No-Reference Bitstream-based Visual Quality Impairment Detection for High Definition H.264/AVC Encoded Video Sequences

Nicolas Staelens, *Associate Member, IEEE,* Glenn Van Wallendael, Karel Crombecq, Nick Vercammen, Jan De Cock, *Member, IEEE,* Brecht Vermeulen, Rik Van de Walle, *Member, IEEE,* Tom Dhaene, *Senior Member, IEEE,* and Piet Demeester, *Fellow, IEEE*

***Abstract*—Ensuring and maintaining adequate Quality of Experience towards end-users are key objectives for video service providers, not only for increasing customer satisfaction but also as service differentiator. However, in the case of High Definition video streaming over IP-based networks, network impairments such as packet loss can severely degrade the perceived visual quality. Several standard organizations have established a minimum set of performance objectives which should be achieved for obtaining satisfactory quality. Therefore, video service providers should continuously monitor the network and the quality of the received video streams in order to detect visual degradations. Objective video quality metrics enable automatic measurement of perceived quality. Unfortunately, the most reliable metrics require access to both the original and the received video streams which makes them inappropriate for real-time monitoring. In this article, we present a novel no-reference bitstream-based visual quality impairment detector which enables real-time detection of visual degradations caused by network impairments. By only incorporating information extracted from the encoded bitstream, network impairments are classified as visible or invisible to the end-user. Our results show that impairment visibility can be classified with a high accuracy which enables real-time validation of the existing performance objectives.**

***Index Terms*—Quality of Experience (QoE), Objective video quality, No-Reference, H.264/AVC, High Definition.**

I. Introduction

QUALITY of Experience (QoE), defined as the overall acceptability of an application or service as perceived subjectively by the end-user [1], is considered a key parameter for determining the success or failure of broadband video services such as Internet Protocol Television (IPTV) and Video on Demand (VoD) [2]. When switching to new digital video services, customers expect superior quality compared to the traditional TV services (such as analogue TV), for which they are willing to pay [3], [4], [5]. However, delivering high quality video services over existing IP-based networks can be a real challenge for video service providers, certainly when taking into account the packet-based, best-effort characteristics of such networks. In the event of network impairments, the perceived audiovisual quality can fluctuate significantly and drop below the acceptability thresholds [6], [7], [8]. To some extent, consumers tolerate minor impairments if the (discount) price for the provided service is acceptable and the error frequency remains low enough. But on the other hand, very few consumers accept more than one visual artifact per hour [5]. Overall, service providers should strive for optimizing and maximizing the QoE of the offered video services, not only as a service differentiator but also as means for increasing customer satisfaction and revenues [9], [10].

Ensuring adequate QoE requires that the quality of the streamed video sequence is continuously monitored. On the network level, objective Quality of Service (QoS) parameters such as packet loss, bandwidth, delay and jitter can be measured in order to detect possible network failures which could lead to a degradation of the received video signal [11], [12], [13], [14].

Several standard organizations have established a minimum set of objective QoE performance measurements which should be achieved in order to maintain satisfactory QoE. Recommendations such as ITU-T Recommendation G.1080 [15] or Technical Report TR-126 of the DSL Forum [2], specify for example the maximum allowable number of error events, defined as the loss of a small number of IP packets, per time unit in order to ensure adequate QoE. In the case of High Definition (HD) video streaming, a maximum of one error event per 4 hours of video playback is tolerable.

From the end-user point of view, results have shown that frame freezes are perceived differently from visual impairments caused by random packet loss [16] and that visual impairment visibility also depends both on the type of video content and the location of the loss [17]. As such, not all network impairments result necessarily in a visible degradation, which has also been accounted for in ITU-T G.1080 and TR-126. This indicates that additional information from the video level is required in order to determine and predict the influence of network impairments on perceived quality.

Measuring perceived video quality can be automated using objective video quality metrics. Currently, the most reliable quality metrics either require access to the complete original video sequence or require a complete decoding of the received video signal [18], [19], [20]. Furthermore, these metrics usu-

N. Staelens, N. Vercammen, B. Vermeulen, T. Dhaene and P. Demeester are with Ghent University - IBBT, Department of Information Technology, Ghent, Belgium (e-mail: {nicolas.staelens, nick.vercammen, brecht.vermeulen, tom.dhaene, piet.demeester}@intec.ugent.be).

G. Van Wallendael, J. De Cock and R. Van de Walle are with Ghent University - IBBT, Department of Electronics and Information Systems, Ghent, Belgium (e-mail: {glenn.vanwallendael, jan.decock, rik.vandewalle}@ugent.be).

K. Crombecq is with the Department of Mathematics and Computer Science, University of Antwerp, Antwerp, Belgium (e-mail: karel.crombecq@ua.ac.be).

ally process the entire video sequence and output an overall average quality rating. In the case of real-time video streaming, service providers want to receive instantaneous feedback when visual quality degradations occur.

In this article, we present a novel no-reference bitstream-based objective video quality metric for real-time detection of visual artifacts resulting from network impairments. Our approach focuses on modeling the visibility of network impairments to the average end-user based on a decision tree classifier. As we are targeting a no-reference bitstream-based metric, only parameters extracted solely from the received encoded video bitstream are taken into account.

The remainder of this article is structured as follows. We start by providing a high level overview on the different types of metrics and how they can be classified based on the information they use from the video stream in section II. Then, in section III, we present related work and highlight the importance of the research presented in this article. Our entire test methodology, including a description of the different sequences, encoding parameters and impairment scenarios is detailed in section IV. In section V, we motivate the use of decision trees for modeling packet loss visibility. Based on different parameters extracted from the received video bitstream, we constructed several decision trees for modeling impairment visibility. These results are presented in section VI. Finally, conclusions are drawn in section VII.

II. Objective video quality metric classification

Objective video quality metrics can be classified into different categories based on several criteria as depicted in Figure 1. A first classification is made based on the availability of the original video sequence and the amount of information which is extracted from it. As such, objective quality metrics are classified into Full-Reference (FR), Reduced-Reference (RR) and No-Reference (NR) metrics. FR metrics, as standardized in [19] , require access to the complete original video sequence whereas NR metrics only estimate visual quality based on the received video sequence. The RR-based objective metrics extract certain features from the original video sequence and signal this information over an ancillary error-free channel which is then combined with the information extracted from the received video sequence [18]. Next, quality metrics can also be categorized based on the type of information or processing level where the information is extracted. Using this classification, metrics are called pixel-based, bitstream-based or parametric-based. Pixel-based metrics process the decoded video sequence in order to access pixel data whereas bitstream-based quality metrics extract the necessary information directly from the encoded video stream without fully decoding the sequence. By parsing the encoded video stream, information on the macroblocks and motion vectors can be obtained [21]. The third kind of metrics, called parametric video quality metrics, estimate quality using only information available in the packet headers in the case of streaming video delivery [22]. Video quality metrics can also use a combination of pixel, bitstream and network information for calculating perceived quality. Such objective metrics are also known as hybrid metrics and are currently being evaluated by the Video Quality Experts Group (VQEG) [23], [24].

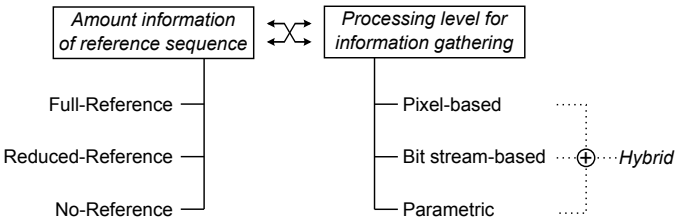

Fig. 1. Classification of video quality metrics based on the amount of information which is used from the reference sequence or based on the processing level for extracting information in order to model perceived quality.

In order to enable real-time impairment detection, we are targeting a no-reference bitstream-based objective quality metric which does neither require a complete decoding of the video stream nor access to the original video sequence. This way, quality monitoring can also be performed at intermediate measurement points along the video delivery channel where the decoded video stream itself is not available.

In this section, we only presented a brief overview of different types of objective video quality metrics. For a more complete and detailed overview of objective video quality metric classifications, the reader is referred to [25] or [26].

III. Visual quality impairment detection

Traditionally, most existing objective video quality metrics output a quality rating for the entire sequence which can be directly mapped to the Mean Opinion Score (MOS), as used during subjective video quality assessment [27]. However, as explained in the introduction, we are interested in real-time detection of visible impairments resulting from network failures during video playback. Consequently, our main objective is to determine when network impairments will be deemed visible or invisible to the end-users.

Suresh *et al.* [28] introduced the Mean Time Between Failures (MTBF), commonly used for measuring QoS, as a new means for subjective video quality assessment. The MTBF represents how often a typical viewer perceives a visual impairment and simplifies subjective testing as viewers only need to evaluate a video sequence and indicate when they perceive a visual artifact [29], [30]. Closely related to the MTBF, the authors define the PFAIL metric as the fraction of the viewers who find a given video portion of acceptable quality. During video playback, viewer's frequency of indicating the occurrence of visual impairments correlates well with perceived video quality. As such, lower quality video will result in a higher frequency of impairment detection and a lower MTBF.

In order to estimate the MTBF in real-time, the authors developed the Automatic Video Quality (AVQ) metric [31], [32], [33], capable of detecting both compression artifacts and network artifacts. The former are detected using a combination of the quantization step size and scene activity whereas the latter are detected during error concealment. Therefore, the AVQ metric requires access to both the network bitstream and the decoded video sequence [34].

Reibman *et al.* [35] used a decision tree to model the visibility of individual packet loss in the case of MPEG-2 encoded video. The classifier is built using a combination of parameters which are extracted from the received and decoded video stream and parameters which are computed from the original video sequence and signaled inside the bitstream. These parameters include, among others, the spatial and temporal extent of the error and the mean squared error of the initial error averaged over the macroblocks initially lost. As the latter can only be estimated from the complete bitstream, the proposed classifier can be regarded as a hybrid RR objective metric. In order to construct and validate the classifier, a subjective experiment was conducted during which test subjects had to indicate when they perceived a visual impairment. The followed methodology thus corresponds with measuring the MTBF as proposed by Suresh *et al.*

The classification accuracy when only considering parameters which can be extracted from the decoded video stream or the received encoded bitstream was further investigated by Kanumuri *et al.* [36]. Results indicate that there is a slight drop in performance when only considering NR parameters. However, no significant differences were found between the classifier built using parameters extracted only from the decoded pictures and the classifier built using parameters extracted from the encoded bitstream.

Based on a regression analysis, Kanumuri *et al.* also modeled the probability that a packet loss is visible to the average end-user by incorporating the same parameters used by the decision tree classifier. However, classifying packet loss visibility using regression resulted in a lower classification accuracy compared to classification based on the decision tree.

Reibman and Kanumuri both classify packet loss visibility in MPEG-2 encoded video sequences using two different detection thresholds. If 75% or more of the subjects perceived the impairment, the packet loss is classified as visible. When the visual degradation is detected by only 25% or less of the subjects, the packet loss remains invisible. All other packet losses which cannot be classified according to these two thresholds were not taken into account when constructing the decision tree. This implies that not every packet loss occurrence can be classified using the proposed decision tree.

In [37], Kanumuri *et al.* further applied a linear model to predict the probability that visual impairments in H.264/AVC encoded video sequences caused by isolated or multiple lost packets will be perceivable by the end-user. The study was focused on small Source Input Format (SIF) [1] resolution video sequences which were encoded using a fixed number of B-pictures and a fixed GOP length. The parameters building up the proposed linear model are extracted from the decoded (erroneous) video stream and the original encoded video sequence resulting in an RR objective quality metric. In [38], the authors used support vector regression to continuously predict packet loss visibility for SD and HD H.264/AVC encoded sequences based on parameters extracted and calculated solely from the received bitstream.

Lin *et al.* [39] combined the research of Reibman and Kanumuri and used the above mentioned visibility models to implement an efficient packet dropping policy algorithm for routers. Upon arrival in the router, each packet is labeled high or low priority depending on whether the loss of that particular packet would result in respectively a visible or an invisible degradation in the decoded video.

Research [40] has shown that the accuracy of predicting packet loss visibility can be increased when taking into account scene-level characteristics such as camera motion. Results show that impairments in scenes with a still camera are significantly less detectable compared to degradations in scenes with global camera motion. In general, camera motion (e.g., panning) does influence packet loss visibility. According to Liu *et al.* [41], saliency is also an important parameter for estimating packet loss visibility. Although, it has been shown [42] that impairments outside of the Region Of Interest (ROI) are only rated better quality in case of video content with a clear distinct ROI area.

In previous research [17], we proposed an NR bitstream-based visual quality impairment detector, called ViQID, for estimating packet loss visibility in H.264/AVC encoded video sequences of Common Interchange Format (CIF) [2] resolution. We also used a decision tree classifier to model packet loss visibility, but only using parameters which can be extracted solely from the received encoded video bitstream. Furthermore, the research was mainly focused on the effect and visibility of losing one or more entire pictures. Our results show that it is possible to classify packet loss visibility with high accuracy, using only a limited number of parameters.

The goal of the research presented in this article is to extend our existing classifier in order to enable packet loss visibility prediction of HD H.264/AVC encoded sequences taking into account NR parameters which can be estimated using only the received encoded video bitstream without the need for complete decoding. Furthermore, in this article we also consider multiple encoding settings.

IV. Assessing the visibility of network impairments

Prior to the construction of an objective video quality metric, proper training data and validation data must be collected [43]. Currently, the most reliable way for obtaining such data is using subjective video quality assessment where human observers evaluate the visual quality of a series of short video sequences. These sequences usually contain visual artifacts caused by encoding and/or network impairments. As such, in order to conduct subjective experiments, different video sequences must be selected, encoded and impaired. In the next sections, we will discuss in more details how different test sequences were selected, encoded and impaired. The impaired sequences where then used to conduct a subjective experiment in order to assess the visibility of visual artifacts caused by network impairments.

[1]SIF resolution corresponds with 352 by 240 pixels.

[2]CIF resolution corresponds with 352 by 288 pixels.

A. Source sequences selection and encoding

When assessing the influence of visual impairments on perceived quality, the type of video can have a significant impact [44], [45], [46]. It is generally known that, for example, the amount of motion and spatial details in a video sequence affect the visibility of visual degradations [47]. In order to quantify the spatial and temporal information of a video sequence, ITU Recommendation P.910 [44] defines two complexity measures which are calculated for each video frame at time n (F_n). The spatial perceptual information measurement (SI) is based on a Sobel-filter and is calculated as the maximum value of the standard deviations of each Sobel-filtered frame at time n:

$$SI = max_{time}\{std_{space}[Sobel(F_n)]\}. \quad (1)$$

The temporal perceptual information (TI) is similarly calculated as SI, but is based on the pixel difference between two consecutive frames:

$$TI = max_{time}\{std_{space}[M_n(i,j)]\}, \quad (2)$$

where $M_n(i,j) = F_n(i,j) - F_{n-1}(i,j)$.

Since both the SI and TI values for a video sequence are represented by the maximum value over all the video frames, peaks (caused by for example scene cuts) may result in an overall SI or TI value which is not representative for a particular video sequence [48]. Results showed that the human perception of spatial and temporal information in a video sequence can be better approximated by calculating the upper quartile value instead of the maximum value. Hence, the authors in [48] propose the $Q3.SI$ and $Q3.TI$ perceptual information measurements. These are calculated similar to equations 1 and 2, but the third quartile value (Q3) is calculated over all the video frames instead of taking the maximum SI and TI value.

In the case of subjective video quality testing, the selected test sequences should span a wide range of spatial and temporal information. Therefore, we calculated the $Q3.SI$ and $Q3.TI$ values for a series of test sequences available from the Consumer Digital Video Library (CDVL) [49][3], the Technical University of Munich (TUM) and open source movies and selected eight different sequences to be used in our subjective test. All original test sequences were taken from progressively scanned content with a resolution of 1920x1080 pixels and a frame rate of 25 frames per second. Each video sequence was trimmed to a duration of exactly 10 seconds. Figure 2 shows the calculated $Q3.SI$ and $Q3.TI$ values for our selected sequences and a short description of each sequence is provided in Table I. Sequences marked with a star indicate that the content is taken from an open source movie.

For encoding our eight selected HD sequences, we first collected realistic encoding settings by analyzing HD content from online video services and by inspecting the default settings recommended by commercially available H.264/AVC encoders. More specifically, we are interested in discovering the most commonly used number of slices per picture, the Group Of Pictures (GOP) size and the number of B-pictures. We found that the video sequences are usually encoded using 1, 4 or 8 slices per picture and a GOP structure containing 0, 1 or 2 B-pictures. We also noticed that different GOP sizes are used but an average GOP size between 12 and 15 frames is typically used in an IPTV environment [50]. In this article, we are primarily interested in the influence of network impairments on perceived quality. Therefore, the encoding bit rate was set high enough in order to avoid the presence of visual coding artifacts. Based on this analysis, we used the following encoding settings:

- Number of slices: 1, 4 and 8
- Number of B-pictures: 0, 1 and 2
- GOP size: 15 (0 or 1 B-picture) or 16 (2 B-pictures)
- Closed GOP structure
- Bit rate: 15 Mbps

This way, each original sequence was encoded using nine different configurations giving a total number of 72 encoded sequences. We also visually inspected each encoded sequence in order to ensure the absence of any coding artifacts.

[3]http://www.cdvl.org

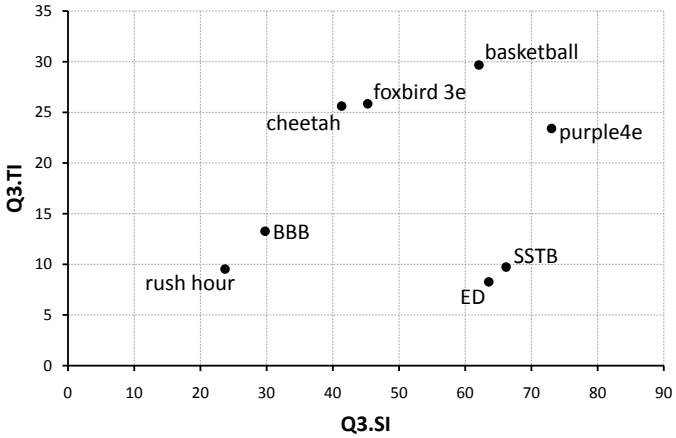

Fig. 2. Calculated Q3.SI and Q3.TI values, as suggested in [48], for our eight selected source sequences.

TABLE I
DESCRIPTION OF THE SELECTED TEST SEQUENCES.

Sequence	Source	Description
basketball	CDVL	Basketball game with score. Camera pans and zooms to follow the action.
BBB*	Big Buck Bunny	Computer-Generated Imagery. Close-up of a big rabbit. Slight camera pan while following a butterfly in front to the rabbit.
cheetah	CDVL	Cheetah walking in front of a chainlink fence. Camera pans to follow the cheetah.
ED*	Elephants Dream	Computer-Generated Imagery. Fixed camera focusing on two characters. Motion in the background.
foxbird3e	CDVL	Cartoon. Fox running towards a tree and falling in a hole. Fast camera pan with zoom.
purple4e	CDVL	Spinning purple collage of objects. Many small objects moving in a circular pattern.
rush hour	TUM	Rush hour in Munich city. Many cars moving slowly, high depth of focus. Fixed camera.
SSTB*	Sita Sings the Blues	Cartoon. Close-up of two characters talking. Slight camera zoom in.

B. Impairment generation

Any H.264/AVC compliant bitstream must only contain complete slices. Even in the case when only a small portion of a slice is lost, the entire slice should be discarded. Therefore, we impaired the encoded sequences by dropping particular slices. We used Sirannon[4] [51], our in-house developed open source modular multimedia streamer, in order to drop specific slices from an encoded H.264/AVC video stream according to the configuration depicted in Figure 3. In our setup, a raw H.264/AVC Annex B bitstream is first packetized into RTP packets according to RFC3984 [52]. No aggregation was used during the packetization process which implies that a single RTP packet never contains data belonging to more than one encoded picture. Next, slices are dropped by discarding all the RTP packets carrying data belonging to the same slice. After unpacketizing, the resulting impaired H.264/AVC Annex B compliant bitstream is saved to a new file.

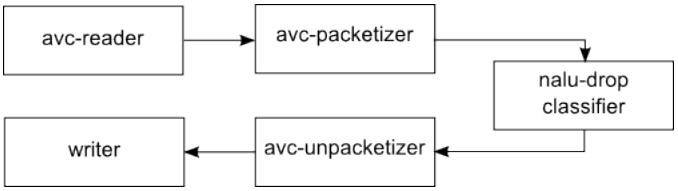

Fig. 3. RTP packets, which carry data from particular slices, are dropped using the *nalu-drop classifier* component. After unpacketizing, the resulting impaired sequence is saved to a new file.

We are interested in investigating the visibility of losing one or more slices and one or more entire pictures. Therefore, the following parameters were used to create different impairment scenarios:

- Number of B-pictures between two reference pictures (0, 1 or 2)
- Type of picture in which the first loss is inserted (I, P or B)
- Location within the GOP where the loss is inserted (begin, middle or end)
- Number of consecutive slice drops (1, 2 or 4)
- Location within the picture of the first dropped slice (top, middle or bottom)
- Number of consecutive entire picture drops (0 or 1)

In the case of consecutive slice drops, all dropped slices belonged to the same picture. As such, dropping two or four consecutive slices was not taken into account when the pictures were encoded using a single slice. Consecutive pictures were only dropped in sequences encoded with one slice per picture. Furthermore, we only considered dropping two consecutive P-pictures and two consecutive B-pictures.

Creating a full factorial [53] of all possible combinations of the parameters for impairing the sequences would result in 486 scenarios. However, not all combinations are feasible; for example, changing the location within the picture of the first dropped slice is only meaningful when the sequence is encoded using multiple slices. Therefore, all illegal combinations were removed after generating the full factorial. To further reduce the number of scenarios we searched for the least unique scenarios and removed them from the design. This will result in a design for which the remaining points lie as far away from each other as possible [54]. As each scenario is identified by a combination of the six parameters (6-tuple) described above, the least unique scenario is the one for which every parameter value occurs the most in the design. For example, in the following scenarios the third one is the least unique:

(1, I, begin, 2, bottom, 0)
(1, P, middle, 1, top, 1)
(2, P, begin, 2, top, 0)

Using this experimental design, 48 impairment scenarios were selected which we applied to our eight selected sequences resulting in a total number of 384 impaired video sequences. No visual impairments were injected in the first and last two seconds of video playback.

Since the impaired video sequences cannot be decoded properly with the H.264/AVC reference software except in the simplest cases of loss patterns, we adjusted the JM Reference Software version 16.1 [55] to enable error concealment in case of picture drops [56]. As a concealment technique, frame copy has been implemented.

C. Subjective quality assessment methodology

The different sequences were presented to the subjects based on the Single Stimulus (SS) Absolute Category Rating (ACR) assessment methodology as specified in [44]. Using this methodology implies that the video sequences are shown one at a time without the presence of an explicit reference sequence. This also corresponds with watching television, where viewers can only evaluate the received video signal [57], [58].

Prior to the start of the subjective experiment, all subjects received specific instructions on how to evaluate the different video sequences. After screening the subjects for color vision and visual acuity (using Ishihara plates and a Snellen chart, respectively), three training sequences were presented. This training session was used to get the subjects familiarized with the different kinds of impairments which could be perceived during the experiment. After watching each sequence, viewers first had to indicate whether they perceived a visual impairment. If the latter was the case, they were also asked to provide a quality score for that particular sequence using the 5-grade ACR scale depicted in Figure 4.

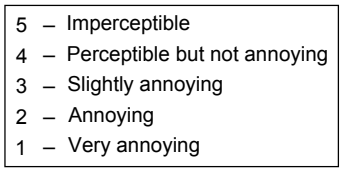

Fig. 4. Five-level grading scale [44] presented to the subjects, after each sequence, for recording impairment visibility and annoyance.

In order to avoid viewer fatigue, the overall experiment duration should not exceed 30 minutes. Therefore, we created

[4]Sirannon is formerly known as xStreamer.

six different datasets each containing 76 sequences, which include both original encoded and impaired video sequences. As such, the duration of each dataset was limited to about 20 minutes. The order of the video sequences within a single dataset was randomized at the start of the trail so that no two subjects evaluated the sequences in exactly the same order.

The video sequences were displayed on a 40 inch full HD LCD screen with subjects seated at a distance of 4 times the picture height.

A total number of 40 non-expert viewers, aged between 18 and 34 years old, participated with the subjective experiment. Amongst them, 11 were female and 29 were male test subjects. Each dataset was evaluated by exactly 24 subjects. As a result, most of the subjects evaluated more than one dataset although not necessarily on the same day. We also performed a post-experiment screening of our test subjects using the methodology described in Annex V of the VQEG HDTV report [20] in order to ensure no outliers were present in our subjective data. This methodology is based on the linear Pearson correlation coefficient and rejects a subject's quality scores in case the correlation with the average of all the other subjects' quality ratings drops below the acceptability threshold.

V. Decision tree-based modeling of impairment visibility

In this article, we are interested in determining whether the loss of some part of the video stream will result in a visible impairment. In other words, we want to be able to classify the occurrence of packet loss as visible or invisible for the average end-user.

Reibman *et al.* [35] and Kanumuri *et al.* [36] both used a decision tree classifier for modeling packet loss visibility, which we also used in our previous research [17].

A decision tree, as depicted in Figure 5, is built up using different nodes and end nodes (also known as leaves) and is traversed from top to bottom.

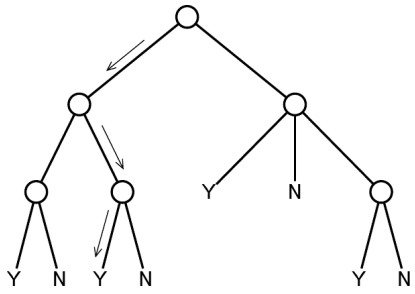

Fig. 5. Basic decision tree composed of different nodes and leaves which is traversed from top to bottom.

While traversing the tree, a decision on which path to follow down is made at every node based on the value of one or more of the attributes used to construct the tree. This evaluation continues until a leaf node is reached at which point the classification is completed. The label associated with this leaf node then determines, for example, error visibility.

The use of a decision tree for performing classification offers several advantages. First of all, the decision tree is a white box showing the complete internal structure of the classifier. This implies that an in-depth analysis can be performed which can lead to better insights and more conclusions on how the classification is performed. As the evaluation of a node in the decision tree comes down to an if-else evaluation, a decision tree can also easily be implemented. Another big advantage of decision trees is that they can handle both numerical and categorical parameters [59].

For evaluating the performance and reliability of a decision tree, the overall classification accuracy and the true positive (TP) rate can be considered. The classification accuracy is defined as the ratio between the number of correctly made classifications and the total number of classifications. In the case of classifying packet loss as visible or invisible, the TP rate for the visible packet losses represents the percentage of visible losses correctly classified as being visible.

VI. Results

For modeling packet loss visibility, parameters need to be extracted from the network and/or video bitstream in order to identify the location and extent of the initial loss. These parameters are then used for building different decision trees. In this section, we first provide an overview of the different parameters used for predicting impairment visibility. Next, we present different decision trees for classifying packet loss as visible of invisible to the average end-user.

A. Parameter extraction

As we are targeting a no-reference bitstream-based visual quality metric, only information extracted from the network and the received encoded (impaired) video stream is available for constructing our decision tree. Different parameters are extracted from the network and the video bitstream in order to identify the location of the loss and characterize the video sequence.

The location of the loss is identified by the type of the lost slice, the location of this slice within the picture and the GOP, and the number of consecutive slice losses.

Characterizing the pictures affected by the loss is performed by extracting information at the macroblock and the motion vector level. As such, we calculate the average motion vector lengths and standard deviations. Statistics concerning the macroblock partitions and types are also calculated.

All these statistics are calculated within the GOP containing the loss. In case the loss occurs in the I picture, the statistics are calculated from the remaining B and P pictures in the GOP. When the impairment originates from a P picture, statistics are calculated from the I picture (at the beginning of the GOP) and all the other P pictures in the GOP, the B-pictures are not used in this case. Similar when a loss occurs in a B picture, only the I picture and the other B-pictures in the GOP are taken into account when calculating the statistics.

An overview of all extracted parameters is listed in Table II.

B. Modeling packet loss visibility

For constructing different decision trees, we used the Waikato Environment for Knowledge Analysis (WEKA) [60],

TABLE II
OVERVIEW OF ALL PARAMETERS EXTRACTED FROM THE RECEIVED VIDEO BITSTREAM IN ORDER TO IDENTIFY THE LOCATION OF THE LOSS AND CHARACTERIZE THE VIDEO SEQUENCE.

Parameter	Description
b_pics, nb_slices, gop_size	Number of B-pictures, slices per picture and GOP size as specified during encoding.
contentclass	Sequence content classification (see Table IV).
imp_pic_type, perc_pic_lost	Type (I, P or B) and percentage of slices lost of the picture where the loss originates.
imp_in_gop_pos, imp_in_pic_pos	Temporal location within the GOP (begin, middle, end) and spatial location within the picture (top, bottom, middle) of the first lost slice.
imp_cons_slice_drops, imp_cons_b_slice_drops, imp_pic_drops	Number of consecutive slice drops, number of consecutive B-slice drops and number of entire picture drops.
drift	Temporal duration of the loss.
perc_pb_4x4, perc_pb_8x8, perc_pb_16x16, perc_pb_8x16, perc_pb_16x8, perc_i_4x4, perc_i_8x8, perc_i_16x16	Percentage of I, P & B macroblocks of type 4x4, 8x8, 16x16, 8x16 and 16x8, averaged over the pictures in the GOP containing the loss.
perc_i_mb, perc_skip, perc_ipcm	Percentage of macroblocks encoded as I, skip and PCM, averaged over the pictures in the GOP containing the loss.
I_perc_4x4, I_perc_8x8, I_perc_16x16	Percentage of macroblocks of type 4x4, 8x8 and 16x16 in the first I or IDR picture of the GOP containing the loss.
abs_avg_coeff, avg_qp	Absolute average value of the macroblock coefficients and QP value, averaged over the P or B pictures in the GOP containing the loss.
I_abs_avg_coeff, I_avg_qp	Absolute average value of the macroblock coefficients and QP value in the first I or IDR picture of the GOP containing the loss.
perc_zero_coeff, I_perc_zero_coeff	Percentage of zero coefficients, averaged over the P or B pictures in the GOP containing the loss and average of zero coefficients in the first I or IDR picture of the GOP containing the loss.
avg_mv_x, avg_mv_y, stdev_mv_x, stdev_mv_y	Average absolute motion vector length and standard deviation in x- and y-direction, averaged over the P or B pictures in the GOP containing the loss. Motion vector magnitudes have quarter pixel precision.
avg_mv_xy, stdev_mv_xy	Average and standard deviation of the sum of the motion vector magnitudes in x- and y-direction, averaged over the P or B pictures in the GOP containing the loss. Motion vector magnitudes have quarter pixel precision.
perc_zero_mv	Average percentage of zero motion vectors, calculated over the P or B pictures in the GOP containing the loss.

an open source data mining software package. As our total number of sequences is limited to 384, we used 10-fold cross validation for constructing and validating the built trees. During k-fold cross validation, the entire dataset is split into k subsets of which $k-1$ are used for building the tree and one dataset is used for validating the tree. This process is repeated exactly k times, each time selecting a different subset for validation. A common value used for k is 10, which minimizes the variance over the different runs [61].

During the subjective experiment, subjects were required to indicate whether they perceived a visual impairment or not. Based on these results, we classify packet loss to be visible when 75% or more of the subjects perceived the impairment. Otherwise, the impairment is classified as invisible. When plotting the Mean Opinion Score (MOS) of each sequence against the percentage of the subjects who perceived an impairment in the corresponding sequence, as depicted in Figure 6, we also noticed that the MOS drops below 4 starting from a detection threshold of 75%.

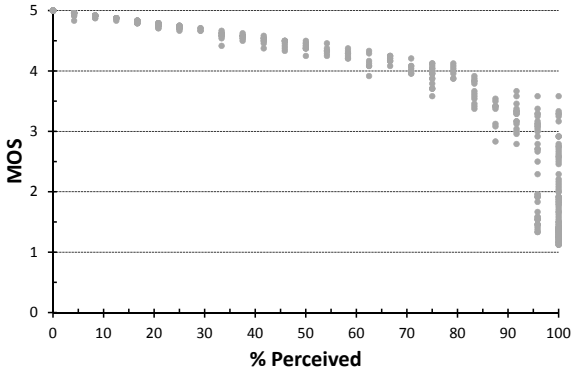

Fig. 6. Percentage of the viewers who perceived the impairment versus the MOS of the corresponding sequence.

As such, we classify packet loss visibility based on a single threshold as opposed to the two thresholds (75% and 25%) used by Reibman [35] and Kanumuri [36] as explained in section III. Our previous research [17] showed that the classification accuracy can be improved when considering the two thresholds mentioned above. However, the drawback of this approach is that not all packet losses can be classified, i.e. packet losses which have a detection threshold between 25% and 75%. By using a single detection threshold, we ensure that all losses will be classified as visible or invisible.

In our previous research [17], we used high-level information extracted from the bitstream for modeling packet loss visibility without the need for parsing the video data. Our results showed that the obtained decision trees had a high accuracy taking into account that only a limited amount of parameters were used during the modeling process. Using the data obtained in this article, we start off by modeling a decision tree using the following high level parameters: `imp_pic_type`, `imp_in_gop_pos`, `imp_in_pic_pos`, `imp_cons_slice_drops`, `imp_cons_b_slice_drops`, `perc_pic_lost`, `imp_drop_next_pic`.

The resulting tree, depicted in Figure 7a, shows that only five parameters are needed to classify packet loss visibility.

Looking at the tree into more detail, we see that a loss of up to two B-pictures is not perceivable and that losses in I-pictures are always perceived, even if only one out of eight slices is lost. In case the loss originates from a P-picture, error visibility depends on the percentage of the slices lost and the location of the P-picture within the GOP. To be more precise, when only a small portion of the slices is lost, the impairment is not perceived. When more than 25% of the slices in the picture is lost, impairment visibility is determined by the

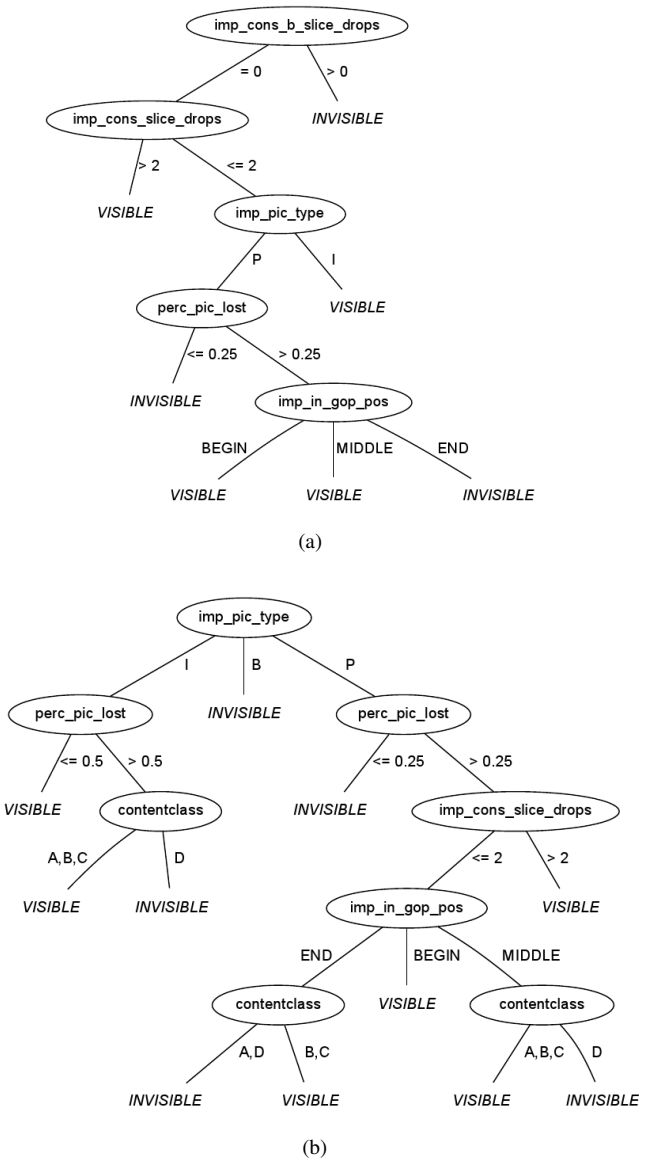

Fig. 7. Decision trees for classifying the occurrence of packet loss as visible or invisible to the average end-users, using only high level parameters extracted solely from the received encoded video bitstream (a) and with additional content classification (b).

temporal extent of the error (drift). As such, impairments are not visible in case the packet loss affects a P-picture located at the end of the GOP which results in a short drift of the error. Table III shows the average drift caused by losses in P-pictures, depending on the location of that picture within the GOP. In our experimental design, we dropped up to four slices in our sequences encoded with eight slices. As such, the branch `imp_cons_slice_drops > 2` implies that the errors are always perceived when 50% of a P- or I-picture is lost in these sequences. Our data analysis showed that impairments are not always perceived when 50% of a picture, encoded with four slices, is lost. Hence, a lower number of slices per picture might be preferred as this appears to be better in terms of error visibility.

The overall classification accuracy of this tree equals 83.1%. The TP rates for visible and invisible impairments are respectively 84.0% and 82.1%. Taking into account that only a limited number of high-level parameters are used, packet loss visibility can be predicted with a high accuracy.

TABLE III
CALCULATED AVERAGE DRIFT (TEMPORAL EXTENT) AND STANDARD DEVIATIONS OF IMPAIRMENTS ORIGINATING FROM PACKET LOSS IN P-PICTURES, DEPENDING ON THE LOCATION WITHIN THE GOP OF THE FIRST AFFECTED P-PICTURE.

	Location within GOP		
	BEGIN	**MIDDLE**	**END**
avg(drift)	14	9	4
stdev(drift)	1	2	2

Results in [17] and [41] show that the prediction accuracy can be increased when taking into account the video content and characteristics. Therefore, we clustered our eight sequences (cfr. Figure 2) into four different content classes as shown in Table IV and make this additional parameter ('contentclass') available to the modeling process.

TABLE IV
CLUSTERING OF THE DIFFERENT VIDEO SEQUENCES, BASED ON THE AMOUNT OF MOTION AND SPATIAL DETAILS, INTO FOUR CONTENT CLASSES.

Content class	**Characteristics**	**Sequences**
A	low motion, low spatial details	BBB, rush hour
B	high motion, medium spatial details	cheetah, foxbird3e
C	high motion, high spatial details	basketball, purple4e
D	low motion, high spatial details	ED, SSTB

As can be seen in Figure 7b, including the content classification increases the overall complexity of the tree in terms of tree size. However, still only five parameters are used throughout the entire tree.

Similar to the previous tree, losses in B-pictures are never detected, independent of the type of video. According to the tree, content classification becomes an important factor in case packet loss occurs in I- or P-pictures. Perceptibility of impairments originating in I-pictures depends on the number of slices lost. During our impairment generation, we dropped 25%, 50% or 100% of the slices belonging to a particular picture. As such, the branch corresponding with `perc_pic_lost > 0.5` implies that an entire I-picture is lost. At this point, our error concealment comes into play and shows that impairments can be masked in sequences with low amounts of motion (content class D). The fact that packet loss impairments are less visible in video sequences with still camera motion also corresponds with the research findings of Reibman *et al.* [40]. From Figure 2 it can be seen that the amount of motion in the *rush hour* sequence corresponds with the sequences of content class D. According to the tree, losing an entire I-picture in content class A sequences results in a visible impairment. However, inspecting the classification accuracy of that branch revealed that only 50% of the predictions is correct. The data analysis showed that loosing an entire I-picture in the *rush hour* sequence is never perceived whereas the loss is perceived in case it occurs in the *BBB* sequence. This indicates that

it might be better to drop an entire I-picture (even if only a limited number of slices is lost) in low motion sequences and use the concealment at the decoder to mask the error. In our case, low motion sequences are characterized by a $Q3.TI$ value $\leq$ 13.

Losses originating from P-pictures are not perceivable in case only a small portion of the picture is lost. As was the case in our previous tree, the branch `imp_cons_slice_drops > 2` again only applies to the sequences encoded with eight slices per picture. Again, losing 50% of the slices in our sequences encoded with eight slices per picture results in a visible impairment. The path `imp_cons_slice_drops <= 2` corresponds with losing either 50% or 100% of a picture encoded with one or four slices per picture. In that case, impairment visibility again depends on the location of the P-picture within the GOP but also on the content type. Corresponding with our previous tree, losses occurring in P-pictures located in the beginning of the GOP result in a large drift (cfr. Table III) and always result in a visible impairment. When the P-picture, where the loss originates from, is located near the middle or the end of the GOP, impairments are again masked in low motion sequences.

By taking into account content classification, the accuracy of the tree increases up to 86.3%. The TP rate for correctly predicting visible impairments is in this case 84.6%, which is more or less the same compared to our previous tree. The TP rate for the invisible impairments increases to 88.3%, which is an increase of more than 6%. As such, less false alarms of visible impairments are triggered by this tree.

In [41], the authors showed that visual attention influences packet loss visibility. Furthermore, motion contrast and scene movement also determines visual attention [62]. According to both trees depicted in Figure 7, the location of the loss within the picture itself is not an important parameter for determining packet loss visibility. However, Figure 7b shows that packet loss visibility is influenced by the temporal duration of the impairment and the amount of motion in the video sequence. This indicates that impairment drift has a higher impact on error visibility compared to the initial spatial location of the loss.

Incorporating content classification, as proposed in the tree depicted in Figure 7b, implies that this classification is performed as a pre-processing step and signaled as part of the bitstream. In the case of RTP streaming, an RTP extension header [63] could be used to signal this side information. As part of the H.264/AVC coding standard, content class identification could be included as unregistered user data Supplemental Enhancement Information (SEI) messages [64].

Instead of using content classification as a pre-processing step before streaming the video sequence, content information can also be extracted from the encoded bitstream [65], [66]. In [67], for example, the amount of details and motion in a video sequence is determined based on the encoded frame sizes of I-, P- and B-pictures. As explained in section VI-A, we also extracted information at the macroblock and motion vector level in order to characterize the pictures affected by the loss (see Table II). In a last step towards modeling packet loss visibility, we replace the content classification by including the different extracted parameters as part of the decision tree construction process resulting in the tree depicted in Figure 8.

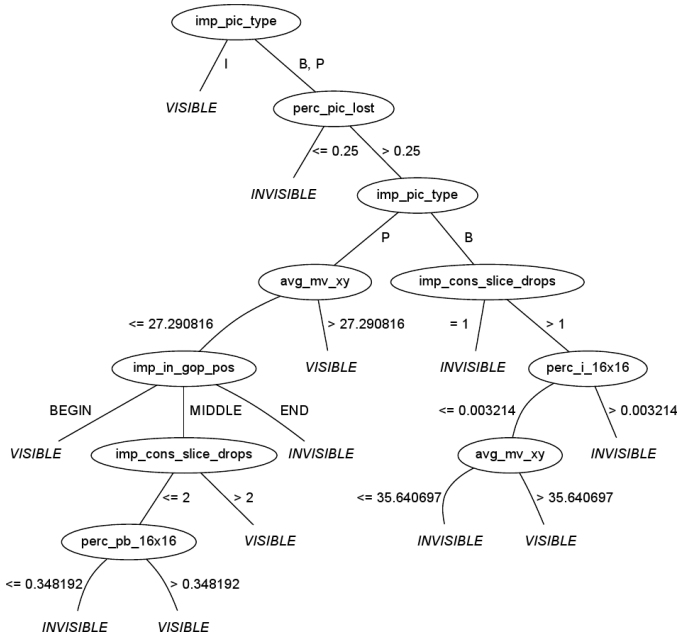

Fig. 8. Decision tree for classifying packet loss visibility based on parameters extracted solely from the received (impaired) encoded video bitstream.

Similar to the tree from Figure 7b, impairment visibility is first determined based on the type of picture where the loss originates from. Losses occurring in I-pictures are now generally classified as deemed visible to the average end-user. However, inspecting our data into more detail showed that, in correspondence with the previous proposed tree, losses of entire I-pictures can be masked by the employed error concealment. When only a very small portion (up to 25%) of a B- or P-picture is missing, impairments are usually not perceived.

In case of losses in B-pictures, the impairment is again properly concealed when entire pictures are dropped. The branch `imp_cons_slice_drops > 1` refers to losses of half a picture or losing two consecutive B-pictures. In that case, impairment visibility is content dependent and more clearly visible in high motion areas. Content is identified based on the distribution of the macroblocks and the average length of the motion vectors. As mentioned in section VI-A, these parameters are calculated only using the information available in the correctly received B-pictures of the current GOP.

The amounts of motion and spatial details are also important factors when more than 25% of a P-picture is lost. According to the tree, impairments are again easier detected in areas with high motion, corresponding with our previous trees and our previous research [17]. The classification `avg_mv_xy <= 27.290816` corresponds in our case typically with sequences belonging to content classes A or D. In that case, impairment visibility further depends on the location of the loss within the GOP which, in turn, affects the impairment drift (cfr. Table III). If the slice loss occurs in a P-picture located near the middle of the GOP, impairments are easier detected in our sequences encoded with eight slices

(`imp_cons_slice_drops > 2`), similar to our previous tree. When losing two out of four slices or two consecutive P-pictures, impairment visibility is dependent on the amount of spatial information. The parameter `perc_pb_16x16` refers to the average percentage of inter coded 16x16 macroblock partitions in the correctly received P-pictures of the current GOP. The split `perc_pb_16x16 <= 0.348192` corresponds, in our case, with sequences from content class D. This indicates that impairments are masked in areas with high amounts of spatial detail.

In general, the overall structure of the tree presented in Figure 8 is very similar to the one depicted in Figure 7b except for the fact that no explicit content classification is required as a pre-processing step. The classification accuracy of the last tree equals 85.5% which is also more or less the same compared to the tree with explicit content classification. The TP rates for correctly classifying visible and invisible impairments are respectively 87.8% and 83.2%. This is a small shift compared to the previous tree but indicates that the tree is slightly more accurate towards predicting visible impairments.

To summarize, a performance comparison of the different trees is provided in Table V.

TABLE V
PERFORMANCE COMPARISON OF THE DIFFERENT DECISION TREE.

	Only high-level parameters	**High-level parameters + content class**	**Full bitstream processing**
Overall accuracy	83.1%	86.3%	85.6%
TP rate visible	84.0%	84.6%	87.8%
TP rate invisible	82.1%	88.3%	83.2%

It is clear that taking into account the type of content improves classification accuracy. However, no clear difference is noticed in case this content classification is performed as a pre-processing step or if the type of video content is identified by extracting temporal and spatial information directly from the encoded video stream. The latter influences the depth at which the encoded video stream must be processed which, in turn, can result in a higher processing complexity requiring more processing power [68].

VII. CONCLUSION

In this article, we presented a novel no-reference bitstream based objective video quality metric which enables real-time detection of visual degradations caused by network impairments such as packet loss. Based on parameters extracted solely from the received encoded video bitstream, different decision tree based classifiers are constructed which classify each occurrence of packet loss as visible or invisible to the average end-user. The parameters used during the modeling process range from high level parameters up to the level of the macroblocks and the motion vectors.

Our in-depth analysis of the different decision trees showed that errors are less visible in low motion sequences and that dropping entire pictures and relying on the error concealment can result in better error masking in low motion sequences.

We also found that impairment visibility when loosing up to half of a picture depends on the number of encoded slices per picture. In our case, impairments are easier detected in the sequences encoded with eight slices per picture compared to the sequences encoded with only four slices.

Video content also plays a significant role in determining error visibility. By taking into account the type of content during the decision tree modeling process, impairment visibility can be estimated with a significant higher accuracy. We also showed that, instead of performing this content classification as a pre-processing step, information concerning the macroblock distribution and the magnitude of the motion vectors can be extracted from the received video bitstream and used for content classification. This approach no longer requires a pre-processing step, but requires more in-depth processing of the received video stream.

Packet loss visibility can also be estimated in a reliable way by only taking into account high level parameters. Pre-processing or in-depth processing of the received video sequence is therefore not required.

Overall, our results show that impairment visibility can be determined with high accuracy, even by only taking into account a limited number of high-level parameters. This enables content providers to continuously monitor video quality and check conformance with current existing objective QoE performance indicators as defined in ITU-T Recommendation G.1080 or Technical Report TR-126.

ACKNOWLEDGMENT

The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT) and the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT). This paper is the result of research carried out as part of the OMUS project funded by the IBBT. OMUS is being carried out by a consortium of the industrial partners: Excentis, Streamovations, Technicolor and Televic in cooperation with the IBBT research groups: IBCN, WiCa & Multimedia Lab (UGent), SMIT (VUB), PATS (UA) and COSIC (KUL).

Glenn Van Wallendael and Jan De Cock would also like to thank the Institute for the Promotion of Innovation through Science and Technology in Flanders for financial support through their Ph.D. and postdoctoral grant, respectively.

REFERENCES

[1] ITU-T Recommendation P.10/G.100 Amd 2, "Vocabulary for performance and quality of service," International Telecommunication Union (ITU), 2008.

[2] DSL Forum Technical Report TR-126, "Triple-play Services Quality of Experience (QoE) requirements," DSL Forum, 2006.

[3] K. Yamori and Y. Tanaka, "Relation between willingness to pay and guaranteed minimum bandwidth in multiple-priority service," in *The 2004 Joint Conference of the 10th Asia-Pacific Conference on Communications and the 5th International Symposium on Multi-Dimensional Mobile Communications.*, vol. 1, August 2004, pp. 113 – 117.

[4] M. Ries, O. Nemethova, and M. Rupp, "On the willingness to pay in relation to delivered quality of mobile video streaming," in *International Conference on Consumer Electronics (ICCE 2008)*, January 2008, pp. 1–2.

[5] G. Cermak, "Consumer Opinions About Frequency of Artifacts in Digital Video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 336 –343, April 2009.

[6] F. Boulos, B. Parrein, P. Le Callet, and D. Hands, "Perceptual effects of packet loss on H.264/AVC encoded videos," *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-09)*, January 2009.

[7] F. De Simone, M. Naccari, M. Tagliasacchi, F. Dufaux, S. Tubaro, and T. Ebrahimi, "Subjective assessment of H.264/AVC video sequences transmitted over a noisy channel," in *International Workshop on Quality of Multimedia Experience (QoMEX)*, July 2009, pp. 204 –209.

[8] M. Pinson, S. Wolf, and G. Cermak, "HDTV subjective quality of H.264 vs. MPEG-2, with and without packet loss," *IEEE Transactions on Broadcasting*, vol. 56, no. 1, pp. 86–91, March 2010.

[9] A. Perkis, "Does quality impact the business model? case: Digital cinema," in *International Workshop on Quality of Multimedia Experience (QoMEX)*, July 2009, pp. 151–156.

[10] R. Serral-Gracià, E. Cerqueira, M. Curado, M. Yannuzzi, E. Monteiro, and X. Masip-Bruin, "An overview of quality of experience measurement challenges for video applications in IP networks," in *Wired/Wireless Internet Communications*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6074, pp. 252–263.

[11] J. Asghar, F. Le Faucheur, and I. Hood, "Preserving video quality in IPTV networks," *IEEE Transactions on Broadcasting*, vol. 55, no. 2, pp. 386 –395, june 2009.

[12] M. Claypool and J. Tanner, "The effects of jitter on the peceptual quality of video," in *Proceedings of the seventh ACM international conference on Multimedia (Part 2)*, ser. MULTIMEDIA '99. New York, NY, USA: ACM, 1999, pp. 115–118. [Online]. Available: http://doi.acm.org/10.1145/319878.319909

[13] R. Pastrana-Vidal, J. Gicquel, C. Colomes, and H. Cherifi, "Sporadic frame dropping impact on quality perception," *Human Vision and Electronic Imaging IX*, vol. 5292, 2004.

[14] S. Gulliver and G. Ghinea, "The perceptual and attentive impact of delay and jitter in multimedia delivery," *IEEE Transactions on Broadcasting*, vol. 53, no. 2, pp. 449–458, June 2007.

[15] ITU-T Recommendation G.1080, "Quality of Experience requirements for IPTV services," International Telecommunication Union (ITU), 2008.

[16] N. Staelens, S. Moens, W. Van den Broeck, I. Mariën and, B. Vermeulen, P. Lambert, R. Van de Walle, and P. Demeester, "Assessing quality of experience of IPTV and Video on Demand services in real-life environments," *IEEE Transactions on Broadcasting*, vol. 56, no. 4, pp. 458–466, December 2010.

[17] N. Staelens, N. Vercammen, Y. Dhondt, B. Vermeulen, P. Lambert, R. Van de Walle, and P. Demeester, "ViQID: A no-reference bit stream-based visual quality impairment detector," in *Second International Workshop on Quality of Multimedia Experience (QoMEX)*, June 2010, pp. 206 –211.

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

[19] ITU-T Recommendation J.247, "Objective perceptual multimedia video quality measurement in the presence of a full reference," International Telecommunication Union (ITU), 2008.

[20] Video Quality Experts Group (VQEG), "Report on the Validation of Video Quality Models for High Definition Video Content," June 2010. [Online]. Available: http://www.its.bldrdoc.gov/vqeg/projects/hdtv/

[21] F. Yang, S. Wan, Q. Xie, and H. R. Wu, "No-reference quality assessment for networked video via primary analysis of bit stream," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1544–1554, November 2010.

[22] M. Garcia and A. Raake, "Parametric packet-layer video quality model for IPTV," in *10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA)*, May 2010, pp. 349–352.

[23] Video Quality Experts Group (VQEG), "Hybrid Perceptual/Bitstream Group Test Plan," ftp://vqeg.its.bldrdoc.gov/Documents/Projects/hybrid/, April 2011.

[24] N. Staelens, I. Sedano, M. Barkowsky, L. Janowski, K. Brunnström, and P. Le Callet, "Standardized toolchain and model development for video quality assessment - the mission of the joint effort group in VQEG," in *Third International Workshop on Quality of Multimedia Experience (QoMEX)*, September 2011.

[25] S. Winkler and P. Mohandas, "The evolution of video quality measurement: From PSNR to hybrid metrics," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 660–668, September 2008.

[26] S. Chikkerur, V. Sundaram, M. Reisslein, and L. J. Karam, "Objective video quality assessment methods: A classification, review, and performance comparison," *IEEE Transactions on Broadcasting*, 2011.

[27] ITU-R Recommendation BT.500, "Methodology for the subjective assessment of the quality of television pictures," International Telecommunication Union (ITU), 2002.

[28] N. Suresh and N. Jayant, "Mean Time Between Failures: A Functional Quality Metric for Consumer Video," *First International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-05)*, January 2005.

[29] ——, "Mean time between failures: A subjectively meaningful video quality metric," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, May 2006, p. II.

[30] N. Suresh, N. Jayant, and O. Yang, "Mean Time Between Failures: A Subjectively Meaningful Metric for Consumer Video," *Second International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-06)*, January 2006.

[31] N. Suresh, O. Yang, and N. Jayant, "AVQ: A Zero-reference Metric for Automatic Measurement of the Quality of Visual Communications," *Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-05)*, January 2007.

[32] N. Suresh, P. Mane, and N. Jayant, "Real-Time Prototype of a Zero-Reference Video Quality Algorithm," in *International Conference on Consumer Electronics (ICCE)*, January 2008, pp. 1–2.

[33] N. Suresh, R. Palaniappan, P. Mane, and N. Jayant, "Testing of a No-Reference VQ Metric: Monitoring Quality and Detecting Visible Artifacts," *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-05)*, January 2009.

[34] R. Palaniappan, N. Suresh, and N. Jayant, "Objective Measurement of Transcoded Video Quality in Mobile Applications," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2008, pp. 1–6.

[35] A. Reibman, S. Kanumuri, V. Vaishampayan, and P. Cosman, "Visibility of individual packet losses in MPEG-2 video," in *International Conference on Image Processing*, vol. 1, October 2004, pp. 171–174.

[36] S. Kanumuri, P. Cosman, A. Reibman, and V. Vaishampayan, "Modeling packet-loss visibility in MPEG-2 video," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 341–355, April 2006.

[37] S. Kanumuri, S. Subramanian, P. Cosman, and A. Reibman, "Predicting H.264 packet loss visibility using a generalized linear model," in *International Conference on Image Processing*, October 2006, pp. 2245–2248.

[38] S. Argyropoulos, A. Raake, M. Garcia, and P. List, "No-reference video quality assessment for SD and HD H.264/AVC sequences based on continuous estimates of packet loss visibility," in *Third International Workshop on Quality of Multimedia Experience (QoMEX)*, September 2011, pp. 31–36.

[39] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman, and A. Reibman, "A versatile model for packet loss visibility and its application to packet prioritization," *IEEE Transactions on Image Processing*, vol. 19, no. 3, pp. 722–735, March 2010.

[40] A. R. Reibman and D. Poole, "Predicting packet-loss visibility using scene characteristics," in *Packet Video 2007*, nov. 2007, pp. 308 –317.

[41] T. Liu, X. Feng, A. Reibman, and Y. Wang, "Saliency inspired modeling of packet-loss visibility in decoded videos," *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-09)*, January 2009.

[42] M. Mu, R. Gostner, A. Mauthe, G. Tyson, and F. Garcia, "Visibility of individual packet loss on H.264 encoded video stream: a user study on the impact of packet loss on perceived video quality," *Multimedia Computing and Networking*, 2009.

[43] M. Farias and S. Mitra, "A Methodology for Designing No-Reference Video Quality," *Fourth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-09)*, January 2009.

[44] ITU-T Recommendation P.910, "Subjective video quality assessment methods for multimedia applications," International Telecommunication Union (ITU), 1999.

[45] Y. Zhong, I. Richardson, A. Sahraie, and P. McGeorge, "Influence of task and scene content on subjective video quality," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, A. Campilho and M. Kamel, Eds. Springer Berlin / Heidelberg, 2004, vol. 3211, pp. 295–301.

[46] P. Corriveau, *Video Quality Testing*, ser. Digital Video Image Quality and Perceptual Coding. CRC Press, 2006, ch. 4, pp. 125–153.

[47] G. E. Legge and J. M. Foley, "Contrast masking in human vision," *Journal of the Optical Society of America*, vol. 70, no. 12, pp. 1458–1471, Dec 1980.

[48] A. Ostaszewska and R. Kloda, "Quantifying the amount of spatial and temporal information in video test sequences," in *Recent Advances in Mechatronics*. Springer Berlin Heidelberg, 2007, pp. 11–15.
[49] M. Pinson, S. Wolf, N. Tripathi, and C. Koh, "The consumer digital video library," *Fifth International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM-10)*, January 2010.
[50] G. O'Driscoll, *Next Generation IPTV Services and Technologies*. New York, NY, USA: Wiley-Interscience, 2008.
[51] A. Rombaut, N. Staelens, N. Vercammen, B. Vermeulen, and P. Demeester, "xStreamer: Modular Multimedia Streaming," in *Proceedings of the seventeenth ACM international conference on Multimedia*, 2009, pp. 929–930.
[52] S. Wenger, M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer, "RTP Payload Format for H.264 Video," February 2005.
[53] D. C. Montgomery, *Design and Analysis of Experiments*. John Wiley & Sons, 2008.
[54] K. Crombecq, E. Laermans, and T. Dhaene, "Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling," *European Journal of Operational Research*, vol. 214, no. 3, pp. 683 – 696, 2011.
[55] Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, "Doc. JVT-AE010: H.264/14496-10 AVC Reference Software Manual," MPEG / ITU-T, Tech. Rep., Jun. 2009.
[56] N. Staelens and G. Van Wallendael, "Adjusted JM Reference Software 16.1 with XML Tracefile Generation Capabilities," VQEQ_JEG_Hybrid_2011_029_jm with xml tracefile_v1.0, Hillsboro, Oregon, US, December 2011.
[57] S. Winkler, *Digital Video Quality - Vision Models and Metrics*. John Wiley & Sons, 2005.
[58] Q. Huynh-Thu, M.-N. Garcia, F. Speranza, P. Corriveau, and A. Raake, "Study of rating scales for subjective quality assessment of high-definition video," *IEEE Transactions on Broadcasting*, vol. 57, no. 1, pp. 1–14, march 2011.
[59] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
[60] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.
[61] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143.
[62] O. Le Meur, A. Ninassi, L. C. P., and D. Barba, "Overt visual attention for free-viewing and quality assessment tasks. impact of the regions of interest on a video quality metric." *Elsevier Signal Processing: Image Communication*, vol. 25, pp. 547–558, August 2010.
[63] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Morgan Kaufmann, 2007.
[64] ITU-T Recommendation H.264, "Advanced video coding for generic audiovisual services," International Telecommunication Union (ITU), 2010.
[65] T. Yap-Peng, D. Saur, S. Kulkami, and P. Ramadge, "Rapid estimation of camera motion from compressed video with application to video annotation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 133 –146, feb 2000.
[66] S. Jeannin and A. Divakaran, "Mpeg-7 visual motion descriptors," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 720–724, 2001.
[67] A. Clark, "Method and system for viewer quality estimation of packet video streams," U.S. Patent 2009/0041114 A1, February 2009.
[68] N. Vercammen, N. Staelens, B. Vermeulen, and P. Demeester, "Distributed video quality monitoring," in *2nd International Conference on Computer Science and its Applications*, December 2009, pp. 1–5.

Nicolas Staelens obtained his Master's degree in Computer Science at Ghent University (Belgium, 2004). He started his career in 2004 as an R&D engineer at Televic (a Belgian company that develops, designs and manufactures high-end network systems and software applications for the healthcare market). In 2006, he joined the IBCN-group (Internet Based Communication Networks and Services) at Ghent University as a Ph.D. student. His research focuses on studying the effects of network impairments on the perceived quality of audiovisual sequences. As of 2007, he is also actively participating within the Video Quality Experts Group (VQEG) and is currently co-chair of the Tools and Subjective Labs support group and the JEG-Hybrid project.

Glenn Van Wallendael obtained the M.Sc. degree in Applied Engineering from the University College of Antwerp, Antwerp, Belgium, in 2006 and the M.Sc. degree in Engineering from Ghent University, Ghent, Belgium in 2008. Since then, he is working towards a Ph.D. at Multimedia Lab, Ghent University, with the financial support of the Agency for Innovation by Science and Technology (IWT). Main topics of interest are video compression including multiview video compression and transcoding.

Karel Crombecq received his masters degree in Computer Science from the University of Antwerp, Belgium in 2006. Currently he is a PhD student with the CoMP research group at the department of Computer Science and Mathematics of the University of Antwerp, Belgium. He is funded by an FWO fellowship (Flemish Fund for Scientific Research). His research interests lie in distributed surrogate modeling, sequential experimental design and machine learning.

Nick Vercammen received his degree in Master of Industrial Sciences in Electromechanics in 2000 from the KaHo Sint-Lieven (Belgium). In 2001, he obtained the degree of Master in Applied Informatics at Ghent University. From 2001 to 2006, he worked as a software engineer for different industrial companies. In 2006, he joined the IBCN-group (Internet Based Communication Networks and Services) at Ghent University. His main research interests include video quality measurements in the network, automation of video quality testing and real-time video traffic analysis.

Jan De Cock (M06) obtained the M.S. and Ph.D. degrees in Engineering from Ghent University, Belgium, in 2004 and 2009, respectively. Since 2004 he has been working at Multimedia Lab, Ghent University, and the Interdisciplinary Institute for Broadband Technology (IBBT). In 2010, he obtained a Post-doctoral Research Fellowship from the Agency for Innovation by Science and Technology (IWT). His research interests include (high-efficiency) video compression and transcoding, scalable video coding, and multimedia applications..

Brecht Vermeulen received his Electronic Engineering degree in 1999 from Ghent University, Belgium. In June 2004, he received his Ph.D. degree for the work entitled 'Management architecture to support quality of service in the internet based on IntServ and DiffServ domains.' at the department of Information Technology of Ghent University. Since June 2004, he is leading a research team within the IBCN group of Prof. Demeester which investigates network and server performance and quality of experience in the fields of video, audio/voice and multiple play. Since the start of IBBT (Interdisciplinary institute for BroadBand Technology) in 2004, he leads also the IBBT Technical Test Centre in Ghent, Belgium.

Rik Van de Walle received his M.Sc. and Ph.D. degrees in Engineering from Ghent University, Belgium in 1994 and 1998 respectively. After a visiting scholarship at the University of Arizona (Tucson, USA), he returned to Ghent University, where he became professor of multimedia systems and applications, and head of the Multimedia Lab. His current research interests include multimedia content delivery, presentation and archiving, coding and description of multimedia data, content adaptation, and interactive (mobile) multimedia applications.

Tom Dhaene (M94, SM05) was born in Deinze, Belgium, on June 25, 1966. He received the Ph.D. degree in electrotechnical engineering from the University of Ghent, Ghent, Belgium, in 1993. From 1989 to 1993, he was Research Assistant at the University of Ghent, in the Department of Information Technology, where his research focused on different aspects of full-wave electro-magnetic circuit modeling, transient simulation, and time-domain characterization of high-frequency and high-speed interconnections. In 1993, he joined the EDA company Alphabit (now part of Agilent). He was one of the key developers of the planar EM simulator ADS Momentum. Since September 2000, he has been a Professor in the Department of Mathematics and Computer Science at the University of Antwerp, Antwerp, Belgium. Since October 2007, he is a Full Professor in the Department of Information Technology (INTEC) at Ghent University, Ghent, Belgium. As author or co-author, he has contributed to more than 220 peer-reviewed papers and abstracts in international conference proceedings, journals and books.

Piet Demeester received the Master's degree in Electro-technical engineering and the Ph.D degree from the Ghent University, Gent, Belgium in 1984 and 1988, respectively. He is a full-time professor at Ghent University where he teaches courses in communication networks. He is the head of the Internet Based Communication Networks and Services group. His research interests include: multilayer IP-optical networks, mobile networks, end-to-end quality of service, grid computing, network and service management, distributed software and multimedia applications. He has published over 500 papers in these areas in international journals and conference proceedings. In this research domain he was and is a member of several program committees of international conferences, such as: OFC, ECOC, ICC, Globecom, Infocom and DRCN. He is a fellow of the IEEE.