviewed journals and scientific conferences.**

Paper nº1: **Sedano, I.,** Brunnström, K., Kihl, M., Aurelius, A., “Full-reference video quality metric assisted the development of no-reference bitstream video quality metrics for real-time network monitoring”, EURASIP Journal on Image and Video Processing 2014 2014:4. Cited by 13.

Summary: we propose a method to speed up the development of no-reference bitstream objective metrics for estimating QoE. This method uses full-reference objective metrics, which makes the process significantly faster and more convenient than using subjective tests. In this process, we have evaluated six publicly available full-reference objective metrics in three different databases, the EPFL-PoliMI database, the HDTV database, and the Live Video Wireless database, all containing transmission distortions in H.264 coded video. We show statistically that the full-reference metric Video Quality Metric (VQM) performs best considering all the databases. Further, we use VQM to train a lightweight no-reference bitstream model, which uses the packet loss rate and the interval between instantaneous decoder refresh frames, both easily accessible in a video quality monitoring system.

Paper nº2: **Sedano, I**.; Kihl, M.; Brunnstrom, K.; Aurelius, A., ”Reconstruction of incomplete decoded videos for use in objective quality metrics”, Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on, pp.376,379, 11-13 April 2012. Cited by 1.

Summary: Many full-reference objective metrics require that the original and the degraded video contain the same number of frames. Most codecs are not able to decode the video properly when the video is subject to packet losses and produce incomplete video files with lower number of frames than the original video. In this paper we present a simple method to reconstruct the degraded videos so that it has the same length as the original. Information regarding the frame numbers is inserted into the original uncompressed video and used later on to identify the missing frames and reconstruct the degraded video.

Paper nº3: **Sedano, I.**; Kihl, M.; Brunnstrom, K.; Aurelius, A., "Evaluation of video quality metrics on transmission distortions in H.264 coded video", Broadband Multimedia Systems and Broadcasting (BMSB), 2011 IEEE International Symposium on, pp.1,5, 8-10 June 2011. Cited by 10.

Summary: In this paper we study the accuracy of full-reference objective methods for assessing the quality degradation due to the transmission distortions. We evaluated several well-known publicly-available full-reference objective metrics on the freely available EPFL-PoliMI (Ecole Polytechnique Fédérale de Lausanne and Politecnico di Milano) video quality assessment database, which was specifically designed for the evaluation of transmission distortions. The full-reference metrics are usually evaluated using a reference which is uncompressed. Instead, we study the performance of the metrics when the reference videos are lightly compressed to ensure high quality.

Paper nº4: M. Barkowsky, **I. Sedano**, K. Brunnström, M. Leszczuk, and N. Staelens, "Hybrid video quality prediction: reviewing video quality measurement for widening application scope," Multimed. Tools Appl., vol. 74, no. 2, pp. 323–343, 2015. Cited by 10.

Summary: This paper lists several perceptual artifacts that may be computationally measured in an isolated algorithm and some of the modeling approaches that have been proposed to predict the resulting quality from those algorithms. These algorithms usually have a very limited application scope but have been verified carefully. The paper continues with a review of some standardized and well-known video quality measurement algorithms that are meant for a wide range of applications, thus have a larger scope. Their individual artifacts prediction accuracy is usually lower but some of them were validated to perform sufficiently well for standardization. Several difficulties and shortcomings in developing a general purpose model with high prediction performance are identified such as a common objective quality scale or the behaviour of individual indicators when confronted with stimuli that are out of their prediction scope. The paper concludes with a systematic framework approach to tackle the development of a hybrid video quality measurement in joint research collaboration.

Paper nº5: Staelens, N.; **Sedano, I.**; Barkowsky, M.; Janowski, L.; Brunnstrom, K.; Le Callet, P, "Standardized toolchain and model development for video quality assessment — The mission of the Joint Effort Group in VQEG", Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on, pp.61,66, 7-9 Sept. 2011. Cited by 19.

Summary: In this paper, we introduce the JEG (Joint Effort Group) of VQEG (Video Quality Experts Group) and provide an overview of the different ongoing activities within this newly started group.

Paper nº6: Brunnström, K., Wang, K., **Sedano, I.**, Barkowsky, M., Kihl, M., Aurelius, A., Le Callet, P., and Sjöström, M., "2D no-reference video quality model development and 3D video transmission quality", Proc. 6th Inter. Workshop on Video Processing and Quality Metrics for Consumer Electronics, 2012. Cited by 5.

Summary: In order to decrease the development time of no-reference metrics, it might be possible to use full-reference metrics for this purpose. In this work, we have evaluated six full-reference objective metrics in three different databases. We show statistically that VQM performs the best. Further, we use these results to develop a lightweight no-reference model. We have also investigated users' experience of stereoscopic 3D video quality by performing the rating of two subjective assessment datasets, targeting in one dataset efficient transmission in the transmission error free case and error concealment in the other. Among other results, it was shown that, based on the same level of quality of experience, spatial down-sampling may lead to better bitrate efficiency while temporal down-sampling will be worse.

Paper nº7: Geng Yu; Westholm, T.; Kihl, M.; **Sedano, I.**; Aurelius, A.; Lagerstedt, C.; Odling, P "Analysis and characterization of IPTV user behavior", Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on, pp.1,6, 13-15 May 2009. Cited by 22.

Summary: The purpose of this paper is to characterize IPTV traffic and study end user behavior by analyzing and modeling IPTV traffic collected from a Swedish municipal network. The focus of the measurements was put on Internet Group Management Protocol (IGMP) packets. Apart from the measurement results and analysis, the paper provides background information about the technologies and issues of IPTV. IP multicast which is used for transferring Live TV content is based on the concept of a group. IGMP is used to manage the membership of multicast groups. Based on this information, traffic parameters for analysis were chosen and measured.

Paper nº8: Barkowsky, M.; Staelens, N.; Janowski, L.; Koudota, Y.; Leszczuk M.; Urvoy, M.; Hummelbrunner, P.; **Sedano, I.**; Brunnström, K., "Subjective experiment dataset for joint development of hybrid video quality measurement algorithms", QoEMCS 2012 - Third Workshop on Quality of Experience for Multimedia Content Sharing, Berlin, Germany, 2012. Cited by 6.

Summary: The application area of an objective measurement algorithm for video quality is always limited by the scope of the video datasets that were used during its development and training. This is particularly true for measurements which rely solely on information available at the decoder side, for example hybrid models that analyze the bitstream and the decoded video. This paper proposes a framework which enables researchers to train, test and validate their algorithms on a large database of video sequences in such a way that the often limited scope of their development can be taken into consideration. A freely available video database for the development of hybrid models is described containing the network bitstreams, parsed information from these bitstreams for easy access, the decoded video sequences, and subjectively evaluated quality scores.

Paper nº9: **Iñigo Sedano**, Gorka Prieto, Kjell Brunnström, Maria Kihl, Jon Montalban, "Application of full-reference video quality metrics in IPTV", IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2017, June 2017.

Summary: Executing an accurate full-reference metric such as VQM can take minutes in an average computer for just one user. It is unfeasible to analyze all the videos received by users in an IPTV network for example consisting of 10.000 users using a single computer running the VQM metric. Therefore it would be required to use other type of systems or lightweight no-reference metrics [2] in addition to the full-reference metric mentioned. Lightweight no-reference metrics can be used for discarding potential situations to evaluate because they are accurate enough for that task, and where more accuracy is required the full-reference metric VQM could be used. The main work in this paper is focused on determining how many situations/users can be analyzed using the VQM metric in a computer with good performance. The full-reference metric is applied on the transmitter using a method specified in the recommendation ITU BT.1789 "A method to reconstruct received video using transmission error information for packet video transmission".

5.3 Future lines

In order to enable IPTV monitoring of quality of experience, accurate no-reference models must be developed because they are capable of real-time execution in every host of the IPTV network. The current no-reference models are accurate only if the scope of application is greatly reduced.

In order to achieve greater accuracy, it is necessary to consider more parameters as input to the no-reference model, than the ones typically considered by a pure bitstream no-reference model or a pure pixel-based no-reference model.

The hybrid no-reference models will enable to improve current performance of the models and IPTV monitoring and thus the current research is starting to point in that direction.

A hybrid no-reference model uses both video pixel information in combination with the bit-stream information eventually also doing a full decoding of the video payload.

In the VQEG, the Joint Effort Group has among its objectives the development of accurate no-reference hybrid models by a means of cooperation between different entities [36].

VQEG's JEG is free and open to everyone, both from academia as well as private industries. No subscription fees are involved for joining VQEG JEG. Contributions can be made concerning every step involved in subjective and objective video quality assessment.

As a continuation for this thesis we think the next step would be to develop hybrid no-reference models and start to compare their accuracy in relation with typical bitstream based no-reference models. Then we will be one step closer to achieve real Quality of Experience IPTV monitoring [36].

Annex A

In this Annex, the processing of the subjective scores on the EPFL-PoliMI (Ecole Polytechnique Fédérale de Lausanne and Politecnico di Milano) video quality assessment database [60][61][58] by the author of this thesis is detailed.

The author of this thesis decided to merge the data from the two labs and calculate the DMOS (Difference Mean Opinion Scores) and the confidence intervals.

Prior to the merging, we performed a Student T test considering the overall mean and standard deviation of the raw MOS individual scores of each lab that showed at 95 % confidence level that the data from the two labs were not significantly different. Additionally, the DMOS and confidence interval values (after normalization, screening and re-scaling) were calculated for each content and distortion type, and compared between the two labs. This additional verification confirmed that the data from the two labs were sufficiently similar to be merged. In this process seventy two videos were checked, corresponding to six different packet loss rate, two different channel realizations for each PLR, for each of the six source sequences.

In the scatter plot in Figure 43 the linear correlation between the DMOS values obtained in the PoliMI lab and the EPFL lab is shown. There is a high degree of correlation (0,986). The process followed to obtain the DMOS consisted on the following steps:

We calculated the difference scores by subtracting the scores of the degraded videos to the score of the reference videos.

The difference scores for the reference videos were set to 0 and were removed. A lower difference score indicates a higher quality.

Each subject may have used the rating scale differently and with different offset. In order to account for this, the Z-scores were computed for each subject separately by means of the Matlab zscore function. The Z-scores transform the original distribution to one in which the mean becomes zero and the standard deviation becomes one. Indeed, this normalization procedure reduces the gain and offset between the subjects.

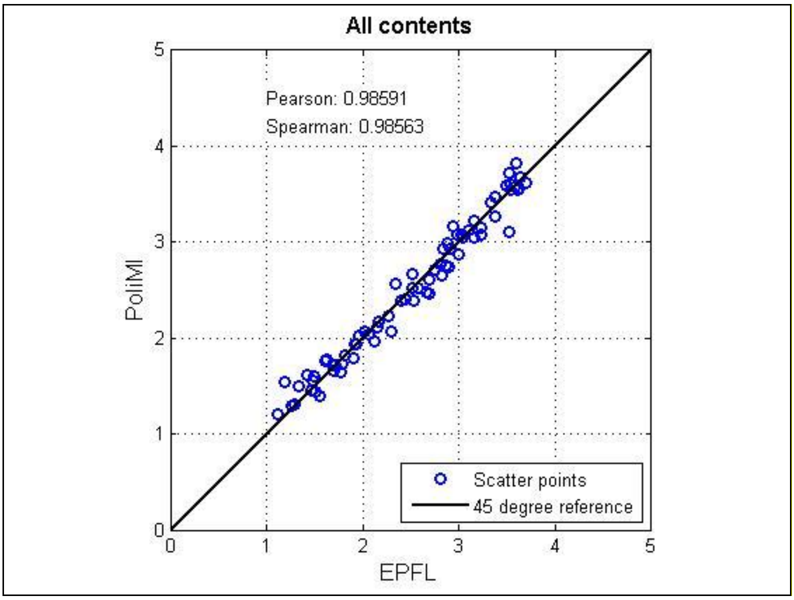


Figure 43. Scatter plot showing the correlation between the DMOS values obtained in the EPFL lab and in the PoliMI lab

The outliers were detected according to the guidelines described in ITU-T Rec 910 Annex 2 Section 2.3.1 [51] and removed.

The Z-scores were re-scaled to the range [0,5]. The Z-scores are assumed to be distributed as a standard Gaussian. Consequently, 99% of the scores will be in the range [−3,3]. In our study, 100% of the scores were placed in that range. All the data was in fact in the range[−3,3] so no clipping was done. The re-scaling was performed by linearly mapping the data range [−3,3] to the range [0,5] using the following formula: $z' = 5 * (z+3) / 6$

The Difference Mean Opinion Score (DMOS) of each video was computed as the mean of the rescaled Z-scores from the 36 subjects that remained after rejection. The confidence intervals were also computed.

Annex B

Resources used by packet loss simulator, decoder and VQM metric with multiple processes

We start by analysing the resource consumption of the packet loss simulator in Table 17. The packet loss percentage introduced by the packet loss simulator was set to 1%. The resource consumption is analysed in the case of running one or multiple processes.

Table 17. Resources packet loss simulator - disk SSD - i7 - memory RAM 32 GB (packet loss percentage 1%)

RESOURCES PACKET LOSS SIMULATOR - DISK SSD - i7 - MEMORY RAM 32 GB (Packet loss percentage 1%)				
NUMBER OF SIMULTANEOUS PROCESSES	**MEMORY USED IN THE COMPUTER (%)**	**CPU USED IN THE COMPUTER (%)**	**DISK TRANSFER USED IN THE COMPUTER (%)**	
1	6	14	1	
2	6	28	2	
10	7	88	7	
20	8	100	10	Saturated CPU
50	12	100	24	
100	17	100	32	
200	21	100	33	
300	30	100	33	
400	39	100	34	
450	43	100	33	
500	46	100	34	

RESOURCES USED JUST BY THE PACKET LOSS SIMULATOR IN ONE PROCESS - CPU USAGE: 1,7% / MEMORY: 18,2 MB /DISK TRANSFER: 1,3 MB/s

The CPU used in the computer as it appears on Table 17 is the percentage of CPU busy in the computer, and therefore it includes the operating system consumption.

However in the bottom part of Table 17 we show the resources used just by the packet loss simulator in one process. In that case the consumption of the operating system is not included in the measurement.

In the machine used (i7 with 8 cores, 32 GB RAM, SSD disk) the bottleneck seems to be on the CPU usage of the packet loss simulator, rather than in the RAM memory or disk transfer capacity.

Now we show the resources used by the decoder in Table 18. The packet loss percentage present in the file to be decoded was 1%. The resource consumption is analysed in the case running one or multiple processes.

Table 18. Resources decoder - disk SSD - i7 - memory RAM 32 GB (packet loss percentage 1%)

RESOURCES DECODER - DISK SSD - i7 - MEMORY RAM 32 GB (Packet loss percentage 1%)				
NUMBER OF SIMULTANEOUS PROCESSES	**MEMORY USED IN THE COMPUTER (%)**	**CPU USED IN THE COMPUTER (%)**	**DISK TRANSFER USED IN THE COMPUTER (%)**	
1	8	30	20	
2	12	58	41	
3	13	82	60	
4	14	100	80	Saturated CPU
5	14	100	98	
6	15	100	99	
7	17	100	100	Saturated disk
8	18	60	100	
9	19	50	100	
10	20	40	100	
20	23	35	100	
50	32	33	100	
100	45	32	100	

RESOURCES USED JUST BY THE DECODER IN ONE PROCESS - CPU USAGE: 28% / MEMORY: 85 MB /DISK TRANSFER: 185 MB/s

The meaning of the parameters displayed in Table 18 is the same as in Table 17. We can see that with four processes the CPU becomes saturated and then from 7 processes onward the bottleneck is on disk transfer capacity.

Finally we show the resources used by the VQM metric computation in Table 19.

Table 19. Resources VQM metric - disk SSD - i7 - memory ram 32 GB (packet loss percentage 1%)

RESOURCES VQM METRIC - DISK SSD - i7 - MEMORY RAM 32 GB (Packet loss percentage 1%)				
NUMBER OF SIMULTANEOUS PROCESSES	**MEMORY USED IN THE COMPUTER (%)**	**CPU USED IN THE COMPUTER (%)**	**DISK TRANSFER USED IN THE COMPUTER (%)**	
1	6	23	2	
2	8	46	3	
5	13	93	9	
10	21	100	8	Saturated CPU
20	33	100		
50	66	100	50	
100	90	100	30	

RESOURCES USED JUST BY THE VQM METRIC IN ONE PROCESS - CPU USAGE: 18,3% / MEMORY: 300 MB /DISK TRANSFER: 0,4 MB/s

The meaning of the parameters displayed in Table 19 is the same as in Table 17.

In the execution of the VQM metric in multiple processes the bottleneck is in the CPU.

If we have a look at the resources used by just one process without considering the operating system (that is not considering the percentage of resource busy in the computer but more specifically looking at the usage by that process) and we compare the packet loss simulator, the decoder and the VQM metric we can conclude that:

- The VQM metric consumes more RAM memory (300 MB) than the decoder (85 MB) and the packet loss simulator (18,2 MB).

- The decoder uses more CPU (28%) than the VQM metric (18,3%) and the packet loss simulator (1,7%).

- Regarding the disk transfer the decoder uses by big difference much more resources (185 MB/s) than the packet loss simulator (1,3 MB/s) and the VQM metric (0,4 MB/s)

If we have a look at the case of multiple processes and compare packet loss simulator, decoder and VQM metric we can conclude that the bottleneck of the packet loss simulator and VQM metric is in the CPU and the bottleneck of the decoder is in the disk transfer capacity.

These conclusions are valid for the configuration proposed: i7, 32 GB of RAM memory and SSD technology disk.

Measurements of time for the packet loss simulator, decoder and VQM metric with multiple processes

In this section we show the study we have done for the following machine: i7, 32 GB RAM MEMORY and disk SSD regarding the time needed for execution of the packet loss simulator, decoder and VQM video quality metric separately.

We begin by showing the times needed for execution of the packet loss simulator in Table 20. In this case the whole study was done three times and averaged for better precision

Table 20. Timings packet loss simulator in seconds - disk SSD - i7 - RAM memory 32 GB (study done 3 times and averaged for better precision)

TIMINGS PACKET LOSS SIMULATOR IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 3 times and averaged for better precision)							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS PROCESSES	Number of experiments	1%	5%	10%	20%	30%	40%
1	1	0,118	0,114	0,110	0,112	0,112	0,104
	10	1,180	1,147	1,103	1,124	1,118	1,044
2	1	0,131	0,128	0,124	0,120	0,116	0,109

	10	1,309	1,283	1,239	1,197	1,157	1,093
10	1	0,272	0,287	0,275	0,266	0,259	0,236
	10	2,719	2,866	2,747	2,661	2,588	2,355
20	1	0,541	0,533	0,541	0,505	0,534	0,495
	10	5,406	5,330	5,409	5,046	5,336	4,945
50	1	1,503	1,319	1,467	1,441	1,275	1,194
	10	15,034	13,187	14,666	14,408	12,745	11,937
100	1	2,959	2,971	3,015	2,859	2,670	2,426
	10	29,591	29,712	30,150	28,588	26,696	24,256
200	1	7,866	7,478	7,490	6,649	5,848	5,127
	10	78,656	74,784	74,897	66,489	58,478	51,271
300	1	10,463	10,429	10,156	9,567	8,152	7,202
	10	104,63	104,29	101,56	95,66	81525	72,02
400	1	16,091	16,017	14,944	13,096	10,948	10,502
	10	160,91	160,16	149,44	130,96	109,48	105,02
450	1	25,302	22,258	18,935	14,854	11,973	11,203
	10	253,02	222,58	189,35	148,53	119,72	112,02
500	1	28,691	24,892	23,287	18,009	13,549	12,700
	10	286,91	248,91	232,87	180,09	135,48	127,00

In Table 21 we show the times per process. We have divided the times shown in Table 20 by the number of simultaneous processes.

Table 21. Timings packet loss simulator per process in seconds - disk SSD - i7 - RAM memory 32 GB (study done 3 times and averaged for better precision)

TIMINGS PACKET LOSS SIMULATOR PER PROCESS IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 3 times and averaged for better precision)							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS PROCESSES	**Number of experiments**	1%	5%	10%	20%	30%	40%
1	1	0,118	0,114	0,110	0,112	0,112	0,104
	10	1,180	1,147	1,103	1,124	1,118	1,044
2	1	0,066	0,064	0,062	0,060	0,058	0,055

	10	0,655	0,642	0,620	0,599	0,579	0,547
10	1	0,027	0,029	0,028	0,027	0,026	0,024
	10	0,272	0,287	0,275	0,266	0,259	0,236
20	1	0,027	0,027	0,027	0,025	0,027	0,025
	10	0,270	0,267	0,270	0,252	0,267	0,247
50	1	0,030	0,026	0,029	0,029	0,026	0,024
	10	0,301	0,264	0,293	0,288	0,255	0,239
100	1	0,030	0,030	0,030	0,029	0,027	0,024
	10	0,296	0,297	0,302	0,286	0,267	0,243
200	1	0,039	0,037	0,037	0,033	0,029	0,026
	10	0,393	0,374	0,374	0,332	0,292	0,256
300	1	0,035	0,035	0,034	0,032	0,027	0,024
	10	0,349	0,348	0,339	0,319	271,75	0,240
400	1	0,040	0,040	0,037	0,033	0,027	0,026
	10	0,402	0,400	0,374	0,327	0,274	0,263
450	1	0,056	0,049	0,042	0,033	0,027	0,025
	10	0,562	0,495	0,421	0,330	0,266	0,249
500	1	0,057	0,050	0,047	0,036	0,027	0,025
	10	0,574	0,498	0,466	0,360	0,271	0,254

The time needed for execution of a single instance of the packet loss simulator is very small. Generally speaking, the time per process increases if we increase the number of processes (at least for packet losses between 1% and 20%).

If a higher percentage of packet loss is introduced in the encoded file the time needed for the execution of the packet loss simulator decreases. This is clearly seen in the case where we have 500 processes (the time changes from 28,691 to 12,700).

According to Table 21 we can extract the conclusion that the optimum is to have around 10 to 20 processes of packet loss simulator simultaneously active, if we look at the 1% packet loss rate column.

The decoder was analysed with the same machine i7, 32 GB RAM memory and disk SSD. We differentiate two different test conditions. In the first one, shown in Table 22, the computer was not left inactive between experiments regarding different number of processes. In the second test condition, the computer was left inactive approximately 5 minutes between experiments regarding different number of processes.

Table 22. Timings decoder in seconds - disk SSD - i7 - RAM memory 32 GB (study done 1 time). Computer left inactive 0 minutes between experiments regarding different number of processes

TIMINGS DECODER IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 1 time)							
Computer left inactive 0 minutes between experiments regarding different number of processes							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS PROCESSES	Number of experiments	1%	5%	10%	20%	30%	40%
1	1	5,375	5,394	5,698	5,819	5,569	5,427
	10	53,750	53,944	56,975	58,194	55,693	54,274
2	1	7,414	8,252	9,308	8,488	9,709	7,805
	10	74,137	82,619	93,077	84,878	97,089	78,047
10	1	48,176	59,236	44,826	46,327	48,521	40,361
	10	481,75	592,36	448,26	463,27	485,21	403,61
20	1	119,02	164,40	144,70	134,74	116,47	94,339
	10	1190,2	1644,0	1447,0	1347,4	1164,7	943,38
50	1	347,52	414,50	355,11	370,68	321,02	287,79
	10	3475,2	4145,0	3551,1	3706,8	3210,2	2877,9
100	1	948,96	870,36	800,53	714,66	671,55	638,45
	10	9489,6	8706,3	8005,3	7146,6	6715,5	6384,5

In Table 23 we show the times per process. We have divided the times shown in Table 22 by the number of simultaneous processes.

Table 23. Timings decoder per process in seconds - disk SSD - i7 - RAM memory 32 GB (study done 1 time). Computer left inactive 0 minutes between experiments regarding different number of processes

TIMINGS DECODER PER PROCESS IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 1 time)

Computer left inactive 0 minutes between experiments regarding different number of processes							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS PROCESSES	Number of experiments	1%	5%	10%	20%	30%	40%
1	1	5,375	5,394	5,698	5,819	5,569	5,427
	10	53,750	53,944	56,975	58,194	55,693	54,274
2	1	3,707	4,126	4,654	4,244	4,855	3,903
	10	37,069	41,310	46,539	42,439	48,545	39,024
10	1	4,818	5,924	4,483	4,633	4,852	4,036
	10	48,176	59,236	44,826	46,327	48,521	40,361
20	1	5,951	8,220	7,235	6,737	5,824	4,717
	10	59,513	82,205	72,351	67,371	58,238	47,169
50	1	6,951	8,290	7,102	7,414	6,420	5,756
	10	69,506	82,901	71,022	74,137	64,204	57,560
100	1	9,490	8,704	8,005	7,147	6,716	6,385
	10	94,897	87,064	80,053	71,466	67,156	63,846

For a high number of processes (more than 10) if the packet loss percentage is increased, the decoding time is reduced. This is probably dependent on the decoder. In our case we used the ffmpeg decoder. The time needed for execution of the decoder is much higher than the time needed for execution of the packet loss simulator. Newly as the number of processes increases, so does the time needed per process (for 1% of packet losses) from 5,375 seconds per process to 9,490 seconds per process for 100 simultaneous processes.

Table 24. Timings decoder in seconds - disk SSD - i7 - RAM memory 32 GB (study done 1 time). Computer left inactive 5 minutes between experiments regarding different number of processes

TIMINGS DECODER IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 1 time)							
Computer left inactive 5 minutes between experiments regarding different number of processes							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS	Number of experiments	1%	5%	10%	20%	30%	40%

PROCESSES							
1	1	4,523	4,759	4,766	4,921	5,217	5,427
	10	45,227	47,592	47,657	49,208	52,169	54,274
2	1	6,363	6,498	6,309	7,4658	6,956	7,805
	10	63,632	64,98	63,091	74,658	69,563	78,047
10	1	36,175	36,24	36,633	36,66	39,747	40,361
	10	361,74	362,39	366,33	366,60	397,46	403,61
20	1	93,276	87,416	90,056	89,895	88,694	94,339
	10	932,75	874,16	900,56	898,95	886,94	943,38
50	1	284,22	279,68	281,32	273,29	275,82	287,79
	10	2842,1	2796,8	2813,2	2732,9	2759,2	2877,9
100	1	752,12	628,55	772,55	688,83	603,22	638,45
	10	7521,2	6285,5	7725,5	6888,3	6032,2	6384,5

In Table 25 we show the times per process. We have divided the times shown in Table 24 by the number of simultaneous processes.

Table 25. Timings decoder per process in seconds - disk SSD - i7 - RAM memory 32 GB (study done 1 time). Computer left inactive 5 minutes between experiments regarding different number of processes

TIMINGS DECODER PER PROCESS IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 1 time)							
Computer left inactive 5 minutes between experiments regarding different number of processes							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS PROCESSES	Number of experiments	1%	5%	10%	20%	30%	40%
1	1	4,523	4,759	4,766	4,921	5,217	5,427
	10	45,227	47,592	47,657	49,208	52,169	54,274
2	1	3,1815	3,249	3,1545	3,7329	3,478	3,9025
	10	31,816	32,49	31,545	37,329	34,781	39,023
10	1	3,6175	3,624	3,6633	3,666	3,9747	4,0361
	10	36,174	36,239	36,633	36,660	39,746	40,361
20	1	4,6638	4,3708	4,5028	4,4947	4,4347	4,7169

	10	46,637	43,708	45,028	44,947	44,347	47,169
50	1	5,6844	5,5937	5,6264	5,4659	5,5165	5,7559
	10	56,843	55,937	56,264	54,659	55,185	57,559
100	1	7,5212	6,2855	7,7255	6,8883	6,0322	6,3845
	10	75,212	62,855	77,255	68,883	60,322	63,845

If we leave inactive the computer 5 minutes between experiments regarding different number of processes the time needed for execution is reduced between 10% and 20%. We think it is due to the handling of resources of the operating system, probably something related to the release of used resources.

Among the values studied, the optimum for the decoder is to have two simultaneous processes.

Finally the VQM metric was analysed with the same machine i7, 32 GB RAM memory and disk SSD.

Table 26. Timings VQM metric in seconds - disk SSD - i7 - RAM memory 32 GB (study done 1 time)

TIMINGS VQM METRIC IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 1 time)							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS PROCESSES	**Number of experiments**	1%	5%	10%	20%	30%	40%
1	1	231,13	234,07	237,00	234,44	233,95	231,39
2	1	285,70	289,17	287,77	289,20	289,64	283,56
5	1	574,62	573,04	570,00	570,84	569,89	567,96
10	1	1122,1	1115,2	1113,6	1112,7	1119,4	1095,6
20	1	2239,1	2242,1	2245,3	2234,5	2232,2	2206,6
50	1	5594,7	5581,9	5578,2	5576,3	5565,0	5482,2
100	1	13057	12703	12951	11759	11423	12542

In Table 27 we show the times per process. We have divided the times shown in Table 26 by the number of simultaneous processes.

Table 27. Timings VQM metric per process in seconds - disk SSD - i7 - RAM memory 32 GB (study done 1 time)

TIMINGS VQM METRIC PER PROCESS IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (study done 1 time)							
		Packet losses percentage					
NUMBER OF SIMULTANEOUS PROCESSES	Number of experiments	1%	5%	10%	20%	30%	40%
1	1	231,13	234,07	237,00	234,44	233,95	231,39
2	1	142,85	144,58	143,88	144,60	144,82	141,78
5	1	114,92	114,60	114,00	114,16	113,97	113,59
10	1	112,21	111,52	111,36	111,27	111,90	109,56
20	1	111,95	112,10	112,26	111,72	111,61	110,33
50	1	111,89	111,63	111,56	111,52	111,30	109,64
100	1	130,57	127,03	129,51	117,59	114,23	125,42

We can see that the optimum is to have 10 simultaneous processes regarding the VQM metric. There is no significant variation of the time needed by VQM if we vary the packet losses (no conclusion can be extracted regarding that variation).

Measurements of time for the packet loss simulator, decoder and VQM metric executed as in the ITU-R BT.1789 for i7, 32 GB RAM, SSD disk

We start by showing in Table 28 the response of the system when a given number of processes are started with a time delay between start of processes for a machine with the following configuration: i7, 32 GB RAM, SSD disk.

Table 28. Timings ITU-R BT1789 in seconds - disk SSD - i7 - RAM memory 32 GB (packet losses percentage 1%)

TIMINGS ITU-R BT1789 IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB(Packet losses percentage 1%)							
	TIME BETWEEN START OF PROCESSES						Average per process
TOTAL NUMBER OF	0"	30"	60"	120"	180"	END-START	

PROCESSES							
1	248,58	249,80	249,74	244,73	239,04	234,85	244,460
3	378,57	399,36	437,02	505,05	607,84	733,42	170,07
5	596,90	615,75	648,60	759,00	965,24	1163,6	158,30
10	1162,6	1160,6	1187,8	1371,9	1866,1	2325,9	151,25
20	2325,0	2283,5	2306,9	2579,7	3674,2	4797,2	149,72
40	4657,0	4528,4	4549,4	4980,1	7274,6	9622,8	148,38
60	7007,1	6799,0	6802,0	7371,5	10875	14485	148,17
80	9840,8	9059,6	9057,2	9781,3	14476	19299	148,99
100	12817	11319	11317	12182	18075	24163	149,79
Average per process	122,36	114,15	114,59	124,68	181,99	240,83	

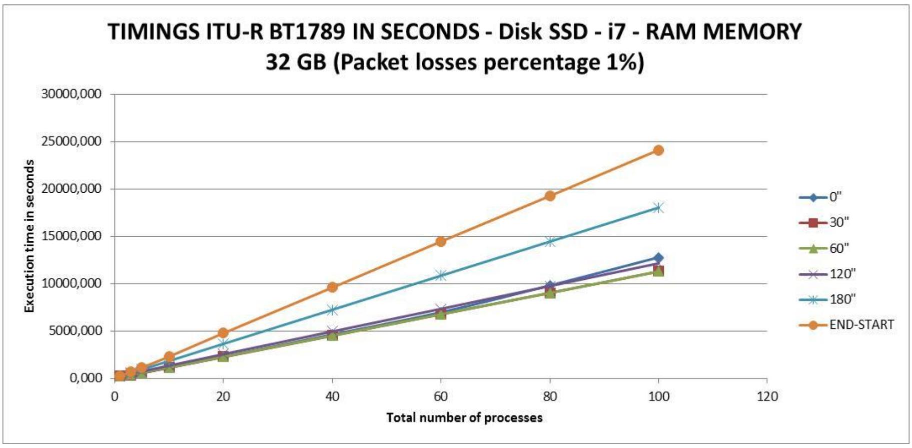


Figure 44. Timings ITU-R BT1789 in seconds - disk SSD - i7 - RAM memory 32 GB (packet losses percentage 1%)

In Table 29 we divide the time of Table 28 by the total number of processes in order to have a measure of time per process.

Table 29. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 - RAM memory 32 GB (packet losses percentage 1%)

TIMINGS ITU-R BT1789 PER PROCESS IN SECONDS - Disk SSD - i7 - RAM MEMORY 32 GB (Packet losses percentage 1%)							
	TIME BETWEEN START OF PROCESSES						Average per process
TOTAL	0"	30"	60"	120"	180"	END-	

NUMBER OF PROCESSES						START	
1	248,58	249,80	249,74	244,73	239,04	234,85	244,46
3	126,19	133,12	145,67	168,35	202,61	244,47	170,07
5	119,38	123,15	129,72	151,80	193,04	232,73	158,30
10	116,26	116,06	118,78	137,19	186,61	232,59	151,25
20	116,25	114,17	115,34	128,98	183,71	239,86	149,72
40	116,42	113,21	113,73	124,50	181,86	240,57	148,38
60	116,78	113,31	113,36	122,86	181,26	241,42	148,17
80	123,01	113,24	113,21	122,26	180,95	241,24	148,99
100	128,17	113,19	113,17	121,82	180,76	241,63	149,79
Average per process	134,56	132,14	134,75	146,94	192,20	238,82	

We can see clearly in the graph that the optimum is to start the processes with a delay of 60 seconds. In Table 29 we see also that the optimum case, highlighted in blue, happens for a number of processes of 100. However with 10 processes only the times are not very far from the times with 100 processes (only 5 seconds per process of difference).

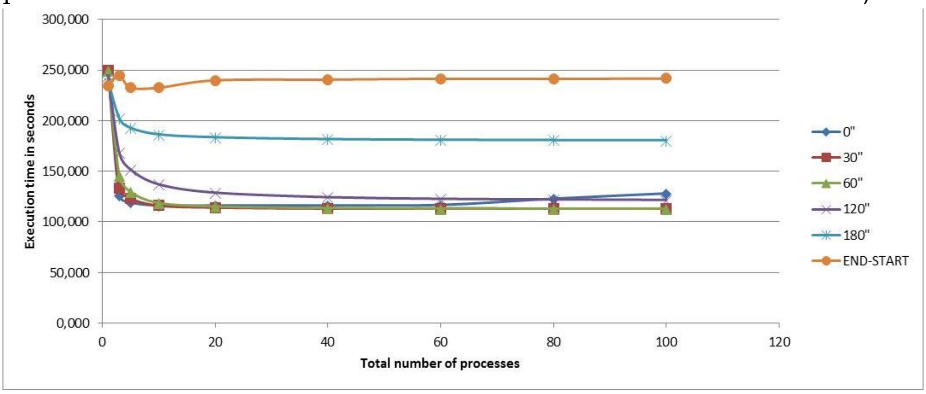


Figure 45. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 - RAM memory 32 GB (packet losses percentage 1%)

Using the tool developed the best case (100 processes, 60 seconds time between start of processes) is shown in Figure 45 and Figure 46.

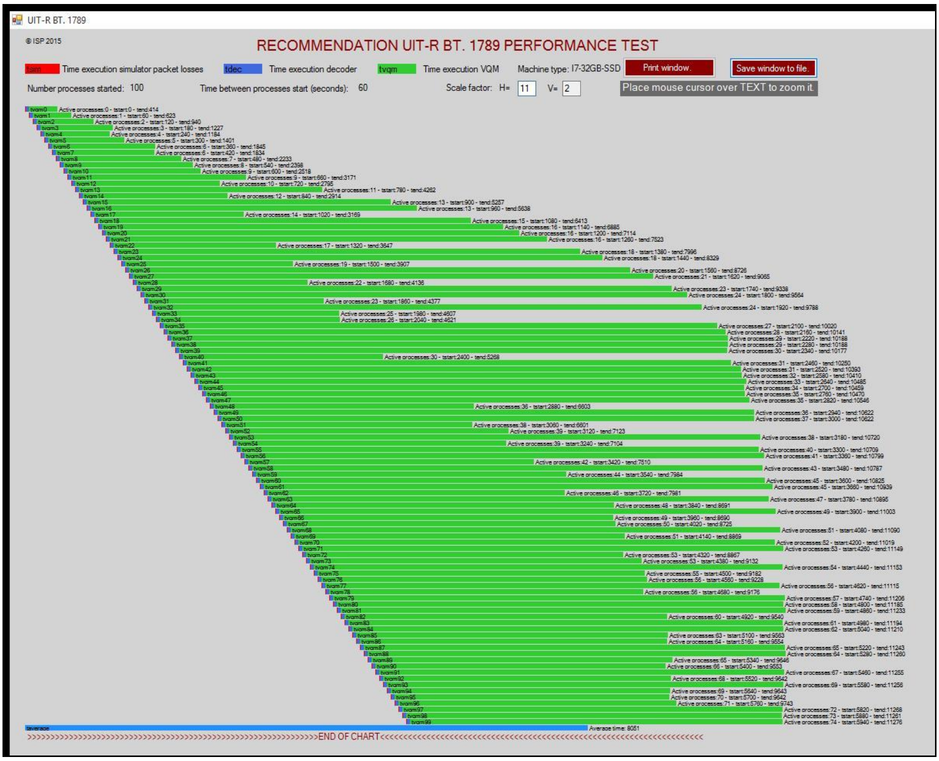


Figure 46. Best case ITU-R BT1789 - disk SSD - i7 - RAM memory 32 GB (packet losses percentage 1%)

As we can see the number of simultaneous processes tends to grow, indicating that if we run the same simulation with many more processes it could become unmanageable in terms of memory, CPU and time needed for execution.

Measurements of time for the packet loss simulator, decoder and VQM metric executed as in the ITU-R BT.1789 for i5, 32 GB RAM, SSD disk

In Table 30 we show the same analysis than in the previous section but for a different machine: i5, 32 GB RAM and disk SSD.

Table 30. Timings ITU-R BT1789 in seconds - disk SSD - i5 - RAM memory 32 GB (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES					
TOTAL NUMBER OF PROCESSES	0"	30"	60"	120"	180"	END-START
1	248,191	247,200	248,053	247,076	247,031	268,830
3	388,846	418,757	452,122	527,338	624,150	740,934
5	592,217	618,775	663,230	783,558	984,747	1235,22
10	1159,79	1181,30	1206,55	1394,87	1884,35	2470,07
20	2362,17	2315,43	2339,89	2605,40	3688,79	5001,52
40	4777,63	4614,60	4623,78	5007,29	7290,24	10039,6
60	8483,08	6923,19	6917,69	7405,79	10888,7	15033,8
80	11907,6	9247,23	9224,25	9805,69	14489,1	20032,9
100	14630,1	11564,7	11570,8	12206,4	18089,6	25049,1
Average per process	139,654	116,399	116,760	125,340	182,404	250,383

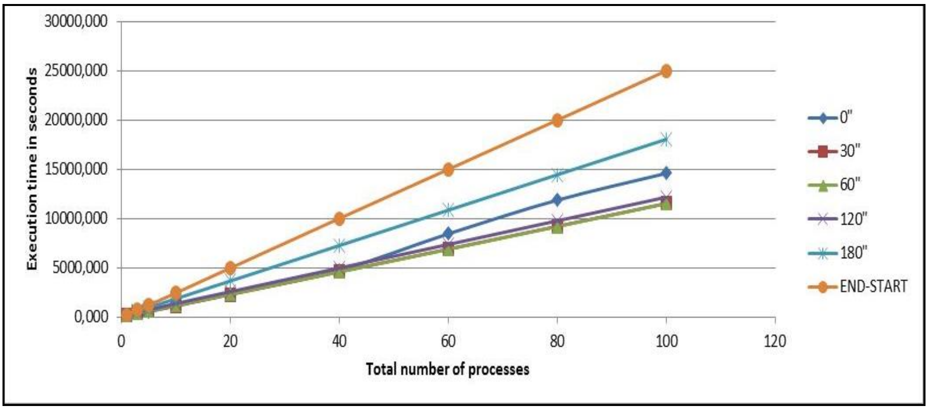


Figure 47. Timings ITU-R BT1789 in seconds - disk SSD - i5 - RAM memory 32 GB (packet losses percentage 1%)

Table 31. Timings ITU-R BT1789 per process in seconds - disk SSD - i5 - RAM memory 32 GB (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES					
TOTAL NUMBER OF PROCESSE	0"	30"	60"	120"	180"	END-START

s						
1	248,191	247,200	248,053	247,076	247,031	268,830
3	129,615	139,586	150,707	175,779	208,050	246,978
5	118,443	123,755	132,646	156,712	196,949	247,046
10	115,979	118,131	120,655	139,487	188,435	247,007
20	118,109	115,772	116,995	130,270	184,440	250,076
40	119,441	115,365	115,595	125,182	182,256	250,990
60	141,385	115,387	115,295	123,430	181,479	250,564
80	148,845	115,590	115,303	122,571	181,114	250,412
100	146,301	115,647	115,709	122,064	180,896	250,492
Average per process	142,923	134,048	136,773	149,175	194,517	251,377

Here the best case happens also for a delay between start of processes of 60 seconds. Regarding the number of processes the optimum is 60, but the there is no significant difference with 20 processes or even with the case of 10 processes.

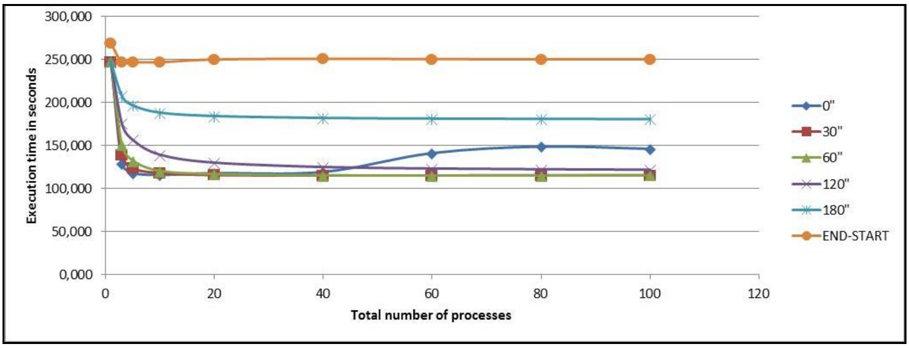


Figure 48. Timings ITU-R BT1789 per process in seconds - disk SSD - i5 - RAM memory 32 GB (packet losses percentage 1%)

Figure 48 shows that only for the case of 0”, that means running all the processes without delay in the start of them, the time per process increases with the total number of processes, especially from 40 to 80 processes.

In the following graph we show the process execution for the best case. The best case occurs with number of processes started equal to 60 and time between processes start equal to 60 seconds.

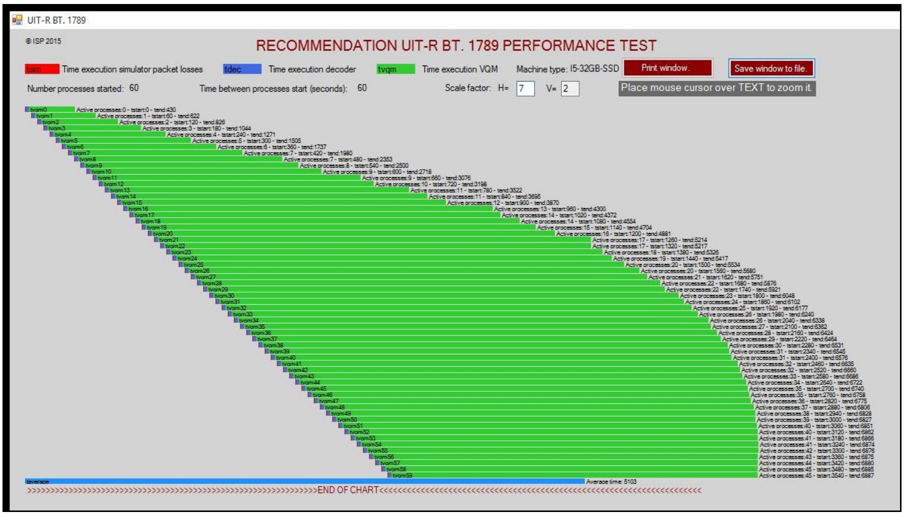


Figure 49. Best case ITU-R BT1789 - disk SSD – i5 - RAM memory 32 GB (packet losses percentage 1%)

In Figure 49 we can see more stable times regarding the time needed for execution of processes than in the previous section. We can see that the number of simultaneous processes being executed tends to grow.

Comparing the results of the machine with i5 and the machine with i7, we can see that they show rather similar performance (slightly better the machine with i7 but only 2 seconds per process).

Measurements of time for the packet loss simulator, decoder and VQM metric executed as in the ITU-R BT.1789 for i7, 32 GB RAM, HD disk

Here we show the same analysis but for a machine with no SSD disk. The configuration is: HD disk (no SSD disk), i7, 32 GB RAM

Table 32. Timings ITU-R BT1789 in seconds - disk HD - i7 - RAM memory 32 GB (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES					
TOTAL NUMBER OF PROCESSES	0"	30"	60"	120"	180"	END-START

1	247,732	246,194	246,220	246,616	246,827	248,187
3	440,903	417,325	449,443	519,346	610,890	741,200
5	723,698	625,637	660,186	765,054	969,080	1189,11
10	1444,58	1172,94	1199,65	1372,71	1869,68	2386,09
20	4569,32	4327,00	3139,38	2596,32	3685,96	5054,86
40	9577,69	8574,41	7811,97	4992,78	7278,95	9969,99
60	15383,8	15407,9	13170,4	7394,22	10881,0	14989,0
80	29865,0	21130,3	19344,4	9785,82	14481,9	19890,4
100	39428,1	30122,6	27538,6	12193,8	18081,3	24884,5
Average per process	318,749	257,130	230,597	124,974	182,150	248,757

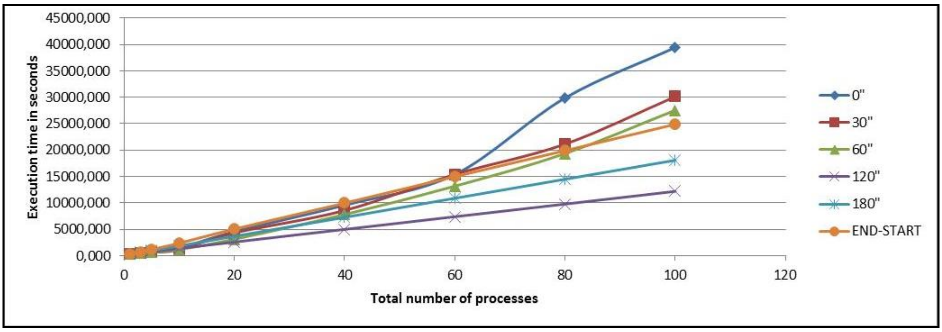


Figure 50. Timings ITU-R BT1789 in seconds - disk HD - i7 - RAM memory 32 GB (packet losses percentage 1%)

Table 33. Timings ITU-R BT1789 per process in seconds - disk HD - i7 - RAM memory 32 GB (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES					
TOTAL NUMBER OF PROCESSES	0"	30"	60"	120"	180"	END-START
1	247,732	246,194	246,220	246,616	246,827	248,187
3	146,968	139,108	149,814	173,115	203,630	247,067
5	144,740	125,127	132,037	153,011	193,816	237,824
10	144,458	117,294	119,966	137,271	186,968	238,610
20	228,466	216,350	156,969	129,816	184,298	252,743

40	239,442	214,360	195,299	124,820	181,974	249,250
60	256,398	256,800	219,508	123,237	181,350	249,817
80	373,313	264,130	241,806	122,323	181,025	248,631
100	394,282	301,226	275,387	121,938	180,814	248,845
Average per process	241,755	208,954	193,001	148,016	193,411	246,775

The optimum case happens for a delay of 30 seconds between start of processes and a total number of processes of 10.

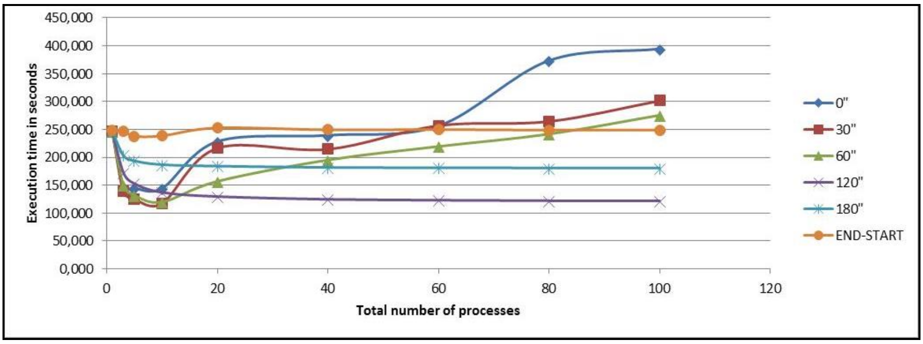


Figure 51. Timings ITU-R BT1789 per process in seconds - disk HD - i7 - RAM memory 32 GB (packet losses percentage 1%)

We can clearly see in the Figure 51 that for the intervals of 60 seconds, 30 seconds and 0 seconds, the time needed for execution per process increases with the total number of processes, meaning that the efficiency is reduced.

For a high number of processes the best is to have an interval of 120 seconds between start of processes as we can see in the graph that the efficiency is maintained even with a high number of processes.

In Figure 52 we show the best case (10 processes, interval 30 seconds).

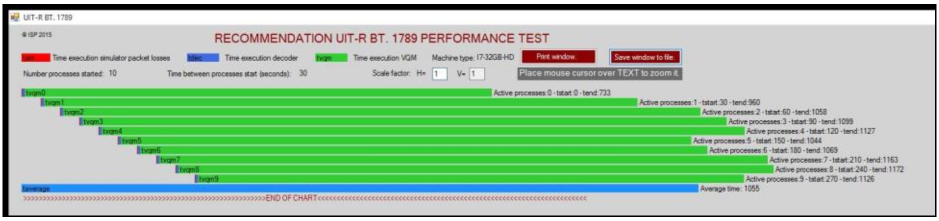


Figure 52. Best case ITU-R BT1789 - disk HD – i7 - RAM memory 32 GB (packet losses percentage 1%)

Comparing this case with the case of having an SSD disk, we can appreciate that with SSD the performance is better than with HD disk. The time per process is reduced in approximately 4 seconds.

Measurements of time for the packet loss simulator, decoder and VQM metric executed as in the ITU-R BT.1789 for i7, 16 GB RAM, SSD disk

Finally we make the analysis for a machine with less RAM memory. The configuration of the machine is the following: i7, 16 GB RAM, SSD disk.

Table 34. Timings ITU-R BT1789 in seconds - disk SSD - i7 - RAM memory 16 GB (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES					
TOTAL NUMBER OF PROCESSES	0"	30"	60"	120"	180"	END-START
1	235,533	236,054	236,138	235,518	236,207	237,480
3	378,051	395,162	432,784	505,856	606,525	696,754
5	577,390	600,517	641,679	762,688	969,587	1189,52
10	1141,31	1142,22	1170,02	1374,49	1874,16	2434,11
20	2278,70	2229,94	2256,39	2575,98	3673,61	4811,79
40	5397,74	4411,31	4431,36	4975,60	7274,07	9613,78
60	10685,3	6611,98	6618,22	7376,38	10873,7	14446,5
80	24355,8	9447,04	8833,57	9777,66	14474,3	19276,0
100	System crack	23151,6	11260,0	12178,3	18074,9	24109,3
Average per process	205,707	151,178	112,477	124,647	181,997	240,801

The first thing to be noticed is that when having 100 processes a system crack is produced if the processes are executed with a delay of 0 seconds between them. This means in fact that the simulation cannot be executed in that case. We get to the conclusion that the memory is the most critical component in order to execute the ITU-T recommendation that is being analysed (at least with the VQM metric).

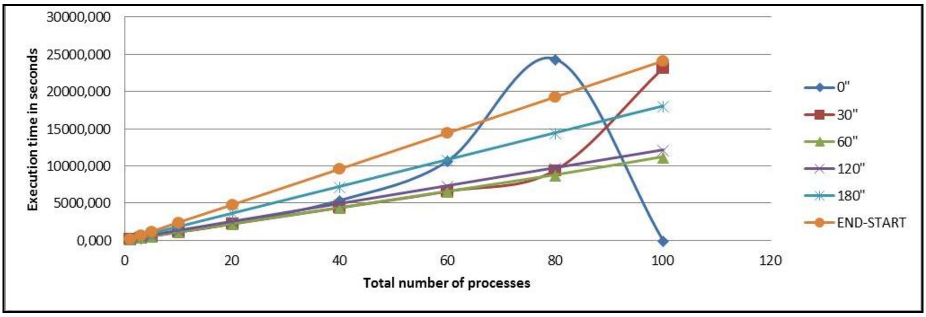


Figure 53. Timings ITU-R BT1789 in seconds - disk SSD - i7 - RAM memory 16 GB (packet losses percentage 1%)

Table 35. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 – RAM memory 16 GB (packet losses percentage 1%)

	TIME BETWEEN START OF PROCESSES					
TOTAL NUMBER OF PROCESSES	0"	30"	60"	120"	180"	END-START
1	235,533	236,054	236,138	235,518	236,207	237,480
3	126,017	131,721	144,261	168,619	202,175	232,251
5	115,478	120,103	128,336	152,538	193,917	237,905
10	114,132	114,222	117,002	137,449	187,417	243,412
20	113,935	111,497	112,820	128,799	183,681	240,590
40	134,944	110,283	110,784	124,390	181,852	240,345
60	178,089	110,200	110,304	122,940	181,229	240,777
80	304,448	118,088	110,420	122,221	180,929	240,950
100	System crack	231,517	112,600	121,784	180,749	241,093
Average per process	165,322	142,632	131,407	146,029	192,017	239,423

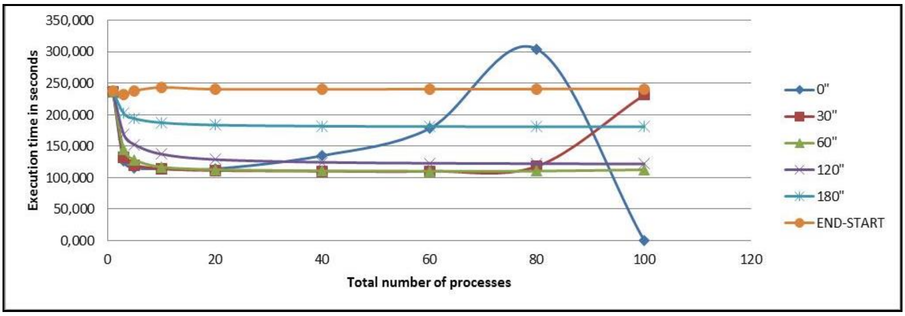


Figure 54. Timings ITU-R BT1789 per process in seconds - disk SSD - i7 - RAM memory 16 GB (packet losses percentage 1%)

As we can see in Table 35 in the case of 0 seconds delay, the memory is probably not enough, leading to an increase of the times needed for execution, and in one of the cases, to a system crack. In the case of 30 seconds between start of processes and 100 processes the time also increases significantly.

The best case is highlighted in blue in Table 35 and happens for a time between start of the processes of 30 seconds and a total of 60 processes. We show it in a detailed view on the following graph.

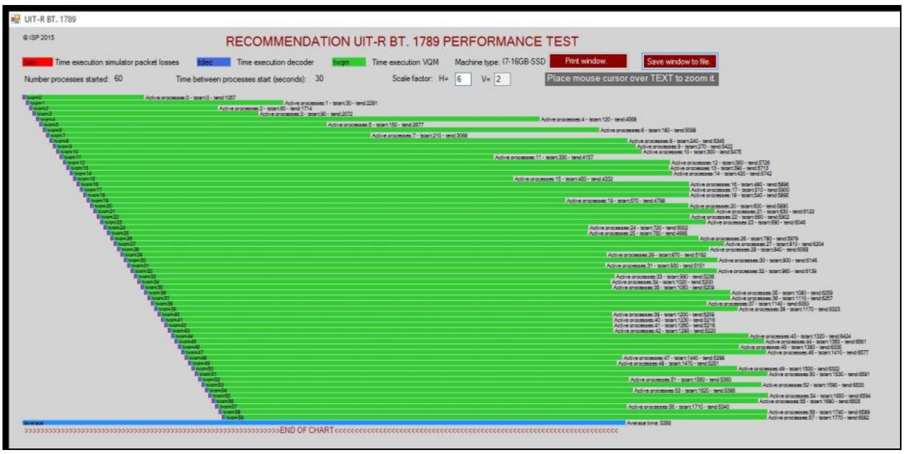


Figure 55. Best case ITU-R BT1789 - disk SSD – i7 - RAM memory 16 GB (packet losses percentage 1%)

The number of simultaneous processes increases gradually as in the previous analyzed best cases (in previous sections).

References

[1] S. Winkler and P. Mohandas, "The evolution of video quality measurement: From PSNR to hybrid metrics," *IEEE Trans. Broadcast.*, vol. 54, no. 3, pp. 660–668, 2008, doi: 10.1109/TBC.2008.2000733.

[2] S. Winkler, *Digital Video Quality: Vision Models and Metrics*. 2013.

[3] S. Akramullah and S. Akramullah, "Video Quality Metrics," in *Digital Video Concepts, Methods, and Metrics*, 2014.

[4] A. R. Reibman, V. A. Vaishampayan, and Y. Sermadevi, "Quality Monitoring of Video Over a Packet Network," *IEEE Trans. Multimed.*, 2004, doi: 10.1109/TMM.2003.822785.

[5] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE Trans. Broadcast.*, 2011, doi: 10.1109/TBC.2011.2104671.

[6] Y. Wang, "Survey of objective video quality measurements," *EMC Corp. Hopkinton, MA*, 2006.

[7] "RISE Acreo Research Institute." [Online]. Available: https://www.acreo.se/. [Accessed: 12-Feb-2020].

[8] "Fundación Centros Tecnológicos - Iñaki Goenaga." [Online]. Available: http://www.euskomedia.org/aunamendi/153723. [Accessed: 12-Feb-2020].

[9] "Lund University." [Online]. Available: http://www.lunduniversity.lu.se/. [Accessed: 12-Feb-2020].

[10] "University of the Basque Country." [Online]. Available: http://www.ehu.eus/en/en-home. [Accessed: 12-Feb-2020].

[11] "Video Quality Experts Group." [Online]. Available: http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx. [Accessed: 13-May-2020].

[12] "VQEG JEG Wiki." [Online]. Available: http://vqegjeg.intec.ugent.be/wiki/index.php/Main_Page. [Accessed: 13-May-2020].

[13] "University of Nantes." [Online]. Available: https://www.univ-nantes.fr. [Accessed: 13-May-2020].

[14] N. Staelens, I. Sedano, M. Barkowsky, L. Janowski, K. Brunnstrom, and P. Le Callet, "Standardized toolchain and model development for video quality assessment # x2014; The mission of the Joint Effort Group in VQEG," in *Quality of Multimedia Experience (QoMEX), 2011 Third International Workshop on*, 2011, pp. 61–66, doi: 10.1109/QoMEX.2011.6065713.

[15] M. Barkowsky *et al.*, "Subjective experiment dataset for joint development of hybrid video quality measurement algorithms," in *Proceedings of Third Workshop on Quality of Experience for Multimedia Content Sharing (QoEMCS 2012)*, 2012, pp. 1–4.

[16] "L'Institut de recherche en communications et cybernétique de Nantes." [Online]. Available: http://www.irccyn.ec-nantes.fr/. [Accessed: 13-May-2020].

[17] G. Y. G. Yu *et al.*, "Analysis and characterization of IPTV user behavior," *2009 IEEE Int. Symp. Broadband Multimed. Syst. Broadcast.*, pp. 1–6, 2009, doi: 10.1109/ISBMSB.2009.5133805.

[18] "Tecnalia Research & Innovation." [Online]. Available: https://www.tecnalia.com/en/. [Accessed: 13-May-2020].

[19] VQEG, "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Video Quality Assessment, Phase II," 2003.

[20] VQEG, "Final Report from the Video Quality Experts Group on the Validation of Objective Models of Multimedia Quality Assessment, Phase I," 2008.

[21] VQEG, "Report on the Validation of Video Quality Models for High Definition Video Content," 2010.

[22] "Recommendation ITU-T J.144 Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference," 2001.

[23] "Recommendation ITU-T J.246 Perceptual visual quality measurement techniques for multimedia services over digital cable television networks in the presence of a reduced bandwidth reference," 2008.

[24] "Recommendation ITU-T J.247 Objective perceptual multimedia video quality measurement in the presence of a full reference," 2008.

[25] "Recommendation ITU-R BT.500-14 Methodology for the subjective assessment of the quality of television pictures," 2019.

[26] "Recommendation ITU-T P 910 Subjective video quality assessment methods for multimedia applications," 1999.

[27] N. Staelens *et al.*, "Assessing quality of experience of IPTV and video on demand services in real-life environments," *IEEE Trans. Broadcast.*, vol. 56, no. 4, pp. 458–466, 2010, doi: 10.1109/TBC.2010.2067710.

[28] M. Larabi, A. Pinheiro, J. You, and A. Zgank, "Factors Influencing Quality of Experience," in *Quality of Experience*, 2014.

[29] H. J. Kim and S. G. Choi, "A study on a QoS/QoE correlation model for QoE evaluation on IPTV service," in *International Conference on Advanced Communication Technology, ICACT*, 2010.

[30] P. Pérez, J. Macías, J. J. Ruiz, and N. García, "Effect of packet loss in video quality of experience," *Bell Labs Tech. J.*, 2011, doi: 10.1002/bltj.20488.

[31] M. Claypool and J. Tanner, "The effects of jitter on the peceptual quality of video," 1999, doi: 10.1145/319878.319909.

[32] "IPTV Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/IPTV. [Accessed: 13-May-2020].

[33] K.-C. Yang, "Perceptual quality assessment for compressed video," 2007.

[34] "Video Quality Metric (VQM) software." [Online]. Available: http://www.its.bldrdoc.gov/vqm/. [Accessed: 01-Jan-2016].

[35] "Recommendation ITU-T P.1401 Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models," 2020.

[36] M. Barkowsky, I. Sedano, K. Brunnström, M. Leszczuk, and N. Staelens, "Hybrid video quality prediction: reviewing

video quality measurement for widening application scope," *Multimed. Tools Appl.*, vol. 74, no. 2, pp. 323–343, 2015, doi: 10.1007/s11042-014-1978-2.

[37] "Recommendation ITU-T P.913 Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment," 2016.

[38] "ITU-R BT.814 Specifications of PLUGE test signals and alignment procedures for setting of brightness and contrast of displays," 2018.

[39] "ITU-R BT.815 Specification of a signal for measurement of the contrast ratio of displays," 1994.

[40] "Software AcrVQWin." [Online]. Available: https://www.acreo.se/. [Accessed: 13-May-2020].

[41] "Recommendation P.1203 Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," 2017.

[42] "Recommendation G.1070 Opinion model for video-telephony applications," 2018.

[43] D. H. Loeda, "Desarrollo de un sistema de medición, monitorización y gestión de redes IPTV," 2013.

[44] P. Le Callet, C. Viard-Gaudin, and D. Barba, "A convolutional neural network approach for objective video quality assessment," *IEEE Trans. Neural Networks*, vol. 17, no. 5, pp. 1316–1327, 2006, doi: 10.1109/TNN.2006.879766.

[45] P. Goudarzi, "A no-reference low-complexity QoE measurement algorithm for H.264 video transmission systems," *Sci. Iran.*, vol. 20, no. 3, pp. 721–729, 2013, doi: 10.1016/j.scient.2013.02.021.

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[47] M. Shahid, A. Rossholm, and B. Lovstrom, "A reduced complexity no-reference artificial neural network based

video quality predictor," in *2011 4th International Congress on Image and Signal Processing*, 2011, vol. 1, no. August 2015, pp. 517–521, doi: 10.1109/CISP.2011.6099931.

[48] K. Zhu, K. Hirakawa, V. Asari, and D. Saupe, "A no-reference video quality assessment based on Laplacian pyramids," in *2013 IEEE International Conference on Image Processing, ICIP 2013 - Proceedings*, 2013, pp. 49–53, doi: 10.1109/ICIP.2013.6738011.

[49] K. Seshadrinathan and A. C. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 335–350, 2010, doi: 10.1109/TIP.2009.2034992.

[50] T. Maki, D. Kukolj, D. Dordevic, and M. Varela, "A reduced-reference parametric model for audiovisual quality of IPTV services," in *2013 5th International Workshop on Quality of Multimedia Experience, QoMEX 2013 - Proceedings*, 2013, pp. 6–11, doi: 10.1109/QoMEX.2013.6603193.

[51] "Recommendation ITU-T P.910 Subjective video quality assessment methods for multimedia applications," 1999.

[52] Y. Xia, Z. Wang, W. Wang, J. Wang, Z. Wan, and W. Lin, "Blind measurement of blocking artifacts of images based on edge and flat-region detection," *J. Softw.*, vol. 8, no. 1, pp. 168–175, 2013, doi: 10.4304/jsw.8.1.168-175.

[53] Grupo Técnico del Foro de la Televisión de Alta Definición en España, "Calidad de video en alta definición," 2008.

[54] M. T. Vega, E. Giordano, D. C. Mocanu, D. Tjondronegoro, and A. Liotta, "Cognitive no-reference video quality assessment for mobile streaming services," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, 2015, pp. 1–6, doi: 10.1109/QoMEX.2015.7148128.

[55] L. Lu, Z. Wang, A. C. Bovik, and J. Kouloheris, "Full-reference video quality assessment considering structural distortion and no-reference quality evaluation of MPEG video," in *Proceedings - 2002 IEEE International Conference on Multimedia and Expo, ICME 2002*, 2002, vol. 1, pp. 61–64, doi: 10.1109/ICME.2002.1035718.

[56] X. Gao, W. Lu, D. Tao, and X. Li, "Image Quality Assessment and Human Visual System," in *Proc. of SPIE Vol.*

7744, 2010, pp. 77440Z-77440Z–10, doi: 10.1117/12.862431.

[57] C. E. B. Hormigos, "Medidas de calidad subjetiva en secuencias de video," 2009.

[58] "EPFL-PoliMI video quality assessment database," 2009. [Online]. Available: http://vqa.como.polimi.it.

[59] "Laboratory for Image & Video Engineering," 2015. [Online]. Available: http://live.ece.utexas.edu/research/Quality/index.htm. [Accessed: 01-Jan-2016].

[60] 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," *2009 Int. Work. Qual. Multimed. Exp.*, pp. 204–209, 2009, doi: 10.1109/QOMEX.2009.5246952.

[61] F. De Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and T. Ebrahimi, "A H.264/AVC video database for the evaluation of quality metrics," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 2430–2433, doi: 10.1109/ICASSP.2010.5496296.

[62] "FOURCC, YUV formats." [Online]. Available: http://www.fourcc.org/yuv.php. [Accessed: 13-May-2020].

[63] M. Barkowsky, M. H. Pinson, R. Pépion, and P. Le Callet, "Analysis of Freely Available Subjective Dataset for HDTV including Coding and Transmission Distortions," *Int. Work. Video Process. Qual. Metrics Consum. Electron.*, 2010.

[64] "MATLAB Central File Exchange." [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/. [Accessed: 13-May-2020].

[65] "The Structural SIMilarity (SSIM) index author's home page." [Online]. Available: http://www.ece.uwaterloo.ca/~z70wang/research/ssim/. [Accessed: 01-Jan-2016].

[66] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," *IEEE Asilomar Conf. Signals, Syst. Comput.*, vol. 2, pp. 9–13, 2003, doi: 10.1109/ACSSC.2003.1292216.

[67] D. M. Chandler and S. S. Hemami, “VSNR: A wavelet-based visual signal-to-noise ratio for natural images,” *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2284–2298, 2007, doi: 10.1109/TIP.2007.901820.

[68] “VSNR implementation from the authors.” [Online]. Available: http://foulard.ece.cornell.edu/dmc27/vsnr/vsnr.html. [Accessed: 01-Jan-2016].

[69] VQEG, “Final Report from the VQEG on the validation of Objective Models of Video Quality Assessment, Pase II,” *online] Disponible en: http://www. vpeg. org*, 2003. .

[70] “x264 software.” [Online]. Available: http://www.videolan.org/developers/x264.html. [Accessed: 13-May-2020].

[71] Y.-K. W. Y. Guo, H. Li, “SVC/AVC loss simulator.” [Online]. Available: http://wftp3.itu.int/av-arch/jvt-site/2005_10_Nice/. [Accessed: 01-Jan-2017].

[72] NTIA, “NTIA Technical Memorandum TM-10-463, A Full Reference (FR) Method Using Causality Processing For Estimating Variable Video Delays,” 2009.

[73] “Recommendation ITU-R BT.1789 A method to reconstruct received video using transmission error information for packet video transmission,” 2007.

[74] I. Sedano, K. Brunnström, M. Kihl, and A. Aurelius, “Full-reference video quality metric assisted the development of no-reference bitstream video quality metrics for real-time network monitoring,” *EURASIP J. Image Video Process.*, vol. 2014, no. 1, p. 4, 2014, doi: 10.1186/1687-5281-2014-4.

[75] “Consumer Digital Video Library.” [Online]. Available: http://www.cdvl.org. [Accessed: 13-May-2020].

[76] I. Sedano, M. Kihl, K. Brunnström, A. Aurelius, and I. Technology, “Reconstruction of incomplete decoded videos for use in objective quality metrics,” in *Systems, Signals and Image Processing*, 2012, pp. 376–379.