

EFFICIENT FRAME COMPLEXITY ESTIMATION AND APPLICATION TO G.1070 VIDEO QUALITY MONITORING

Beibei Wang, Dekun Zou, Ran Ding, Tao Liu, Sitaram Bhagavathy, Nirranjan Narvekar, Jeffrey Bloom

Dialogic Media Labs, 12 Christopher Way, Suite 104, Eatontown, NJ 07724, USA
{beibei.wang, dekun.zou, ran.ding, tao.liu, sitaram.bhagavathy, nirranjan.narvekar, jeffrey.bloom}@dialogic.com

ABSTRACT

ITU has standardized a computational model as Recommendation G.1070 for Quality of Experience (QoE) planning [1]. In our previous work, we proposed a system for calculating the G.1070 visual quality estimate in a monitoring scenario [2]. In G.1070, the visual quality is based, in part, on frame rate, bitrate, and packet-loss rate. For a fixed frame rate and a fixed packet-loss rate, the G.1070 visual quality score will decrease with decreases in bitrate. However, G.1070 cannot distinguish between cases in which a decrease in bitrate truly does represent a decrease in quality and cases in which the underlying content is easy to encode, thus resulting in a lower bitrate without a corresponding decrease in quality. In this paper, we propose a modification to G.1070 model to account for this difference by including an analysis of the underlying complexity of the video content. More specifically, we propose a quality measure in which the bitrate input to G.1070 is replaced with a normalized bitrate, where the normalization is based on an estimate of the complexity of the compressed content. With this proposed enhancement to the model (named as G.1070E), it allows a much better approximation to MOS values and the NTIA-VQM[3].

Index Terms— QoE, G.1070, content complexity, bitrate normalization

1. INTRODUCTION

With the proliferation of broadband multimedia access networks, there has been an increasing need for effective ways to monitor the perceptual quality of video information, communications, and entertainment (also referred to as “quality of experience” or “QoE”). The International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) has standardized a computational model for estimating such QoE. Known as ITU-T Recommendation G.1070 “Opinion model for video-telephony applications” or the “G.1070 model,” this computational model can be employed as a QoE planning tool for use in estimating the effects on QoE due to variations in one or more quality parameters, including coding bitrate parameters, video frame rate parameters, packet

loss rate parameters, loudness parameters, echo parameters, video resolution parameters, etc. [1]. Specifically, the G.1070 model includes three distinct models, namely, a speech quality estimation model, a video quality estimation model, and a multimedia quality integration model.

The G.1070 video quality estimation model can be used to estimate the subjective effects on QoE of the quality parameters relating to video, such as the coding bitrate parameter, the video frame rate parameter, the packet loss rate parameter, the video resolution parameter, etc. For example, given assumptions about the coding bitrate, the frame rate, and the packet loss rate, the G.1070 video quality estimation model can be used to generate an estimate, typically in the form of a quality score, of the perceptual quality of the video that is delivered to the end user. Assuming a constant frame rate and a packet loss rate of zero, the G.1070 video quality estimation model typically produces higher quality scores for higher bitrates of compressed video information, and lower quality scores for lower bitrates of compressed video information. The original scope of g.1070 has been conceived for two-way video telephony applications only, but since its standardization, the G.1070 model has been widely used, studied, extended, and enhanced. Yamagishi and Hayashi [4] proposed to use G.1070 in the context of IPTV quality. Since the G.1070 model is codec dependent, Belmudez and Moller [5] extended the model, originally trained for H.264 and MPEG4 video, to MPEG-2 content. Joskowicz and Ardao [6] enhanced G.1070 with both resolution- and content-adaptive parameters.

Although the G.1070 model has been successfully employed as a QoE planning tool, the G.1070 model has drawbacks in that its conventional mode of operation is unsuitable for use as a QoE monitoring tool. We proposed a system for calculating the G.1070 visual quality estimate in a monitoring scenario in our previous work [2], which is illustrated in Figure 1. A Data Collector/Estimator is used to analyze the encoded bitstream, extract useful information, and estimate the bitrate, frame rate, and packet-loss rate. Details of the Data Collector/Estimator are not discussed in this paper. These three estimates are then provided as input to a

G.1070 Video Quality Estimator to yield the Video Quality Estimate. The G.1070 Video Quality Estimator implements the Video quality estimation function as defined in Section 11.2 of Rec. G.1070 [1].

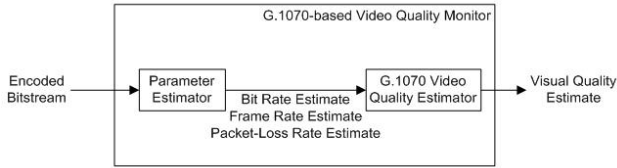


Fig. 1. A video quality monitoring system using the G.1070 video quality model.

Moreover, although the G.1070 model is generally suitable for estimating aspects of the perceptual quality of video that are related to the network, such as the expected packet loss rate, information about the content of the video is generally not considered. For example, a video scene with a complex background and a high level of motion, and another scene with relatively less activity or texture, may have dramatically different perceived qualities even if they are encoded at the same bitrate and frame rate. A video scene may include a relatively complex background with a high level of motion, whereas another video scene may include a relatively simple background with little or no motion. Each video frame of the second subsequent video scene may therefore be easier to predict from a reference video frame, and the coding bitrate required to achieve high quality coding of this scene may be relatively low. However, because the G.1070 video quality estimation model typically produces lower quality scores for lower bitrates of compressed video information, the G.1070 model may produce a relatively low quality score for the second subsequent video scene, notwithstanding the fact that the perceptual quality of that video scene may actually be high, perhaps even higher than the perceptual quality of the first video scene.

Another such video scene may be a very complex video scene, but the instantaneous coding bitrate required to represent the complex video scene with high quality may exceed the capabilities of the video channel and/or the video decoder. In that case, a bitrate control algorithm implemented in the video encoder may operate to limit the coding bitrate at a relatively high level, but not high enough to assure high quality coding of the complex video scene. Because the G.1070 video quality estimation model typically produces higher quality scores for higher bitrates of compressed video information, it may produce a relatively high quality score for the complex video scene, even though the perceptual quality of that video scene may actually be low.

Accordingly, in certain cases, the G.1070 video quality estimation model may either underestimate or overestimate the perceptual quality of video scenes, disadvantageously pro-

ducing quality scores that may not correlate well with subjective quality scores of the end user. To address this issue, this paper proposes a modified G.1070 model that takes frame complexity into consideration.

This paper is organized as follows. Section 2 describes the algorithm which estimates the video scene complexity and uses the complexity to normalize the bitrates for G.1070. Section 3 illustrates the simulation results and shows improvements in video quality estimation when the proposed complexity-normalization approach is employed in G.1070. Section 4 concludes the paper.

2. BITRATE NORMALIZATION USING FRAME COMPLEXITY

2.1. General Frame Complexity Estimation

The complexity of a frame is a combination of the spatial complexity of the picture and the temporal complexity of the scene in which it is found. Pictures with more detail have higher spatial complexity than those with little detail. Scenes with high motion have higher temporal complexity than those with little or no motion. In a general video compression process, for a fixed level of quantization, frames with a higher complexity yield more bits. Similarly, for a fixed target number of bits, frames with higher complexity result in larger quantization step sizes. Therefore, the coding complexity can be estimated based on the number of coded bits and the level of quantization. In this paper, the number of bits used and level of quantization are used to estimate the number of bits that would have been used at a reference quantization level. In the following we refer to the matrix of actual quantization step sizes as $M_{Q.input}$ and the matrix of reference quantization step sizes as $M_{Q.ref}$.

The following derivation applies to many video compression standards including MPEG2, MPEG4, and H.264/AVC. For a given frame, the number of bits that would have been used at the reference quantization level, denoted by $bits(M_{Q.ref})$, can be estimated by the actual bits used to encoding this frame, denoted by $bits(M_{Q.input})$, and the quantization matrices as shown in Equation (1). The quantization step size matrices M are either 8×8 or 4×4 depending on the specific video compression standard. Thus, each quantization step size matrix has either 64 or 16 entries. In Equation (1), the number of entries in the quantization step size matrix is denoted by N .

$$bits(M_{Q.ref}) \approx \frac{\sum_{i=0}^{N-1} a_i \times m_{Q.input.i}}{\sum_{i=0}^{N-1} a_i \times m_{Q.ref.i}} \times bits(M_{Q.input}) \quad (1)$$

The reference quantization step size matrix M_Q is arranged in zigzag order and m_Q is an entry in the matrix. To evaluate the effects of the quantization step size matrix, we consider a weighted sum of all the elements m_Q where

the averaging factor, a , for each element depends on the corresponding frequency. In natural imagery, the energy tends to be concentrated in the lower frequencies. Thus the weighted sums in Equation (1) allow the lower frequencies to be weighted more heavily than the higher frequencies.

In many cases, different macroblocks can have different quantization step size matrices. Thus, the matrices specified in Equation (1) are averaged over all the macroblocks in the frame. Some compression standards allow macroblocks to be skipped. This usually occurs when the macroblock data can be well predicted from previously coded data. So, to be more specific, the quantization step size matrices specified in Equation (1) are averaged over all the coded (not skipped) macroblocks in the frame.

Equation (1) can be simplified by considering only binary averaging factors, a . The average factors associated with low frequency coefficients are assigned a value of 1 and the average factors associated with high frequency coefficients are assigned a value of 0. Since the coefficients are stored in zig zag order, which is roughly ordered from low frequency to high, Equation (1) can be rewritten as Equation (2):

$$\text{bits}(M_{Q.ref}) \approx \frac{\sum_{i=0}^{K-1} a_i \times m_{Q.input.i}}{\sum_{i=0}^{K-1} a_i \times m_{Q.ref.i}} \times \text{bits}(M_{Q.input}) \quad (2)$$

We have found that for matrices that are 8×8 , the first 16 entries represent low frequencies and thus we set $K = 16$. For 4×4 matrices, the first 8 entries represent low frequencies and thus we set $K = 8$. For convenience of notation, assuming a fixed reference quantization matrix $Q.ref$ and using Q to represent $Q.input$, a quantization complexity factor, $fn(Q)$ can be defined as:

$$fn(Q) = \frac{\sum_{i=0}^{K-1} m_{Q.i}}{\sum_{i=0}^{K-1} m_{Q.ref.i}} \quad (3)$$

Then Equation (2) can be rewritten as:

$$\text{bits}(M_{Q.ref}) \approx fn(Q) \times \text{bits}(M_Q) \quad (4)$$

Finally, in order to derive a measure of frame complexity that is resolution independent, we normalize the estimate of the number of bits necessary at the reference quantization level by the number of 16×16 macroblocks in the frame ($frame_num_MB$). This gives the hypothetical number of bits per macroblock at the reference quantization level.

$$\begin{aligned} frame_compeity &= \frac{\text{bits}(M_{Q.ref})}{frame_num_MB} \\ &\approx \frac{fn(Q) \times \text{bits}(M_Q)}{frame_num_MB} \end{aligned} \quad (5)$$

The frame complexity estimation is designed for all video compression standards. Different video standards use different quantization step size matrices and, in the following text,

we derive the frame complexity functions for H.264/AVC, MPEG2, and MPEG4. In MPEG4, one of two quantization modes can be used. Since mode 0 is similar to MPEG2 and mode 1 is similar to H.264, here we only describe the frame complexity estimation for H.264 and MPEG2 in the details that follow.

2.2. H.264 Frame Complexity Estimation

H.264 (also known as MPEG4 Advanced Video Coding or AVC) uses a quantization parameter (QP) to determine the quantization level. The QP can take one of 52 values [7]. The QP is used to derive the quantization step size, which in turn is combined with a scaling matrix to derive the quantization step size matrix. An increase of 1 in QP results in a corresponding increase in quantization step size of approximately 12%. This change in quantization step size results in a corresponding increase in quantization step size matrix of a factor of approximately 1.1 and a decrease in the number of frame bits by a factor of $\frac{1}{1.1}$. Similarly a decrease of 1 in QP results in a decrease of approximately 12% in quantization step size, a decrease by a factor of approximately $\frac{1}{1.1}$ in quantization step size matrix, and an increase by a factor of 1.1 in the number of frame bits.

When calculating the quantization complexity factor, $fn(Q)$, for H.264, the reference QP used is 26 (the midpoint of possible QP values) to represent average quality. This factor, defined in Equation (3), is shown specifically for H.264 in Equation (6). The denominator, the reference quantization step size matrix, is that obtained using a QP of 26 and the numerator, is the average of all the quantization step size matrices in the frame. If the average QP in the frame is 26, then the ratio becomes unity. If the average QP in the frame is 27, then the ratio is 1.1, an increase by a factor of 1.1 from unity. Each increase in QP by 1 increases the ratio by another factor of 1.1. Thus, the ratio can be written with the power function shown on the right hand side of Equation (6). The frame complexity can then be calculated with Equation (5).

$$\begin{aligned} fn(Q) &= \frac{\sum_{i=0}^7 m_{frame.QP.avg.i}}{\sum_{i=0}^7 m_{QP26.i}} \\ &= 1.1^{(frame.QP.avg-26)} \end{aligned} \quad (6)$$

2.3. MPEG2 Frame Complexity Estimation

In MPEG2, the parameters $quant_scale_code$ and q_scale_type specify the quantization level [8]. The $quant_scale_code$ specifies a $quant_scale$ which is further weighted by a weighting matrix, W , to obtain the quantization stepsize matrix (Equation (7)). The mapping of $quant_scale_code$ to $quantizer_scale$ can be linear or non-linear as specified by the q_scale_type .

$$M = quant_scale \times W \quad (7)$$

MPEG2 uses an 8×8 DCT transform and the quantization step-size matrix is 8×8 , resulting 64 quantization step-sizes for 64 coefficients after DCT transform. As had been pointed out previously, the low frequency coefficients contribute more to the total coded bits. In Equation (2), we set $K = 16$, and the average factors associated with the first 16 low frequency coefficients are assigned a value of 1 and the average factors associated with the high frequency coefficients are assigned a value of 0. Therefore, Equation (3) becomes:

$$\begin{aligned} fn(Q) &= \frac{\sum_{i=0}^{15} m_{Q,input,i}}{\sum_{i=0}^{15} m_{Q,ref,i}} \\ &= \frac{\sum_{i=0}^{15} w_{input,i} \times quant_scale_{input,i}}{\sum_{i=0}^{15} w_{ref,i} \times quant_scale_{ref,i}} \end{aligned} \quad (8)$$

In MPEG2, the *quant_scale_code* has one value for each macroblock, the value of which is between 1 and 31. The *quant_scale_code* is the same at each coefficient position in the 8×8 matrix. Thus, the *quant_scale_input* and *quant_scale_ref*, in Equation (8), are independent of i and can be factored out of the summation. For the reference, we choose 16 as the reference *quant_scale_code* to represent the average quantization. We use the notation *quant_scale* [16] to indicate the value of *quant_scale* when the *quant_scale_code* = 16. For the input bit-stream, we calculate the average *quant_scale_code* for each frame over the coded macroblocks, and we denote it as *quant_scale_input_avg*.

The weighting matrix, W , used for intra-coded blocks is typically different from that used for non-intra blocks. Default weighting matrices are defined in the standard, however the MPEG2 encoder can define and send its own weighting matrix rather than use the defaults. For example, the MPEG2 encoder developed by the MPEG Software Simulation Group (MSSG) uses the default weighting matrix for intra-coded blocks and provides a non-default weighting matrix for non-intra blocks [9]. In the denominator of Equation (9), we use the MSSG weighting matrices as the reference.

$$fn(Q) = \frac{quant_scale_{input_avg} \times \sum_{i=0}^{15} w_{input,i}}{quant_scale [16] \times \sum_{i=0}^{15} w_{ref,i}} \quad (9)$$

To simplify, *quant_scale* [16] = 32 for linear mapping and *quant_scale* [16] = 24 for non-linear mapping. Also, the sum of the first 16 MSSG weighting matrix components for non-intra coded blocks is 301 and that for intra-coded blocks is 329. Thus, the denominator in Equation (9) is a constant and $fn(Q)$ can be rewritten as:

$$fn(Q) = \frac{1}{fnD} \left(quant_scale_{input_avg} \times \sum_{i=0}^{15} w_{input,i} \right) \quad (10)$$

where

$$fnD = \begin{cases} 32 * 301 = 9632 & linear, non - intra \\ 24 * 301 = 7224 & non - linear, non - intra \\ 32 * 329 = 10528 & linear, intra \\ 24 * 329 = 7896 & non - linear, intra \end{cases} \quad (11)$$

The frame complexity can then be calculated with Equation (5).

2.4. Bitrate normalization using frame complexity

As discussed earlier, the bitrate estimate is normalized by the calculated frame complexity to provide an input to G.1070 that will yield measurements better correlated to subjective scores. Because the frame bits are used in the frame complexity estimation, it can be seen that normalization will cause the bitrate to be cancelled and only the quantization stepsize matrix remains. To prevent the cancelling of the bitrates and maintain some consistency with the current G.1070 function inputs (bitrate, frame rate, and packet-loss rate), the normalization process is revised. Another observation is that as the bitrate decreases, fewer macroblocks are coded (more macroblocks are skipped). The percentage of macroblocks that are coded can be used to represent the bitrate in the complexity estimation equations.

$$\begin{aligned} bitrate_norm &= \frac{bitrate}{frame_complexity} \\ &= \frac{bitrate}{\left(\frac{num_coded_MB}{frame_num_MB} \right) \times fn(Q)} \end{aligned} \quad (12)$$

Comparing to the system illustrated in Figure 1, the enhanced system (G.1070E) that use frame complexity to normalize estimated bitrate will be as shown in Figure 2. For a given frame, the Data Collector/Estimator is modified to also extract the quantization stepsize matrix, the number of the coded macroblocks, and the number of coded bits for this frame. This information is used by the Frame complexity Estimator which computes an estimate of the frame complexity. The frame complexity estimate is then used by the Bitrate Normalizer to normalize the bitrate. Finally, the frame rate estimate and packet loss rate estimate from the Data Collector/Estimator as well as the normalized bitrate from the bitrate Normalizer are used by the G.1070 Video Quality Estimator to yield the video quality estimate.

3. SIMULATION RESULTS

In this section, we present the results of the experiments with several testing datasets and the results of comparing the G.1070 and the proposed G.1070E. We have done testing with H.264 and MPEG2 compressed bitstreams. They have a similar performance. Due to the length limitation of this paper, we only present testing results with H.264 compression.

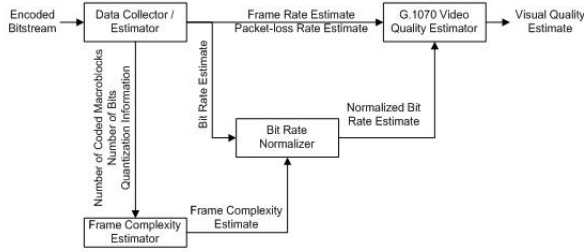
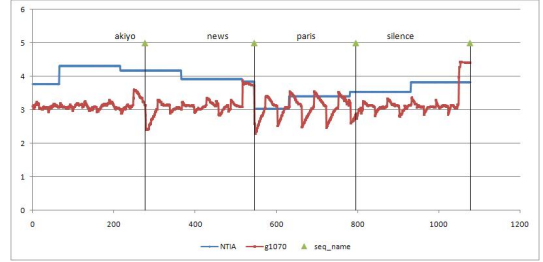


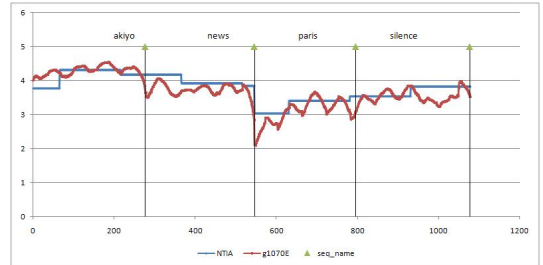
Fig. 2. An extension of the G.1070 video quality model to include bitrate normalization based on an analysis of frame complexity.

To illustrate how the proposed G.1070E measures the video quality by considering the underlying video content complexity, we use a concatenated CIF sequence “scene-change,” which includes 18 video contents (coastguard, bus, foreman, akiyo, news, flower garden, hall, football, waterfall, tempete, paris, silence, mobile calendar, bridge, mother and daughter, Stefan, table tennis and container) and covers various video contents that are easy, medium and tough [10]. Each sequence has 300-400 frames. The scene-change sequence has been coded at constant bitrate (CBR) 192kbps. The original G.1070 model measures the quality only based on the bitrate, but the easy contents (for example, akiyo, news, paris, silence) will have better visual quality with similar bitrates compared to the tough contents, like football. The proposed G.1070E model takes the video content into consideration by normalizing the bitrates using the frame complexity. It reflects the subjective quality more accurately than the standard G.1070 model. To better illustrate the comparison between the G.1070 and G.1070E, only parts of the scene-change bitstream are shown in Figure 3 and a NTIA-VQM score [3], which is generated by a model which uses the original video as reference and is one of the best-performing video quality metrics (VQM) available today, is generated for each 5 second segment from the source video and used as a reference VQM score in Figure 3.

VQEG [11] has a video database library with MOS scores. The videos in VQEG database are in YUV format. It has 22 reference videos and 16 processed HRCs (Hypothetical Reference Condition) for each of the reference video. However, our model is based on features extracted from compressed bitstream. Therefore, we cannot use their HRCs and hence the MOS scores directly. We take the 22 reference videos and from each reference source video content, a high-quality bitstream is created using a fairly low QP (QP 16) and high framerate (25 or 30 fps). For each high-quality bitstream, 18 bitstreams are created by transcoding the high-quality bitstream to a different resolution (QCIF, CIF or VGA), bitrates (64kbps to 512 kbps), QP’s (20 to 40), frame rates (6 to 25/30 fps). Since it is infeasible to collect subjective MOS values for such a large dataset, we chose instead to use the MOS



(a) G.1070 VQM performance



(b) G.1070E VQM performance

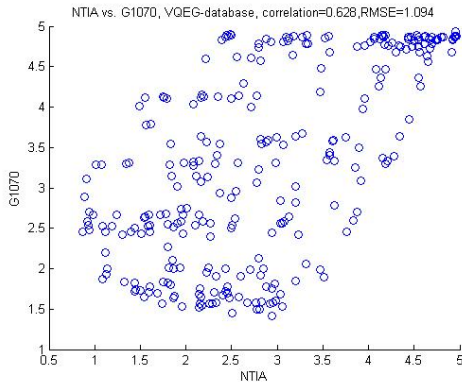
Fig. 3. The comparison between G.1070E and G.1070 for a concatenated scene change sequence.

values predicted by NTIA-VQM [3]. The testing videos are divided into non-overlapping 8-second “test” segments. A NTIA-VQM score is generated for each test segment from the testing bitstreams in comparison to the corresponding segment from the source video. These values lie in the range [1, 5] where 5 (Qmax) represents the highest quality (i.e. as good as the source). The Pearson correlation and the root mean squared error (RMSE) between the NTIA-VQM and G.1070 or G.1070E VQM scores are compared. Figure 4 shows the test results using the 280 test bitstreams. The scatter plots show that G.1070E has higher correlation and lower RMSE than G.1070.

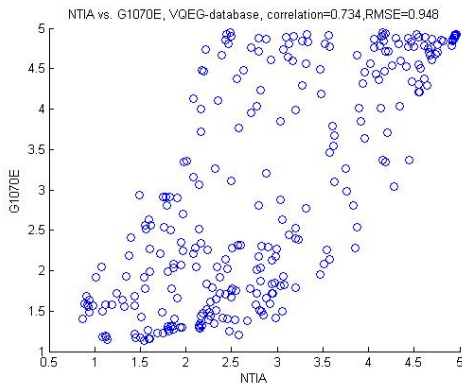
The third experiment was conducted using the Image Group of Instituto de Telecomunicações, Instituto Superior Técnico (IT-IST) dataset [12], which provides the MOS scores. Clearly, the proposed G.1070E outperforms G.1070, as illustrated in Figure 5.

4. CONCLUSION

This paper proposed a no-reference compressed bitstream domain based video objective quality measurement model. This model is an extension of the existing ITU G.1070 model. A complexity measurement is taken into consideration to normalize the bitrate estimation of the G.1070 model. Experimental result shows that the enhanced G.1070 model, G.1070E, has a much higher correlation with subjective MOS scores and can reflect the quality of video experience much better than G.1070.

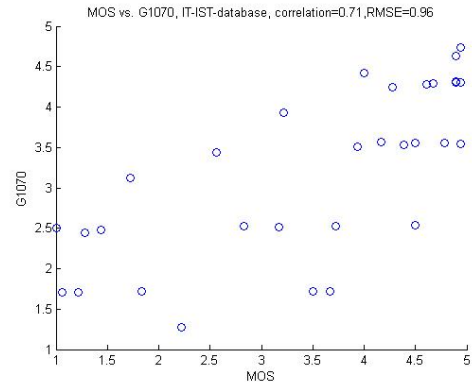


(a) G.1070 VQM performance

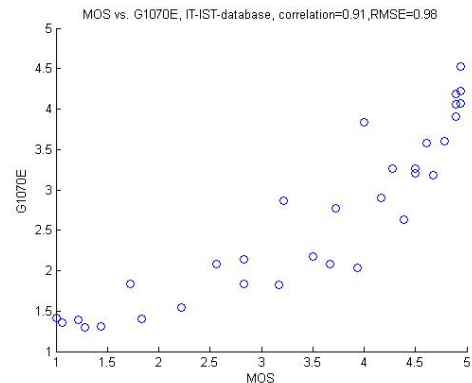


(b) G.1070E VQM performance

Fig. 4. The comparison between G.1070E and G.1070 for the VQEG video sequences.



(a) G.1070 VQM performance



(b) G.1070E VQM performance

Fig. 5. The comparison between G.1070E and G.1070 for the IT-IST H.264 encoded sequences.

5. REFERENCES

- [1] International Telecommunication Union, *ITU-T G.1070*, April 2007.
- [2] N. Narvekar, T. Liu, D. Zou, and J. Bloom, "Extending G.1070 for video quality monitoring," *IEEE International Conference on Multimedia and Expo (ICME)*, submitted, 2011.
- [3] S. Wolf and M. Pinson, "Video quality measurement techniques," *National Telecommunications and Information Administration (NTIA) Report*, June 2002.
- [4] K. Yamagishi and T. Hayashi, "Parametric packet-layer model for monitoring video quality of IPTV services," in *IEEE International Conference on Communications*, May 2008.
- [5] B. Belmudez and S. Moller, "Extension of the G.1070 video quality function for the MPEG2 video codec," in *International Workshop on Quality of Multimedia Experience (QoMEX)*, 2010.
- [6] J. Joskowicz and J. Ardao, "Enhancements to the opinion model for video-telephony applications," in *Fifth International Latin American Networking Conference*, 2009.
- [7] T. Wiegand, G. Bjntegaard, G.J. Sullivan, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, July 2003.
- [8] *ISO/IEC 13818-2 MPEG2*, 1995.
- [9] MPEG-2 video decoder ver. 12, Available online at <http://www.mpeg.org/MPEG/MSSG>.
- [10] Standard Video Sequences, Available online at <http://trace.eas.asu.edu/yuv>.
- [11] VQEG Video Sequences, Available online at <http://www.its.bldrdoc.gov/vqeg/>.
- [12] Instituto Superior Técnico of Instituto de Telecomunicações dataset, Available online at <http://amalia.img.lx.it.pt>.