Compression of VQM Features for Low Bit-Rate Video Quality Monitoring

Mina Makar, Yao-Chung Lin, Andre F. de Araujo and Bernd Girod

Information Systems Laboratory, Stanford University, Stanford, CA 94305

{mamakar,yclin79,afaraujo,bgirod}@stanford.edu

***Abstract*—Reduced reference video quality assessment techniques provide a practical and convenient way of evaluating the quality of a processed video. In this paper, we propose a method to efficiently compress standardized VQM (Video Quality Model) [1] features to bit-rates that are small relative to the transmitted video. This is achieved through two stages of compression. In the first stage, we remove the redundancy in the features by only transmitting the necessary original video features at the lowest acceptable resolution for the calculation of the final VQM value. The second stage involves using the features of the processed video at the receiver as side-information for efficient entropy coding and reconstruction of the original video features. Experimental results demonstrate that our approach achieves high compression ratios of more than $30\times$ with small error in the final VQM values.**

I. Introduction

Video quality monitoring is becoming a crucial part of modern video transmission systems. Reduced reference (RR) video quality assessment techniques [2] are gaining more and more interest since they enable the judgment of the perceptual quality of a processed video sequence by comparing certain features calculated from the original and the corresponding processed videos. Hence, they eliminate the need of the availability of the original video for quality assessment.

In order to possess practical significance, the bit-rate for sending the RR quality features should be small when compared to the bit-rate of the transmitted video. Since PSNR is the most widely used measure of video quality, prior work [3]–[5] proposed sending compressed features of the original video and using them to obtain an acceptable estimate of the PSNR of the received processed video. Similar features can be obtained at the receiver side from the processed video. Hence, authors in [4]–[6] suggest using the processed video features as side-information for efficient compression of the original video features through Distributed Source Coding (DSC) [7], and efficient decoding of these compressed features through Minimum Mean Squared Error (MMSE) reconstruction [6].

The NTIA general Video Quality Model (VQM) [1] has been selected by both ANSI [8] and ITU [9] as a video quality assessment standard based on its performance [1]. Unfortunately, this general model requires a bit-rate of several Mbps (more than 4 Mbps for 30 fps, CIF size video) of quality features for the calculation of the VQM value, which prevents this model from being applied in a practical system. A low-bandwidth VQM is proposed by the same authors [10] in an attempt to reduce the bit-rate required for the transmission of VQM features.

In this paper, we study the problem of compression of VQM features and propose a method for efficient compression of these features that can be applied to any VQM model defined in [1], [10], [11]. The proposed compression is achieved through two stages. First, we remove the redundancy in the VQM features by only transmitting the necessary features at the lowest acceptable resolution for the calculation of the final VQM value. The second stage is inspired by [5], [6] and it involves efficient entropy coding of the original video features using DSC and dequantization using MMSE reconstruction.

The remainder of the paper is organized as follows. Section II presents a review of the VQM standard explaining how different VQM features and model parameters are calculated. In Section III, we discuss the proposed technique for compression of VQM features, including feature selection, subsampling, quantization and efficient entropy coding using DSC. Finally, in Section IV, we present experimental results showing the performance of the proposed compression method in terms of achieving high compression ratios with only small error in the VQM parameters and final values.

II. Review of VQM Standard

In [1], the authors refer to the video transmission system under test as the Hypothetical Reference Circuit (HRC). The VQM value is a number between 0 and 1 that is used to judge the visual quality of the processed video after passing through the HRC, where lower values indicate better quality. To calculate a VQM value, VQM features are calculated from the original video and sent over the video transmission system through an ancillary data channel. On the receiver side, the same VQM features are calculated from the processed video. VQM features from the original and processed videos are compared to provide VQM model parameters. These parameters are linearly combined to obtain the final VQM value. Details of VQM calculation steps are described as follows.

A VQM quality *feature* is defined [1], as 'a quantity of information associated with, or extracted from, a 3D block of a video stream (either original or processed)'. The 3D blocks used for feature calculation are referred to as spatial-temporal (S-T) regions. For example, for a 30 fps video, an S-T region

MMSP'11, October 17-19, 2011, Hangzhou, China.

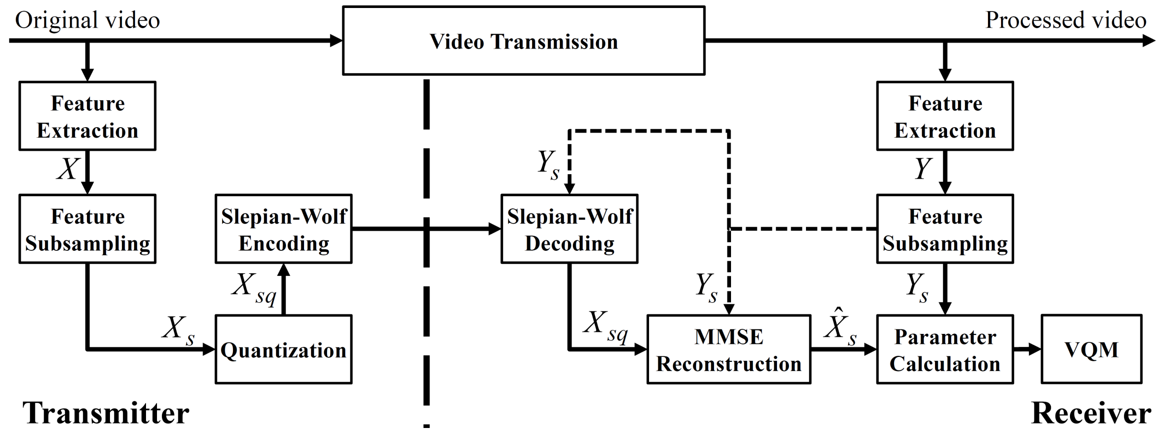

Fig. 1. Block diagram of the proposed VQM features compression technique. On the transmitter side, VQM features are extracted from the original video, subsampled, quantized, encoded using DSC and transmitted over an ancillary data channel. On the receiver side, corresponding features are extracted from the processed video, subsampled and used as side-information (dotted lines) in Slepian-Wolf decoding and MMSE reconstruction of the original video features. Finally, features from original and processed videos and used in the calculation of VQM model parameters and VQM final value.

of dimensions ($8 \times 8 \times 0.2$ sec.) means that each ($8 \times 8 \times 6$) 3D block of pixels is used to calculate one feature value.

VQM features are either based on spatial gradients, chrominance information, contrast information or absolute temporal information (ATI). Four different features are defined in the general VQM [1]: a) f_{SI} which is sensitive to changes in the spatial activity like blurring and noise, b) f_{HV} which is sensitive to changes in the orientation of spatial activity, c) f_{COHER_COLOR} which concatenates two features f_{Cb} and f_{Cr} calculated from Cb and Cr color components respectively, and finally d) f_{CONT_ATI} defined as the product of f_{CONT} feature measuring the contrast information and f_{ATI} feature measuring the absolute temporal information. Different perceptibility thresholds are used during the calculation of f_{SI}, f_{HV} and f_{CONT_ATI} features.

Low-bandwidth VQM [10] uses similar f_{SI}, f_{HV} and f_{COHER_COLOR} features although these features are calculated on bigger S-T regions of size ($32 \times 32 \times 1$ sec.). Instead of f_{CONT_ATI}, low-bandwidth VQM utilizes a simpler feature f_{ATI_rms} that is calculated over the whole video frames and not S-T regions to describe the temporal information. To reduce the bit-rate needed for sending the features, low-bandwidth model uses non-uniform quantizers ranging from 8 to 10 bits to quantize different VQM features.

Note that for all VQM models, a calibration step [11] between the original and processed videos is performed before calculating the features. In this paper, we do not require this calibration step since we use original and processed videos which are aligned spatially and temporally and have no gain and level offset.

To calculate a VQM model *parameter*, features from the original and processed videos are compared using a suitable comparison function. Some comparison functions treat the resulting positive and negative values separately since they produce different effects on quality perception. For example, an increase in the contrast of the processed video may be due to blocking artifacts, while a decrease may indicate a blurring effect. Comparison functions are non-linear including error ratio, logarithmic ratio and Euclidean distance. Error pooling along spatial and temporal directions using non-linear collapsing functions is performed on the result of the comparison function to result in a single parameter value. Finally, an optional step of non-linear scaling and/or clipping of the parameter value may be performed.

A final VQM value is calculated as a linear combination of VQM model parameters. Linear weights are chosen through subjective tests to better match mean opinion score (MOS) results. VQM models differ in the comparison functions and the spatial and temporal collapsing functions used to calculate the model parameters and the linear weights used to combine these parameters. For more details on the calculation of different VQM models, the reader is referred to [1], [10], [11]. In this paper, we focus on general VQM and low-bandwidth VQM, although the proposed techniques can be directly applied to any other VQM model.

III. Compression of VQM Features

Our proposed compression method of VQM features exploits three main observations. First, VQM features measure different properties of the video. Thus, depending on the HRC under test, some features may be unnecessary, since the degradations they measure are not introduced by this HRC. Second, during the quantization of the original video features, we make use of the fact that our final goal is to minimize the distortion in the VQM model parameter values, even if we introduce high distortion in the compressed VQM features. And third, the high correlation between the original video features and the processed video features suggests the use of the processed video features as side-information for efficient entropy coding and better decoding of the quantized features of the original video. Fig. 1 presents a block diagram of the proposed technique and details are described in the following subsections.

A. Feature Selection and Feature Subsampling

We refer to the MSE between a VQM feature of the original video and the compressed version of this feature as VQM *feature distortion*. Similarly, we refer to the squared value of the difference between the VQM model parameter calculated using the original video feature and the same model parameter calculated using the compressed version of the original video feature as VQM *parameter distortion*. Our goal is to minimize parameter distortions in order to preserve the final VQM value. Note that when calculating parameter distortions, the parameter values are always weighted by the weight used in the calculation of the final VQM value. This ensures that we take into account the relative importance of the parameters in terms of the distortion they induce in the VQM value.

As mentioned in Section II, VQM features measure different properties of the video sequence. Thus, depending on the HRC under test, some features may be unnecessary. For example, if we consider an HRC where we only observe blurriness or blocking artifacts but not motion artifacts, we obtain negligible values for the VQM parameters calculated from f_{CONT_ATI}.

To decide which VQM features to transmit, we define a decision threshold T_{select}. If the squared value of the VQM parameter resulting from a certain feature is always less than T_{select} for a certain HRC, this feature is not transmitted, and we assume a zero value for the corresponding parameter when we calculate the final VQM value. Otherwise, this feature is transmitted, and we go to the next step of feature subsampling.

The size of S-T regions for different VQM features is designed to compromise between the amount of data sent for each feature and the accuracy of the feature in detecting localized artifacts. In order to reduce the number of samples for each feature and still keep the VQM calculations standard compliant, we decide not to change the size of S-T regions but subsample the VQM features instead, with a spatial subsampling ratio r_S and a temporal subsampling ratio r_T.

In spatial subsampling, we keep every r_S^{th} row and column in the feature data, shifting the subsampling grid at each time interval defining the temporal extent of S-T regions. This results in data reduction to $1/r_S^2$ of the original size. In temporal subsampling, we keep the features for S-T regions calculated every r_T time intervals. This results in data reduction to $1/r_T$ of the original size. Hence, the subsampled feature data size is $1/r_S^2 r_T$ of the original size. To decide the values of r_S and r_T for a certain feature, we define a threshold $T_{\text{subsample}}$, try all possible combinations of r_S and r_T and choose the values that result in the largest data reduction, while the distortion in the corresponding VQM parameter is below $T_{\text{subsample}}$. The subsampling operation is shown in Fig. 1. We refer to the original video feature as X, the processed video feature as Y and their subsampled versions as X_s and Y_s respectively.

In practice, the decision of the subsampling ratios r_S and r_T for each feature is performed using a training set of different processed videos that represent the HRC under test. As a conservative choice, the smallest spatial and temporal subsampling ratios for each feature over the whole training set is applied in the actual video transmission.

In the case of changing transmission system characteristics, another option is to perform an adaptive choice of subsampling ratios. This is achieved by first transmitting VQM features at their full resolution. The receiver calculates VQM parameters based on the full resolution features, and recalculates the same parameters based on different subsampled versions of the same received features. The receiver decides the subsampling ratios and communicates them back to the transmitter.

B. Feature Quantization and MMSE Reconstruction

To further compress the data of the original video features, we quantize the subsampled original video features X_s as shown in Fig. 1. We use a uniform mid-tread quantizer with M bits (2^M levels). The determination of the value of M for each feature ensures the optimal rate allocation among features and is performed in a Lagrangian optimization framework. For each feature, we try different values of M and for each value, we calculate the cost function $J = D + \lambda R$; where D is the distortion in the parameter value(s) calculated from this feature as defined above and R is the bit-rate spent on sending the subsampled and quantized original video feature X_{sq}. We fix the value of λ and choose the value of M that minimizes the cost function J for each VQM feature.

For the features that use a perceptibility threshold, there is often a peak in the distribution of the feature values at this threshold. It is beneficial to have a representative level at this peak. This is achieved by first subtracting the perceptibility threshold from all the feature values so the peak in the distribution is translated to zero and then using the same uniform quantizer as before. The perceptibility threshold is added back after feature decoding.

At the receiver side, we use the features of the processed video Y_s as side-information to obtain a better estimate of the original video features $\hat{X}_s$. As shown in Fig. 1, the original video feature is decoded as the MMSE reconstruction given the quantized feature X_{sq} and the processed video feature Y_s. The quantized feature value X_{sq} defines the interval where the original feature value lies. If this interval is bounded by X_{LB} and X_{UB}; then the MMSE reconstruction $\hat{X}_s$ is given by

$$\hat{X}_s = E\left[X_s | X_{sq}, Y_s\right] = \frac{\displaystyle\int_{X_{LB}}^{X_{UB}} x_s f_{X_s|Y_s}(x_s|y_s)dx_s}{\displaystyle\int_{X_{LB}}^{X_{UB}} f_{X_s|Y_s}(x_s|y_s)dx_s} \tag{1}$$

We model $f_{X_s|Y_s}(x_s|y_s)$ as a normal distribution $\mathcal{N}\left(y_s, \sigma^2\right)$ with mean y_s and variance σ^2. This distribution represents how correlated X_s and Y_s are. σ^2 is unknown to the receiver; however, it can be estimated using Maximum Likelihood (ML) estimation from N samples of X_{sq} and Y_s. The ML estimate of σ^2 is given by [5]

$$\sigma^2 = \hat{\sigma^2}_{\text{ML}} = \frac{1}{N}\sum_{i=1}^{N} E\left[\left(X_s - Y_s\left(i\right)\right)^2 | X_{sq}\left(i\right), Y_s\left(i\right)\right] \tag{2}$$

C. Entropy Coding Using DSC

Since the processed video features Y_s are highly correlated to X_s and are already available at the decoder, we can use DSC to efficiently encode X_{sq} at a small bit-rate, and then use Y_s as side-information to decode the Slepian-Wolf coded X_{sq}. Each bitplane in X_{sq} is coded at the Slepian-Wolf encoder using *rate-adaptive LDPC* codes [12] as in [5]; where the choice of the coding bit-rate depends on the worst allowable processed videos that can result from the HRC under test. The block diagram in Fig. 1 shows how DSC is incorporated into the whole system.

D. Compression of Low-Bandwidth Model Features

Low-bandwidth VQM requires a much lower bit-rate than general VQM because of larger S-T regions and quantization of the feature values. In [10], the authors report that they can send VQM features at 10 kbps for 30 fps, 672×384 video. Our own experiments indicate that reducing the original features bit-rate makes feature values very sensitive to any subsampling or further quantization as described in Sections III-A and III-B. We obtain unacceptably large changes in the corresponding parameter values when we subsample or quantize low-bandwidth VQM features. Therefore, we apply lossless compression to low-bandwidth VQM. This is achieved through directly applying DSC on the original VQM features X. This allows us to retain the exact VQM value and still obtain high compression ratios as will be shown in Section IV.

IV. Experimental Results

We perform experiments with seven CIF size standard video sequences, *Foreman*, *Football*, *Mother and Daughter*, *Mobile*, *Carphone*, *Stefan* and *Tempete*. We consider two types of HRCs. The first HRC (referred to as HRC 1) has only compression artifacts and no network losses. The videos are encoded in H.264 standard with quantization parameters (QPs) ranging from 22 to 38 which results in different video qualities and hence different VQM model parameters. The second HRC (referred to as HRC 2) considers network losses. The videos are compressed at high quality (QP = 26) and then transmitted through an error-prone network where packets are dropped randomly at packet loss rate (PLR) ranging from 1% to 16%. The decoder performs frame copy error concealment to account for network losses. We use JM Reference Software, version 16.1 [13] for video coding and BVQM Software, version 1.3 [14] for calculating VQM features and model parameters. Section IV-A discusses the results for general VQM model and Section IV-B discusses the results for low-bandwidth VQM model.

A. General VQM Results

For general VQM experiments, the video sequences are divided into a training set containing the first four videos mentioned above and a test set containing *Carphone*, *Stefan* and *Tempete* sequences. Based on the training set, our experiment on feature selection indicates that for HRC 1 with compression artifacts only and for HRC 2 with random but stationary packet losses, there are not many artifacts in the temporal direction which results in negligible value of the parameter calculated from f_{CONT_ATI}. We use $T_{\text{select}} = 10^{-5}$ and we find that f_{CONT_ATI} is not selected in both HRCs. Based on the S-T region sizes defined in the general VQM standard, choosing not to transmit f_{CONT_ATI} results in 22% reduction in the total VQM features data.

The experiment on feature subsampling uses the first non-adaptive methodology described in Section III-A on our training set. The goal is to choose suitable subsampling ratios r_S and r_T for each VQM feature. We use $T_{\text{subsample}} = 10^{-5}$. As expected, the smallest spatial and temporal subsampling ratios usually result from the worst quality processed videos in each HRC. Table I presents the subsampling ratios that we obtain for different features. We find that f_{HV} is most sensitive to subsampling. The total reduction in the size of VQM features data after feature selection and feature subsampling is 86% for HRC 1 and 81.5% for HRC 2.

TABLE I
SUBSAMPLING RATIOS FOR GENERAL VQM

General VQM Feature	HRC 1		HRC 2	
	r_S	r_T	r_S	r_T
f_{SI}	3	1	1	1
f_{HV}	1	1	1	1
f_{COHER_COLOR}	2	2	2	2
f_{CONT_ATI}	not selected		not selected	

The next step in compressing VQM features is to quantize the subsampled features. To decide the number of bits M for quantizing each VQM feature in an optimal rate allocation framework, we perform the following experiment: All videos in the training set are transmitted through the same HRC with the same conditions (same QP for HRC 1 and same PLR for HRC 2). The subsampled VQM features are quantized with uniform quantizers with M ranging from 4 to 9 bits. For each value of M, we calculate the average bit-rate spent on each feature over the whole training set. We also calculate the average parameter distortion resulting from using the subsampled quantized features as compared to the unquantized features at full resolution. If a feature is used to calculate more than one parameter, the parameter distortions are added. Example results for average bit-rates and average parameter distortions at different values of M are shown in Fig. 2. These results represent HRC 1 at QP = 26. The curves show that f_{HV} is most sensitive to the feature quantization operation.

For all QPs in HRC 1 and PLRs in HRC 2, we fix a value of λ and calculate the cost function $J = D + \lambda R$; where D is the average parameter distortion and R is the average bit-rate calculated as mentioned before. We use a small value of λ to keep minimal parameter distortions and preserve the final VQM values. For each VQM feature, we detect the values of M that minimize J, and these values of M are averaged over different QPs in HRC 1 and different PLRs in HRC 2 to obtain a single value of M to be used for the quantization of this feature. Table II presents the number of bits M for the quantizers used in HRC 1 and HRC 2. We observe that

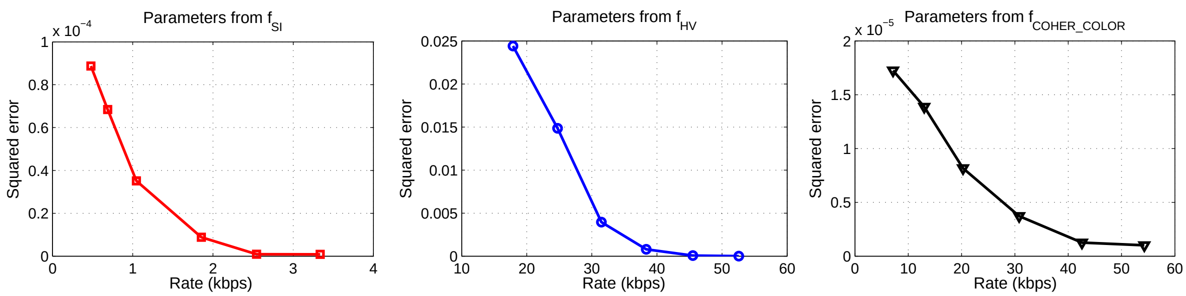

Fig. 2. Average bit-rates and parameter distortions for different VQM features (HRC 1, QP = 26). M varies from 4 to 9 bits.

TABLE II
NUMBER OF BITS M FOR VQM FEATURES QUANTIZATION

General VQM Feature	HRC 1	HRC 2
f_{SI}	8 bits	7 bits
f_{HV}	9 bits	9 bits
f_{COHER_COLOR}	5 bits	5 bits

f_{HV} needs fine quantization, while f_{COHER_COLOR} can be subjected to coarser quantization.

Finally, we use uniform quantizers with values of M indicated by Table II to quantize VQM features for all videos. In Fig. 3, we present the total minimum decodable bit-rates spent on encoding VQM features and the corresponding absolute error in the final VQM value. The figure indicates the effect of all the compression stages proposed in Section III. Figs. 3(a) and 3(b) show the results for HRC 1 and Figs. 3(c) and 3(d) show the results for HRC 2. Training set results represent worst case results in terms of the largest bit-rate and absolute error over all the videos in our training set.

As the processed video quality becomes worse (higher QP in HRC 1 and higher PLR in HRC 2), the bit-rate needed for sending VQM features increases because of lower correlation between original and processed video features, which makes DSC less efficient. Also, the absolute error in the final VQM value has an increasing trend since lower correlation between original and processed video features makes MMSE reconstruction less efficient as well. We observe that bit-rate results are comparable for the training and the test set. Absolute error results are slightly worse for the test set. However, absolute errors are always below 0.03 with negligible effect on the accuracy of the general VQM model.

According to the S-T region sizes defined for the general VQM [1], we need to transmit 1.41×10^5 feature values per sec. for 30 fps, CIF size video. If single precision floating point format (32 bits) is used for signaling these features, this means the uncompressed full resolution features are transmitted at 4.512 Mbps. Fig. 3 indicates that using our proposed technique, VQM features are compressed to bit-rates below 150 kbps. Thus, we can achieve a compression ratio of more than $30\times$ with only small error in the final VQM value.

B. Low-Bandwidth VQM Results

As mentioned in Section III-D, we perform lossless compression on VQM low-bandwidth model features to preserve the exact precision of the final VQM value. We compare the proposed technique of using DSC for feature compression against conventional coding techniques of fixed length coding and Huffman coding. Fig. 4 presents the total bit-rates needed for DSC compression comparing them to fixed length coding and Huffman coding for three video sequences: *Mother and Daughter*, *Mobile* and *Football* representing low, medium and high motion respectively. We find that the minimum decodable bit-rates for sending DSC-compressed low-bandwidth VQM features are between 1 and 3.5 kbps for 30 fps, CIF size video (compared to 4.65 kbps for fixed length coding). These bit-rates are very small compared to the rates used for encoding the video at an acceptable quality.

DSC has better performance than Huffman coding due to the efficient utilization of the processed video features in the compression of the original video features. Fig. 4(a) shows the results for HRC 1 and Fig. 4(b) shows the results for HRC 2. As QP and/or PLR increase, the processed video has more artifacts and its features are less correlated to the features of the original video, and hence the total rate of the DSC-compressed VQM features increases. We observe a large increase in the bit-rate for *Football* at high PLRs versus a small increase for *Mother and Daughter*. This is due to the high motion in *Football* that spreads the error propagation artifacts across the whole sequence after frame copy error concealment. This is not the case for *Mother and Daughter* where artifacts do not propagate as much because of low motion.

V. CONCLUSIONS

We present a method to compress VQM features to bit-rates that are affordable to current video transmission systems. This method relies on the use of the processed video features as side-information on the receiver side. For general VQM model, we manage to remove the redundancies in the feature data through feature selection and feature subsampling. We perform optimal bit-rate allocation between different VQM features and efficiently encode the compressed features using DSC and decode them using MMSE reconstruction. For low-bandwidth VQM model, we efficiently perform lossless compression of VQM features using DSC and compare the performance to conventional Huffman coding.

Experimental results show the efficiency of the proposed techniques in terms of achieving a total compression ratio of $30\times$ with small error in the general VQM model value.

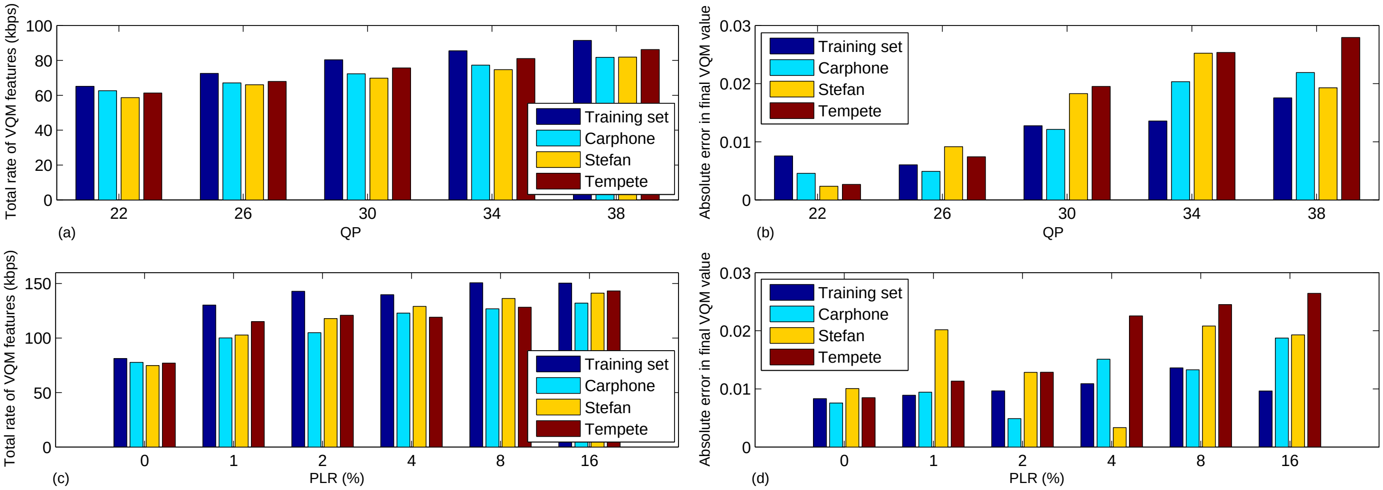

Fig. 3. Total minimum decodable bit-rates of VQM features and absolute errors in the final VQM value for different video sequences and HRCs. (a) HRC 1, total bit-rates, (b) HRC 1, VQM absolute errors, (c) HRC 2, total bit-rates and (d) HRC 2, VQM absolute errors.

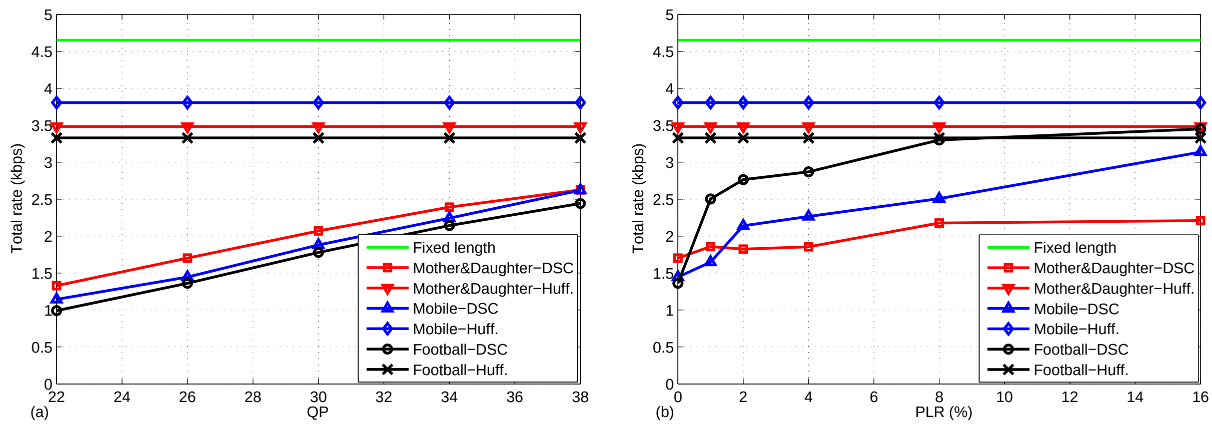

Fig. 4. Total bit-rates for DSC and Huffman coding of low-bandwidth VQM features. (a) HRC 1 and (b) HRC 2.

For the low-bandwidth model, we preserve the exact value of the final VQM with DSC compression that outperforms fixed length coding and Huffman coding. Our results indicate that we can send low-bandwidth VQM features at a negligible bit-rate compared to the rate needed for video transmission.

References

[1] M. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Trans. on Broadcasting*, vol.50, no.3, pp. 312-322, Sept. 2004.

[2] K. Seshadrinathan and A. C. Bovik, *The Essential Guide to Video Processing - Chapter 14: Video Quality Assessment*, 1st ed., Academic Press, May 2009.

[3] *Framework for remote monitoring of transmitted picture signal-to-noise ratio using spread-spectrum and orthogonal transform*, Std., ITU-T Recommendation J.240, 2004.

[4] K. Chono, Y.-C. Lin, D. Varodayan, Y. Miyamoto, and B. Girod, "Reduced-reference image quality estimation using distributed source coding," in *Proc. International Conference on Multimedia and Expo*, Hannover, Germany, June 2008.

[5] Y.-C. Lin, D. Varodayan, and B. Girod, "Video quality monitoring for mobile multicast peers using distributed source coding," in *Proc. 5th International Mobile Multimedia Communications Conference*, London, U.K., Sept. 2009.

[6] G. Valenzise, M. Naccari, M. Tagliasacchi, and S. Tubaro, "Reduced-reference estimation of channel-induced video distortion using distributed source coding," in *Proc. ACM Multimedia*, Vancouver, Canada, Oct. 2008.

[7] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Information Theory*, vol. 19, no. 4, pp. 471-480, July 1973.

[8] *Digital Transport of One-Way Video Signals - Parameters for Objective Performance Assessment*, Std., ANSI T1.801-2003, Sept. 2003.

[9] *Objective Perceptual Video Quality Measurement Techniques for Digital Cable Television in the Presence of a Full Reference*, Std., ITU-T Recommendation J.144, 2004.

[10] S. Wolf and M. Pinson, "Low bandwidth reduced reference video quality monitoring system," in *Proc. First International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, Arizona, Jan 2005.

[11] S. Wolf and M. Pinson, "Video quality measurement techniques," NTIA, Tech. Rep. 02-392, June 2002.

[12] D. Varodayan, A. Aaron, and B. Girod, "Rate-Adaptive Codes for Distributed Source Coding," *EURASIP Signal Processing Journal, Special Section on Distributed Source Coding*, vol. 86, no. 11, pp. 3123-3130, Nov. 2006.

[13] "H.264/MPEG-4 AVC Reference Software Manual," Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-AD010, Jan. 2009.

[14] Video Quality Metric software. [Online]. Available: http://www.its.bldrdoc.gov/vqm/