Reduced-Reference Video Quality Assessment using Distributed Source Coding

M. Tagliasacchi, G. Valenzise, M. Naccari, S. Tubaro

Abstract

In this paper we propose a reduced-reference quality assessment algorithm which computes an approximation of the structural similarity (SSIM) metric exploiting coding tools provided by the distributed source coding theory. The algorithm has been tested to evaluate the quality of decoded video bitstreams after transmission over error-prone networks. We evaluate the accuracy of the proposed quality assessment algorithm by measuring the Pearson's correlation coefficient between the structural similarity metric computed in full-reference mode and the one provided by the proposed reduced-reference algorithm. The proposed reduced-reference algorithm achieves good correlation values (higher than 0.85 with packet loss rate equal up to 2.5%).

Index Terms

Reduced Reference Video Quality Assessment, Distributed Source Coding

I. Introduction

The dissemination of video contents over digital networks has gained an increasing popularity in the last few years, thanks to the improvement of network capabilities in speed, reliability and latency, and to the availability of more and more sophisticated video coding standards, such as the recent state-of-the-art H.264/AVC video codec [2]. For example, in one of the most promising digital video broadcasting applications, IPTV, a centralized content provider uses a packed-switched IP network to transmit the compressed video programs to many end-users. Usually, this kind of networks provides only best-effort services, i.e. there is no guarantee that the content will be delivered to clients without errors. In fact,

This work was presented in part in reference [1] and has been developed within VISNET II, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

The authors are with Dipartimento di Elettronica e Informazione, Politecnico di Milano, P.zza Leonardo da Vinci, 32 20133 - Milano, Italy - Ph. +39-02-2399-7619 - FAX: +39-02-2399-7321 - E-mail: {tagliasa, valenzise, naccari, tubaro}@elet.polimi.it.

the video sequence received by the end-user may contain artifacts due to packet losses, which in turn may propagate along the video stream due to the predictive nature of conventional video coding schemes [3], [4], [5]. This must be added to the distortion introduced by quantization of the original bit-stream at the content producer side. On the other hand, in a typical IPTV contract the content provider and the clients stipulate a Service Level Agreement (SLA) that fixes an expected video quality at the end-user terminal: the provider imposes a price to the customers for assuring the agreed Quality of Service (QoS), and pays a penalty if the SLA is unfulfilled. Thus, both the parties need some objective and efficient quality assessment metric which is also supposed to be related to the perceptual video quality perceived by the end-viewers.

A commonly used objective metric for video is the peak signal-to-noise ratio (PSNR), which is based on the mean square error (MSE) between the original and the received frames. Despite its widespread diffusion as distortion metric, due to its simplicity of calculation and the easiness to deal with in optimization, it is well known that PSNR could be poorly correlated with the actual perceived quality, measured through Mean Opinion Score (MOS) tests [6]. In the last decades, a great deal of effort has been made to develop objective video quality assessment techniques which more accurately resemble perceptual scores as given by human observers. Generally speaking, these methods use some preprocessing stage on the reference and the distorted images in order to imitate the functional characteristics of the human visual system (HVS). Examples of these operations include: geometric alignment and point spread function (PSF) filtering in order to emulate optical transfer function associated to the eye optics; light adaption; channel decomposition into different spatial and temporal frequencies, which are subsequently weighted according to the equivalent transfer function of the neuron responses in the primary visual cortex. Then, for each channel, the errors between the original and the distorted images are then normalized and some masking function is usually applied to account for the effects of image components which interfere with the visibility of other spatially or temporally neighbor image regions [7], [8]. Some of these techniques have been validated by the video quality experts group (VQEG) [9], which has tested several video quality measurement systems in the recent VQEG full reference television (FRTV) phase 2 tests [10]; however, there is not a single winning perceptual metric resulting from these tests, since most works focus on specific forms of distortions, such as blocking [11], [12], blurring [13], or sharpness [14]. More recently, Wang et al. have proposed a structural similarity index (SSIM) for evaluating the perceptive quality of degraded images [15]. In contrast with the philosophy outlined above, which assumes the distorted signal as the sum of a perfect quality reference and an error signal whose impact on the HVS has to be evaluated, SSIM tries to quantify the structural distortion in terms of the perceived loss of "structure" in the image. This is justified by the fact that the ultimate goal of the HVS is to extract the structural information from the visual scene. In [8] these ideas have been extended to the case of video sequences to generate a metric that we denote as video SSIM (VSSIM) in this paper. The results reported by the authors show a very high correlation with MOS and a lower outlier ratio than competing methods, which make this metric one of the most promising perceptual evaluation criteria for video quality assessment. Therefore, we adopt this metric in our work to produce perceptually-significant quality monitoring, we will briefly outline how to compute VSSIM in Section III.

The video quality assessment approaches described so far are known as *full reference* (FR) methods, since, to be computed, they require the complete availability of the reference signal. While this is feasible at the content-provider side (where it has been used e.g. for tasks as perceptual rate-distortion optimization [16]), it is practically impossible to directly compute these metrics at the receiver, since the end-users do not have access to the original frames at their terminals. A practical alternative to FR methods is to send to the receiver a small signature of the original content in addition to the video stream: this approach, known as *reduced reference* RR quality assessment (see Section II), is at the basis of the system proposed in this paper. At the content producer side, we extract a compact feature vector which is then transmitted over the RR channel to the receiver, where it is used to estimate the visual quality of the received video stream. In order to produce perceptually significant estimates, the receiver computes an approximation of the VSSIM between the original and the received streams: therefore, the feature vector is assembled in such a way that it contains sufficient information to obtain an estimation of the FR metric. It comes out that the needed bit budget for encoding this feature vector is scalable with the quality of the VSSIM estimate that we want to obtain and, furthermore, by using distributed source codes for encoding the features we show that the RR channel bandwidth can be kept as low as in comparable state-of-the-art reduced-reference quality assessment systems. This bit-rate can be further reduced by exploiting some a-priori map of the support of the channel errors, as the one produced by the error-tracking module devised in [17], to help decoding the feature vector. In contrast with previous works on video quality monitoring [18], [19], we focus on the errors introduced by the transmission channel *only*: thus, the feature vector at the content producer side is extracted from the video reconstructed in the encoder loop.

The rest of this paper is organized as follows: the next section reviews the main reduced-reference video quality assessment systems proposed in the literature; Section III briefly reviews the VSSIM full-reference metric introduced by Wang et al. [8], which is then used in the rest of the paper; Section IV, provides details about the basic building blocks of the proposed system, while Section V validates the quality of the RR VSSIM estimation and the bit-rate required by the RR channel. Finally, Section VI gives some concluding remarks and hints for future works.

II. Related works

In the literature, the techniques for estimating the distortion at the end-user side fall in two main categories: No-Reference (NR) and Reduced-Reference (RR) methods [7].

In NR methods, the end-user does not have any information about the original video stream, and tries to infer the distortion of the received frames from the reconstructed video available at the output of the decoder or from the transmitted bit-stream itself. These techniques can be easily integrated into existing broadcasting systems, but generally lack in estimation accuracy. The NR method in [18] evaluates the distortion introduced by video coding by automatically and perceptually quantifying blocking artifacts of the DCT coded macroblocks. When also channel losses have to be considered, NR techniques extract detailed local information regarding the spatial impact and the temporal extent of the packet loss, as in the work of Reibman et al. [20]. If some error concealment technique is available at the decoder, the distortion can be determined from the macroblocks for which the concealment is judged to have been ineffective, as in [21].

On the other hand, RR methods can achieve a more accurate distortion estimation than NR approaches without assuming the complete availability of the reference signal, by using some feature vector extracted from the original bit-stream that is made available at the decoder side through an ancillary, low bit-rate, noiseless data channel. The first RR video quality assessment systems have been proposed by Wolf et al. [22], [23]. These methods extract localized spatial and temporal activity features from the video sequence: spatial information (SI) measures the effect of compression on the edge statistics of a frame; temporal information (TI) features account for a coarse description of the motion of consecutive frames, through the standard deviation of difference frames. The bandwidth of the RR channel depends upon the size of the windows over which SI and TI features are computed. An approach based on to watermarking has been used in [24] and [25], where the watermarks are inserted in the video frames, and at the decoder the error rate on the marks is used as an indicator of the distortion introduced in the frame. However, these techniques are not truly RR metrics and, moreover, they may introduce further distortion in the transmitted sequence. Pinson and Wolf [26] have presented an RR video quality monitoring technique that uses less than 10 kbits/s of reference information. The system is based on the same features used in the NTIA General Video Quality Model (VQM) [10]: the amount and angular distribution of spatial gradients and chrominance information are extracted from spatio-temporal windows (32x32 pixel x 1 second). This is a sort of subsampling of the same features used for NTIA, which are extracted in the same identical way but with a finer granularity. The features are then quantized through a non-linear quantizer. To take into account brief temporal disturbances (due for instance to channel errors), an absolute temporal information feature is extracted by taking the root-mean-square error between groups of frames with a length of 0.2 seconds, which is close to the human eye temporal resolution: this temporal spacing enables to easily capture abrupt changes in video frames due to lost slices or macroblocks. In the system of Yamada et al. [19], a representative-luminance value for each original video frame is chosen from the blocks of the images having most of their energy in the mid-frequency range, and a binary map with the pixels having that luminance value is entropy-coded and transmitted as RR information. At the decoder the visual quality is assessed by computing the PSNR only on the pixels marked on the map. While the proposed system is intrinsically scalable (varying the number of chosen pixels one can achieve different RR bit-rates), no specific analysis of the relationship between rate and the quality of the estimated PSNR is provided. A scalable video distortion metric is also proposed in [16], which is obtained by representing the video sequence as a tree of wavelet coefficients, starting from the root (coarsest spatial scale) and proceeding to the leaves (finest scales). The bit-rate of the representation may be scaled from full reference to reduced reference by reducing the depth and number of such trees: in the RR representation, the required bit-rate is less than 10 kbits/s. A very compact RR information can also be obtained by an appropriate natural image statistic model in the wavelet domain. Wang and Simoncelli [27] compare the marginal probability distribution of wavelet coefficients in different wavelet subbands with the probability density function of wavelet coefficients of the decoded image, using Kullback-Leibler divergence as a distance between distributions. To reduce the RR bandwidth while at the same time keeping as good as possible the estimation accuracy, a generalized Gaussian model is fitted to the histograms of the wavelet coefficients, and just the parameters of the distributions are sent to the decoder over the RR channel. The system, originally developed for JPEG2000 still images, does not assume any specific distortion type on the transmitted signal, and has been proved to be effective with degradations due to compression as well as to channel losses. However, the application of the system to video sequences would require to model also the temporal redundancies between frames to be competitive with other approaches.

In order to keep the size of the transmitted feature vector as small as possible, Chono et. al. [28] employ Distributed Source Coding (DSC) tools to efficiently encode the partial reference data, in the scenario of image delivery with distortion due to compression. Some portion of the whitened spectrum of the original image is transmitted as the feature vector, according to a predetermined admissible image quality; at the end-user terminal, the decoder reconstructs the feature vector using the received image as side information and estimates the PSNR from the feature measurements. Inspired by the DSC approach described in that paper for still images, we have proposed in [1] an RR video quality assessment system for channel-induced errors such as packet losses. The content producer builds a feature vector consisting of a few random projections of each frame macroblock, which are then coded using DSC and sent over the RR channel to the receiver; there, the feature vector is used to estimate the MSE at the macroblock level, leveraging when possible some a-priori map of the support of the channel errors to facilitate the decoding of the features. In this paper, we extend our previous work by considering the VSSIM perceptual metric in place of the MSE.

III. Video Structure Similarity Index

The structural similarity index (SSIM), proposed by Wang et al. [15], [8], is a measure of the distance between two images from a "structural information" point of view. While there are many ways of defining a structural information, the proposers of SSIM identify the structural distortion as a product of independent terms, namely the luminance, the contrast and the similarity between two images. This leads to the following SSIM formula [15] between signals $\mathbf{x}$ and $\mathbf{y}$:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{1}$$

where μ_x and μ_y are, respectively, the mean values of $\mathbf{x}$ and $\mathbf{y}$ (which account for the luminance term), σ_x^2 and σ_y^2 represent their variances (an approximation of the contrast) and σ_{xy} is the covariance between the two signals (which, normalized by the variance, gives the cosine distance between $\mathbf{x}$ and $\mathbf{y}$ and gives an indication of their similarity). The two constants C_1 and C_2 are added to avoid possible division by zero, and are selected as $C_i = (K_iL)^2$, $i = 1, 2$ [15], with L equal to the dynamic range of the pixel values (e.g. $L = 255$ for a 8 bits/pixel image); we set $K_1 = 0.01$ and $K_2 = 0.03$ [8]. It is straightforward to show that: a) $\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \mathrm{SSIM}(\mathbf{y},\mathbf{x})$; b) $\mathrm{SSIM}(\mathbf{x},\mathbf{y}) \leq 1$; and c) $\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = 1$ iff $\mathbf{x} = \mathbf{y}$.

The SSIM is computed with a block granularity, the block size depending on many factors such as the type of image or the target application. In [15], the SSIM is calculated on 8×8 sliding windows moved pixel by pixel; in the case of video frames [8], to reduce the computational complexity, the SSIM is computed only on a random subsample of the total number of blocks. In this work, we compute the SSIM (and the related VSSIM, see the following) on a grid of disjoint blocks of size 32×32; nevertheless it can be shown with experimental simulations that the so-computed SSIM is very close to the one that would have been obtained by implementing it as described in [8]. As a matter of fact, in our experiments we have measured a Pearson's correlation coefficient of 0.9846 between the frame level SSIM computed as in [8] and the one computed over a disjoint grid of macroblocks.

Once the SSIM has been calculated for each macroblock, the video SSIM or VSSIM [8] is obtained by aggregating the SSIM either at the frame or sequence levels. The local quality values are combined into a frame-level quality index VSSIM_t:

$$\mathrm{VSSIM}(t) = \frac{\sum_{j=1}^{N} w_j \mathrm{SSIM}_j}{\sum_{j=1}^{N} w_j}, \tag{2}$$

where t is a frame index, N is the total number of blocks in a frame, while SSIM_j and w_j are, respectively, the local value of SSIM for macroblock j of frame t and its weighting factor. The weights w_j are selected as in [8] to consider the fact that dark regions of a frame do not attract fixations and therefore a smaller weighting values can be assigned. Using the mean μ_x of the luminance component of each block as an estimate of the local luminance, the weights w_j are set as:

$$w_j = \begin{cases} 0 & \mu_x \leq 40 \\ (\mu_x - 40)/10 & 40 < \mu_x \leq 50 \\ 1 & \mu_x > 50 \end{cases} \cdot \tag{3}$$

The VSSIM at the sequence-level is computed by a weighted average of the frame-level VSSIM's, with weights W_t:

$$\mathrm{VSSIM} = \frac{\sum_{t=1}^{T} W_t \cdot \mathrm{VSSIM}(t)}{\sum_{t=1}^{T} W_t}, \tag{4}$$

where T is the total number of frames in the sequence. This time, the weighting coefficients W_t take into account the average motion of each frame, with the rationale that errors occurring in fast moving scenes are less annoying than errors in a still or slowly moving background. Thus, to adjust the weights W_t we consider the average normalized length M_t of the motion vectors of frame t:

$$M_t = \frac{\sum_{j=1}^{N} m_j}{N \cdot K_M}, \tag{5}$$

where m_j is the motion vector length for block j of frame t, and K_M is the motion vector search range (e.g. $K_M = 16$ pixels). It is important to notice that macroblock motion vectors can be obtained as side product of the decoding process without any additional cost. Finally, the weights W_t are obtained as a

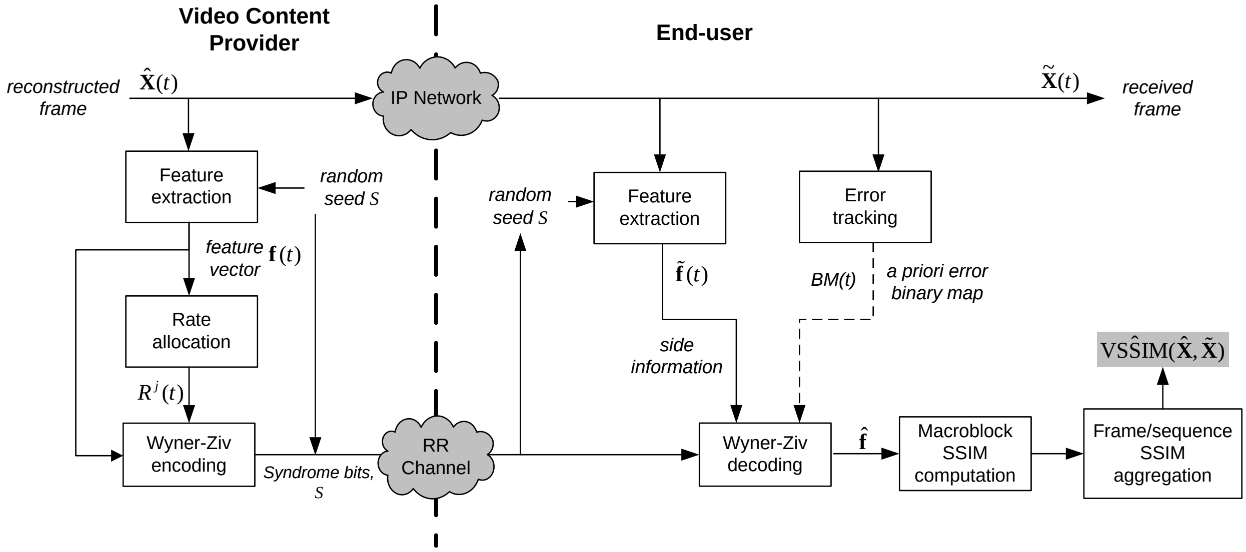


Fig. 1. The overall schema for the proposed RR quality assessment algorithm.

by-product of the average motion per frame [8]:

$$W_t = \begin{cases} \sum_{j=1}^{N} w_j(t) & M_t \leq 0.8 \\ ((1.2 - M_t)/0.4) \sum_{j=1}^{N} w_j(t) & 0.8 < M_t \leq 1.2 \\ 0 & M_t > 1.2 \end{cases}, \tag{6}$$

where we have added a frame index to $w_j(t)$ to specify that we refer to the weights w_j of frame t.

IV. Proposed reduced-reference system

Figure 1 depicts the proposed RR quality assessment algorithm. At the video content provider side, the error-free reconstructed frame at time t, $\hat{X}(t)$, is fed into the *Features extraction* module which computes the features vector $\mathbf{f}(t)$. The feature vector is then presented as input to the *Rate allocation* module which provides as output the quantity $R^j(t)$, i.e. an estimate of the syndrome bits rate R used for the DSC coding of the j-th bitplane of the feature vector at time t. The last step performed at the video content provider side consists in the Wyner-Ziv encoding of the feature vector $\mathbf{f}(t)$. The Wyner-Ziv encoder computes the syndrome bits for each bitplane j of $\mathbf{f}(t)$ on the basis of $R^j(t)$; thus, the information sent over the noiseless RR channel consists of the syndrome bits of $\mathbf{f}(t)$ and of the random seed S used to generate the features (see Section III). The coded frames $\hat{X}(t)$ are sent through an error-prone channel that drops transmitted packets according to a given packet loss rate (PLR).

Algorithm 1 Operations performed at the content-provider side

Require: encoder reconstructed video sequence $\hat{X}$

$t = 1$
for each error-free reconstructed frame $\hat{X}(t)$ **do**
  for each macroblock m belonging to $\hat{X}(t)$ **do**
    Compute the feature vector $\mathbf{f}$
  end for
  Compute the syndrome bits rate $R^j(t)$ for each bitplane j of $\mathbf{f}$
  Perform Wyner-Ziv encoding of $\mathbf{f}$ based on $R^j(t)$ by performing syndrome bits computation
  Send the coded data and syndrome bits through, respectively, the noisy and the RR channels
  $t = t + 1$
end for

At the receiver side, the decoded frame $\tilde{X}(t)$ is used to extract the feature vector $\tilde{\mathbf{f}}$ which might differ from its counterpart at the transmitter side due to channel errors. From the received bitstream, the decoder knows whether or not any packet has been lost in the transmission, and therefore which macroblocks are corrupted due to channel errors. The *Error tracking* module builds a binary map $BM(t)$ that assigns to each macroblock a flag that assesses the integrity or the corruption of that block. As detailed in Section IV-C, a macroblock is flagged as "noisy" if it belongs to a lost packet of the current frame or if it depends by motion-compensation from previously corrupted macroblocks. The binary map together with the feature vector $\tilde{\mathbf{f}}$ are passed to the *Wyner-Ziv decoder* to correct the side information $\tilde{\mathbf{f}}$ into a reconstructed version $\hat{\mathbf{f}}$. Finally, the vector $\hat{\mathbf{f}}$ is used to compute the structural similarity metric for each macroblock. This information is aggregated at the frame and/or sequence level to provide the objective score VSSIM$(\hat{X}, \tilde{X})$. A summary of the above description is reported as pseudo-code in Algorithms 1 and 2, while in the ensuing subsections we provide all the details about the modules illustrated in Figure 1.

A. Feature Extraction and RR VSSIM approximation

In order to compute a RR approximation of the VSSIM at the decoder, we need to extract a significant feature vector $\mathbf{f}$ for each frame from the error-free reconstructed video sequence at the encoder. The feature vector consists of the mean values μ_x of each macroblock $\mathbf{x} \in \mathbb{R}^n$ in the frame, with n the number of pixels in a macroblock, and of a set of m random projections $\mathbf{y} \in \mathbb{R}^m$ computed for each macroblock. The random projections $\mathbf{y} = \mathbf{A}\mathbf{x}$, $\mathbf{A} \in \mathbb{R}^{m \times n}$, are computed as scalar products between the macroblock $\mathbf{x}$ and Gaussian i.i.d. vectors $\mathbf{a}_i$, $i = 1 \ldots m$, generated pseudorandomly starting from a seed S, and normalized in such a way that $\|\mathbf{a}_i\| = 1$. Thus, for frame t the feature vector is the concatenation

Algorithm 2 Operations performed at the receiver side

Require: Received bitstream $\tilde{X}$

$t = 1$
for each frame $\tilde{X}(t)$ **do**
    Decode the received packets and apply error concealment algorithm if any packet has been lost
    Compute the feature vector $\tilde{\mathbf{f}}$
    Apply the error tracking module to produce the binary map $BM(t)$
    Perform Wyner-Ziv decoding using the received syndrome bits and $BM(t)$
    Compute the structural similarity at the macroblock level and aggregate at the frame and sequence level
    $t = t + 1$
end for

of the means and random projections of each macroblock in the frame, and its dimension is $N(m+1)$.

The feature vector is encoded and transmitted to the content-receiver through the RR channel, as detailed in the subsequent sections. Once decoded, a quantized version $\hat{\mathbf{f}}$ of the feature vector computed from the content-provider is used to calculate an approximation of the VSSIM between the received video stream and the error-free sequence reconstructed at the encoder. As explained in Section III, the basic ingredients for computing the VSSIM are: 1) the means μ_x and $\mu_{\tilde{x}}$ of, respectively, the original and the received macroblocks $\mathbf{x}$ and $\tilde{\mathbf{x}}$; 2) the variances σ_x^2 and $\sigma_{\tilde{x}}^2$ of $\mathbf{x}$ and $\tilde{\mathbf{x}}$; and finally 3) the covariances $\sigma_{x\tilde{x}}$ between each error-free macroblock and its corrupted version. Clearly, the means $\mu_{\tilde{x}}$ and the variances $\sigma_{\tilde{x}}^2$ can be readily computed from the received frame $\tilde{X}(t)$; in addition, a quantized version of the original means $\hat{\mu}_x$ can be recovered from the feature vector. However, the variances σ_x^2 and the covariances $\sigma_{x\tilde{x}}$ are not immediately available from the feature vector, and must be estimated at the decoder side using the received video sequence and the RR information. An estimate of the variances $\hat{\sigma}_x^2$ is given by:

$$\hat{\sigma}_x^2 = \frac{1}{m}\sum_{i=1}^{m} \hat{y}_i^2 - \hat{\mu}_x^2, \tag{7}$$

where $\hat{y}$ denotes the quantized random projections in the feature vector. For the estimation of the covariances, the content-receiver extracts for each macroblock a vector of random projections $\tilde{\mathbf{y}} = \mathbf{A}\tilde{\mathbf{x}}$, in a similar fashion to the feature vector computation at the encoder-side, using the same seed S received through the RR channel. At this point, an estimate $\hat{\sigma}_{x\tilde{x}}$ of the covariances between the macroblocks $\mathbf{x}$ and $\tilde{\mathbf{x}}$ is obtained as:

$$\hat{\sigma}_{x\tilde{x}} = \frac{1}{2}\left[\hat{\sigma}_x^2 + \sigma_{\tilde{x}}^2 + (\hat{\mu}_x - \mu_{\tilde{x}})^2 - \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - \tilde{y}_i)^2\right]. \tag{8}$$

B. Rate Allocation and Wyner-Ziv Encoding

Normally, in Distributed Video Coding (DVC) scheme, the encoder sends to the decoder syndrome (or parity) bits until a correct decoding is obtained. A feedback channel is used to implement this strategy. When there is no feedback channel available at the transmitter side, one needs to estimate the number of syndrome bits for each bitplane j of the features vector $\mathbf{f}$ in frame t. In order to simplify the notation, in the following we discard the frame index t, denoting $X = \mathbf{f}_l(t)$ and $Y = \tilde{\mathbf{f}}_l(t)$, $(l = 1 \dots m + 1)$ respectively as the l-th features source and the side information. Also, we assume the following additive noise model:

$$Y = X + Z, \tag{9}$$

where Z is the correlation noise, assumed statistically independent from X. To approximate the distribution of X we use either a Gaussian or a uniform distribution depending on whether we are considering, respectively, the random projections or the mean value.

Let σ_z^2 denote the average distortion between $\mathbf{f}$ and $\tilde{\mathbf{f}}$. In our experiments, we verified that the distribution of the correlation noise Z can be modeled by a Laplacian distribution $p_Z(z)$.

The values for σ_z^2 has been derived by a simple distortion estimation algorithm which works as follows. For the video sequences considered in our experiments, we simulated 30 content transmissions over an error-prone channel and we measured the average Peak Signal-to-Noise Ratio (PSNR) for each decoded sequence. The video sequences have been encoded with the same parameters as described in Section V whereas the error patterns have been generated with a two state Gilbert's model [29] with PLR in the range $[0.1 \dots 2.5]$ and average burst length of 3.1 packets as typically assumed in IP networks [30]. The average values of PSNR $(\bar{K})$ are then related to σ_z^2 by the following:

$$\sigma_z^2 = 255^2 \cdot 10^{-\bar{K}/10}. \tag{10}$$

In Figure 2 we show the average value of PSNR for the two considered video sequences (i.e. the *Mobile & Calendar* and the *Rugby* downloaded by [9]) together with the confidence levels for each value of PLR.

In the computation of σ_z^2 for each PLR we choose the average value of PSNR for each tested sequence diminished of the 10% in order to be conservative with respect to the effect of channel losses.

The rate allocation algorithm (see Figure 3) receives in input the source variance σ_x^2, the correlation noise variance σ_z^2, the quantization step size δ and the number of bitplanes to be encoded J and returns

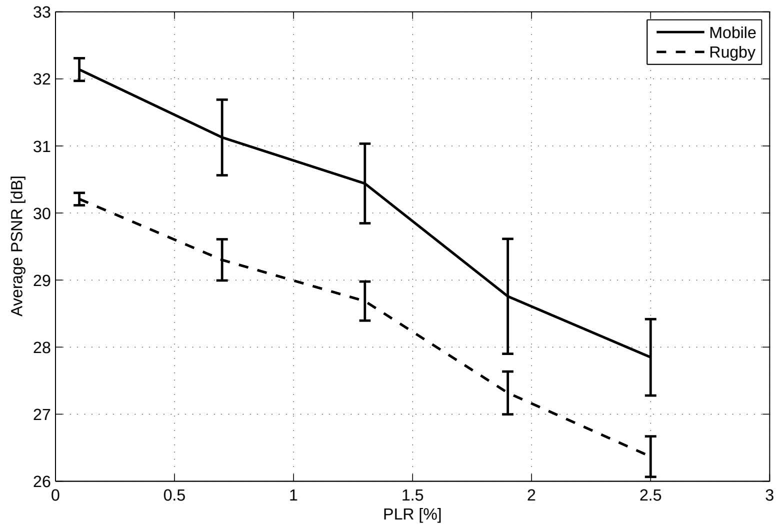


Fig. 2. Average PSNR value for the two considered video sequences.

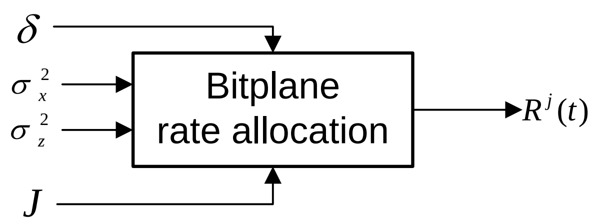


Fig. 3. Illustration of the rate allocation algorithm.

the average number of bits needed to decode each bitplane $R^j, \quad j = 1, \ldots, J$. The Shannon's lower bound on the number of bits is given by

$$R^j \geq H(x^j|Y, x^{j-1}, x^{j-2}, \ldots, x^1) \quad \text{[bits/sample]}, \tag{11}$$

where x^j denotes the j-th bitplane of the source X. In fact, as detailed in the next section, Wyner-Ziv decoding of bitplane j exploits the knowledge of the real-valued side information Y as well as previously decoded bitplanes $x^{j-1}, x^{j-2}, \ldots, x^1$. The value of R^j from equation (11) can be readily computed by numerical integration. In fact, the expression of the entropy in (11) can be written as

$$\begin{aligned} H(x^j|Y, x^{j-1}, x^{j-2}, \dots, x^1) = H(x^j|Y, Q) &= \sum_{q=1}^{2^{j-1}} p(q) H(x^j|Y, Q = q) \\ &= \sum_{q=1}^{2^{j-1}} p(q) \int_{-\infty}^{+\infty} p_{Y|q}(y) H(x^j|Y = y, Q = q) dy, \end{aligned} \tag{12}$$

where Q denotes the quantization bin index obtained decoding up to bitplane $j-1$. The value of $H(x^j|Y = y, Q = q)$ represents the entropy of the binary source x^j when the side information assumes the specific value y and the source X is known to be within quantization bin q, i.e.,

$$H(x^j|Y = y, Q = q) = -p_0 \log_2 p_0 - (1 - p_0) \log_2 (1 - p_0), \tag{13}$$

and

$$p_0 = \Pr\{x^j = 0|Y = y, Q = q\} = \frac{\int_{L_q}^{(L_q+U_q)/2} p_X(x) p_Z(y - x) dx}{\int_{L_q}^{U_q} p_X(x) p_Z(y - x) dx}, \tag{14}$$

where L_q and U_q are the lower and upper thresholds of the quantization bin q.

The expression $p_{Y|q}(y)$ in (12) represents the marginal distribution of Y, conditioned on $X \in q$, and it can be obtained, according to the additive correlation model (9), from the knowledge of the joint distribution $p_{XY}(x, y) = p_X(x) p_Z(y - x)$, i.e.,

$$p_{Y|q}(y) = \frac{\int_{L_q}^{U_q} p_{XY}(x, y) dx}{\int_{L_q}^{U_q} p_X(x) dx} = \frac{\int_{L_q}^{U_q} p_X(x) p_Z(y - x) dx}{\int_{L_q}^{U_q} p_X(x) dx}. \tag{15}$$

Finally, equation (12) can be evaluated by means of numerical integration over y.

To implement the WZ codec, we use LDPC Accumulate codes [31]. The number of bitplanes passed as input to the LDPC encoder for the feature vector $\mathbf{f}$ corresponds to the number of bits that would have been needed if a simple uniform scalar quantizer had been designed in such a way that the signal-to-quantization noise ratio (SQNR) is fixed to 30 dB. The LDPC returns a syndrome for each bitplane.

C. Error Tracking and Wyner-Ziv Decoding

At the receiver side the actual error pattern is known exactly, therefore an error tracking module uses this information to determine which blocks might be affected by errors. The error tracker produces, for each frame, a binary map $BM(t)$ that indicates whether or not the reconstructed block at the decoder might differ from the corresponding one at the encoder. Figure 4 illustrates the binary map computation performed by the error tracking module. A channel error corrupts a coded packet in frame $t-1$. Each slice corresponds to a coded packet, thus corrupting a coded packet results in a damaged row of macroblocks. The binary map at time $t-1$ flags the macroblocks corresponding to the corrupted packet as "noisy" (a 1 entry in $BM(t-1)$). Due to the motion-compensation performed in the decoder loop, the noisy macroblocks will corrupt also the ones whose temporal predictor overlaps with the damaged area in frame $t-1$. In fact, at time t, $BM(t)$ will contain a 1 entry for each corrupted macroblock in the 7-th row (see Figure 4) and furthermore a 1 entry for each macroblock where temporal error propagation occurs as shown in the marked macroblocks at row 4 and 5 in Figure 4. The error tracking module adopted in our RR quality assessment algorithm is similar to the one presented in [32], with the important difference that in our case error tracking is performed at the decoder only.

The Wyner-Ziv decoder can exploit the information contained in the binary map by setting σ_z^2 to zero. However, our simplified error tracking scheme does not account for errors due to intra-prediction or to de-blocking filters propagations. In order to compensate for this inaccuracy we set $\sigma_z^2 = \sigma_L^2$ for those feature vector elements belonging to correctly received macroblocks (a 0 entry in $BM(t)$). Conversely, for the other elements (i.e. those belonging to macroblocks with a 1 entry in $BM(t)$) we set $\sigma_z^2 = \sigma_H^2$. In both the aforementioned situations, the conditional probabilities are calculated using (14).

The WZ decoder (see Figure 5) takes the feature vector $\tilde{\mathbf{f}}$ computed from the concealed frame $\tilde{X}(t)$, then the received syndrome bits are used to "correct" each element of the feature vector $\tilde{\mathbf{f}}(t)$ in order to obtain $\hat{\mathbf{f}}(t)$. LDPC decoding is applied at the bitplane level, starting from the most significant bitplane of each element. After LDPC decoding, some residual errors might occur if the number of received syndrome bits is not sufficient for the exact reconstruction of every bitplane. Error detection at the LDPC decoder is performed by comparing the received syndrome bits with a syndrome generated from the decoded bitplane. In this case, if decoding of bitplane j fails, reconstruction is based on bitplanes $1, \dots, j-1$ only. As before, by denoting with q the quantization bin index obtained decoding up to bitplane $j-1$, optimal reconstruction is obtained by computing the centroid of the quantization interval (L_q, U_q) exploiting the Laplacian model

$$E[X|X \in (L_q, U_q), Y = \tilde{\mathbf{f}}_l^j(t)] = \frac{\int_{L_q}^{U_q} x \, p_X(x) p_Z(x - \tilde{\mathbf{f}}_l^j(t)) dx}{\int_{L_q}^{U_q} p_X(x) p_Z(x - \tilde{\mathbf{f}}_l^j(t)) dx}, \tag{16}$$

where $p_Z(z)$ has been defined in the previous subsection, parameterized by the two-valued σ_z^2. In evaluating (16), we need the knowledge of σ_x^2, which is not directly available at the decoder. Assuming the statistical independence between X and Z and the additive noise model (9), we obtain $\sigma_x^2 = \sigma_y^2 - \sigma_z^2$. In the previous expression, σ_y^2 is estimated from the received macroblocks and $\sigma_z^2 = \{\sigma_L^2, \sigma_H^2\}$ as stated

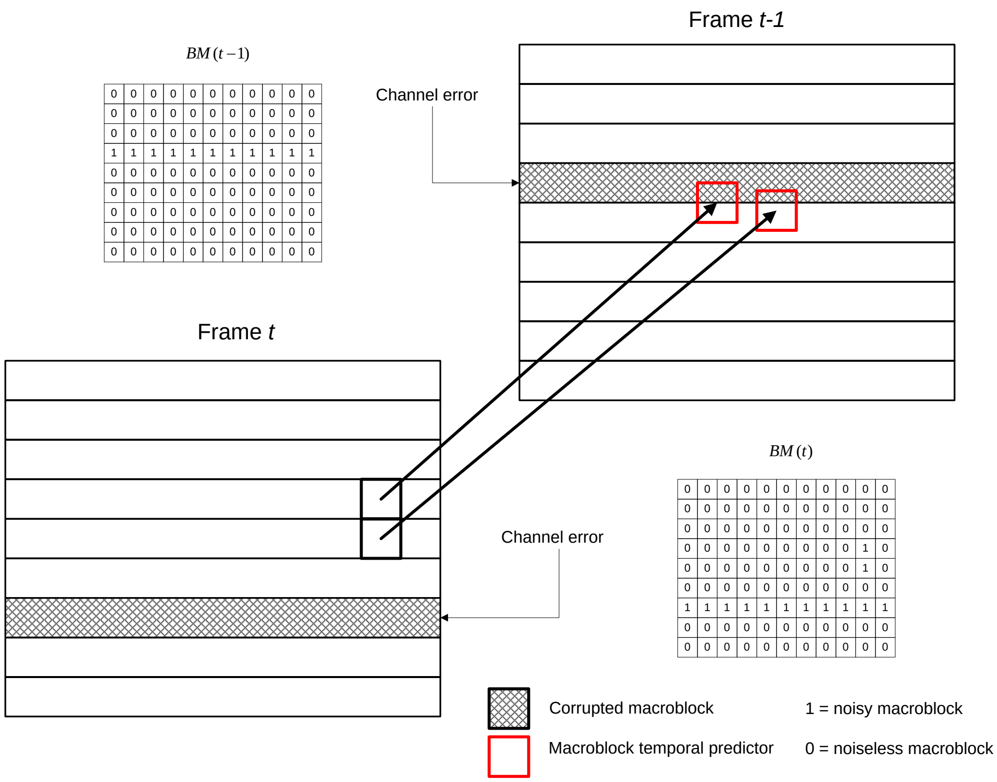


Fig. 4. Graphical illustration of the binary map computation performed by the error tracking module.

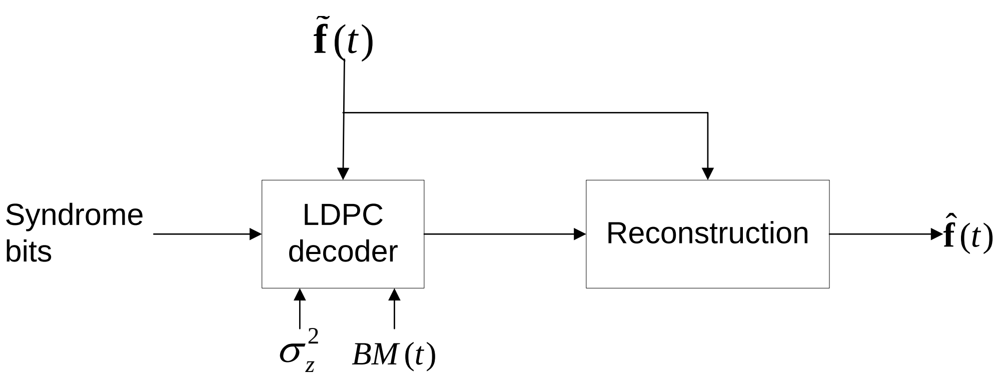
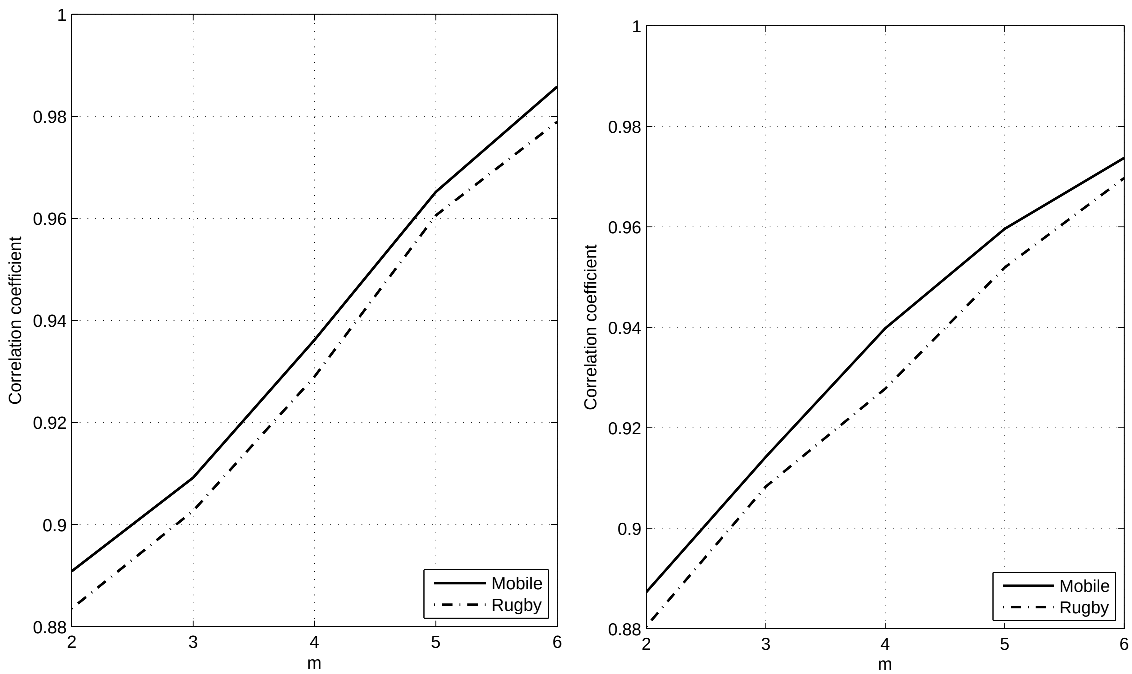


Fig. 5. Block diagram of the Wyner-Ziv decoder to correct $\tilde{\mathbf{f}}(t)$ into $\hat{\mathbf{f}}(t)$.

above.

Finally, the corrected $\hat{\mathbf{f}}$ is fed in the "*Macroblock SSIM computation*" module to compute the RR version of the SSIM metric.



(a) ρ as a function of m for PLR equal to 0.1%

(b) ρ as a function of m for PLR equal to 2.5%

Fig. 6. ρ as a function of m when the PLR is equal to 0.1 and 2.5% and the feedback channel is used.

V. Experimental results

In our experiments two 625 Standard Definition (SD) (720 $\times$ 576 pixels) sequences, namely *Mobile & Calendar* and *Rugby* (downloaded from the VQEG website [9]) have been used. As a first step, the sequences are encoded with the H.264/AVC encoder with the Main profile. The coded bit-rate is 4.5 Mbps while the temporal resolution is 25 fps with an intra coded frame every 15 frames; the total number of frames per sequence is 220. Each frame is divided into slices, each one containing one horizontal row of macroblocks. Each coded slice is packetized using Real-time Transport Protocol (RTP) specifications. The errors on the transmission channels are simulated according to a two state Gilbert's models [29] with average burst length of 3.1 packets and Packet Loss Rate (PLR), expressed in percentages, ranging in $[0.1, \ldots 2.5]$. In order to extract the feature vector, we divide each frame in blocks of size 32×32, and compute m random projections as described in Section IV-A.

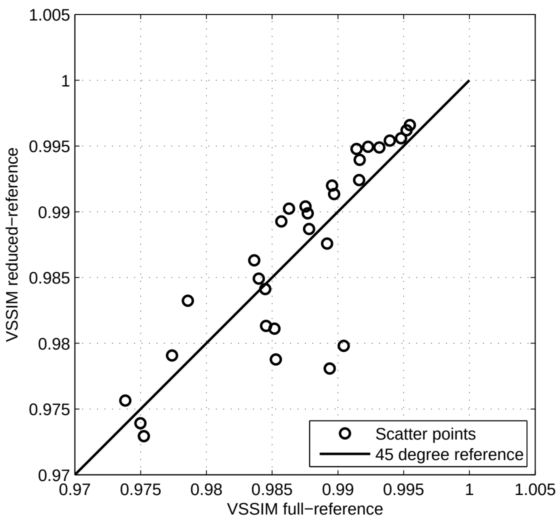
A. Impact of the number of measurements m on the quality of VSSIM estimation

A first kind of experiment aims at assessing the relationship between the reliability of the estimated perceived quality and the number of measurements m, where $m = [2, 3, 4, 5, 6]$. To evaluate the accuracy of the estimated video quality, we compute the Pearson's correlation coefficient ρ between the estimated VSSIM (computed as described in Section IV-A) and ground-truth full-reference VSSIM data. The rationale behind this evaluation criterion is that FR VSSIM is well-correlated with MOS [8], and thus a high correlation between our RR estimated VSSIM and the FR VSSIM implies a good correlation between RR VSSIM and the MOS. In order to compute ρ, we estimate the VSSIM as in equation (4) for 30 channel realizations for each considered PLR and each tested sequence; ρ is therefore the (sequence level) correlation coefficient calculated across the different channel realizations. Figure 6 depicts the correlation for the two tested sequences under the hypothesis that all the rate needed to correctly decode the features is available, when the PLR is, respectively, 0.1% and 2.5%. Unsurprisingly, as the number of measures increases, so does the correlation ρ. This is reasonable, since the random projections $\mathbf{y}$ are used to compute both $\hat{\sigma}_x^2$ (7) and $\hat{\sigma}_{x\tilde{x}}$ (8); as m gets larger, the variance of the estimates becomes smaller, and thus also the estimated VSSIM is more precise. This phenomenon does not depend on the particular tested sequence nor on the PLR but only on the quantization of the random projections.

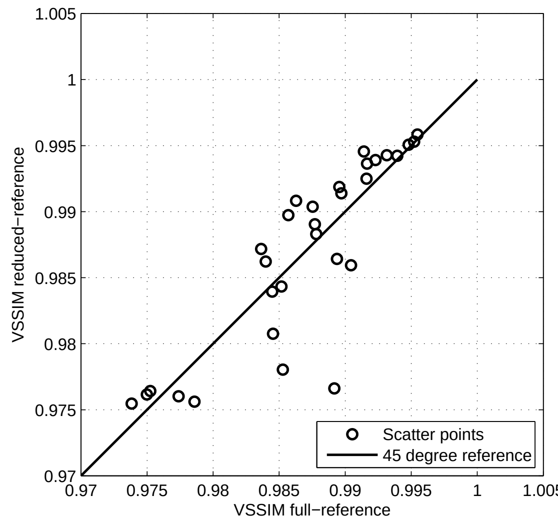
The high correlation between the estimated and the true VSSIM can be also inferred by looking at Figures 7(a)-7(d), where four examples of scatter plots for $m = 2$ and $m = 4$ are reported for both the considered sequences. Each point in the picture corresponds to a channel realization, at a PLR equal to 1.3%.

B. *Rate spent for the reduced-reference information*

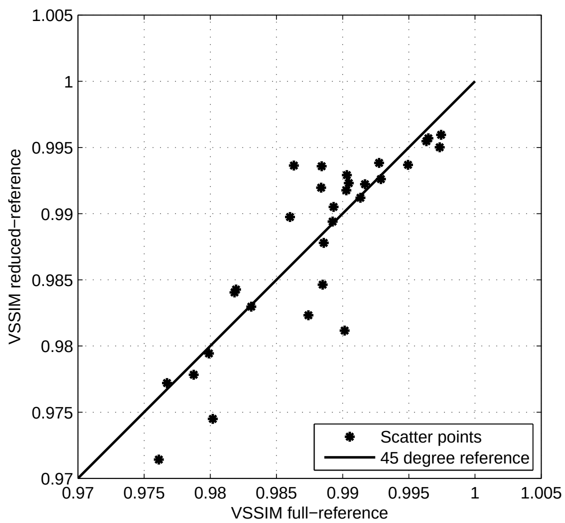
Figure 8 shows the correlation coefficient ρ for varying rates (corresponding to $m = 2, 3, 4, 5, 6$), for the two tested sequences, when the PLR is set to 0.1% and 2.5%. Both the cases where feedback is used or not are illustrated. To incorporate the binary prior map BM described in Section IV-C we have set σ_L^2 of the uncorrupted blocks to 0.016, while the variance of the macroblocks with errors is $\sigma_H^2 = 1600$. Obviously, as the rate spent increases, so does the correlation of the estimated VSSIM with ground-truth data. It can be observed that as the sequence complexity increases (e.g. in *Rugby* there is much more motion than in *Mobile & Calendar*), more bits have to be transmitted on the RR channel in order to obtain the same quality assessment performance, due to the higher energy of correlation noise produced by drift. The use of the rate allocation module (i.e. transmission without feedback) deteriorates both the bandwidth performance and the measured correlation values. As an example, when the PLR is equal to 0.1% and $m = 6$ the rate overhead is 41% with a loss in the correlation value of 4.3% with respect to the case with feedback channel. Conversely, when the PLR is equal to 2.5% and $m = 6$ the rate overhead is 36.87% with a loss in the correlation value of 8% with respect to the case with feedback channel.



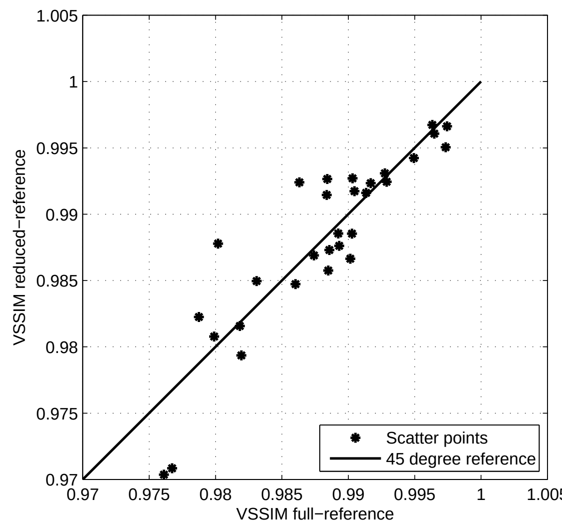
(a) *Mobile & Calendar* with $m = 2$



(b) *Mobile & Calendar* with $m = 4$



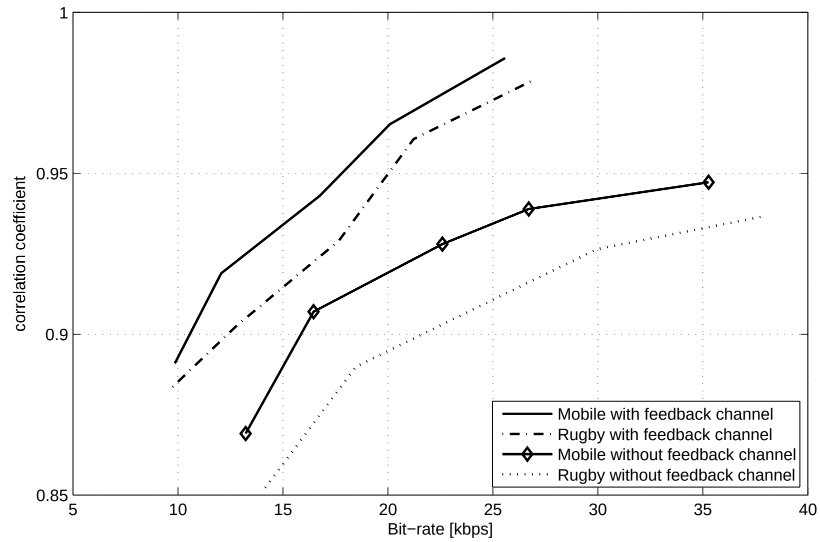
(c) *Rugby* with $m = 2$



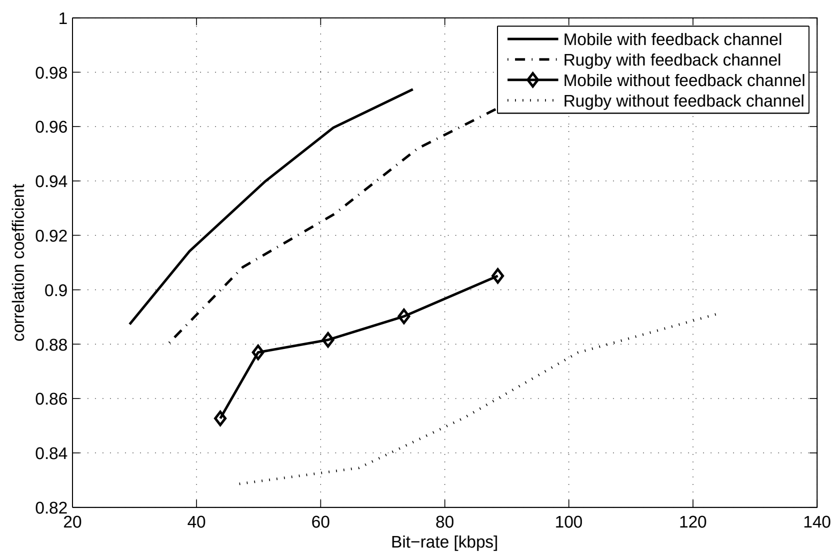
(d) *Rugby* with $m = 4$

Fig. 7. Scatter plots for the considered sequences for $m = 2, 4$ and when the PLR is equal to 1.3%

The large overhead introduces in both cases has an intuitive explanation in the fact that the correlation noise Z, which in this case is represented by the channel-induced errors, is far from being stationary. In fact, channel errors usually occur in bursts and thus the noise energy is concentrated in a few slices. Conversely, the model depicted in equation (9) implicitly assumes that the noise statistics are stationary, i.e. the noise energy is supposed to be equally spread along the whole signal. For these reasons the use of DSC in this particular application scenario requires in practice a feedback channel to be present in order to transmit the RR information effectively spending a reasonable bit budget.



(a) ρ as a function of the bit-rate for PLR equal to 0.1%



(b) ρ as a function of the bit-rate for PLR equal to 2.5%

Fig. 8. ρ as a function of the bit-rate when the PLR is equal to 0.1 and 2.5%.

The dependence of the RR rates from the PLR is illustrated in Figure 9, when $m = 4$, for the two tested sequences. We notice again that drift propagation afflicts more heavily the sequence with higher complexity (*Rugby*); moreover, the degradation due to drift increases for larger PLR since the average number of corrupted slices increases as well. In fact, many of the used concealment strategies in modern decoders are far from perfect, and while in general the concealment is satisfactory for small corrupted frame regions, it is more difficult to correct larger portions of a frame or of consecutive frames. Due to

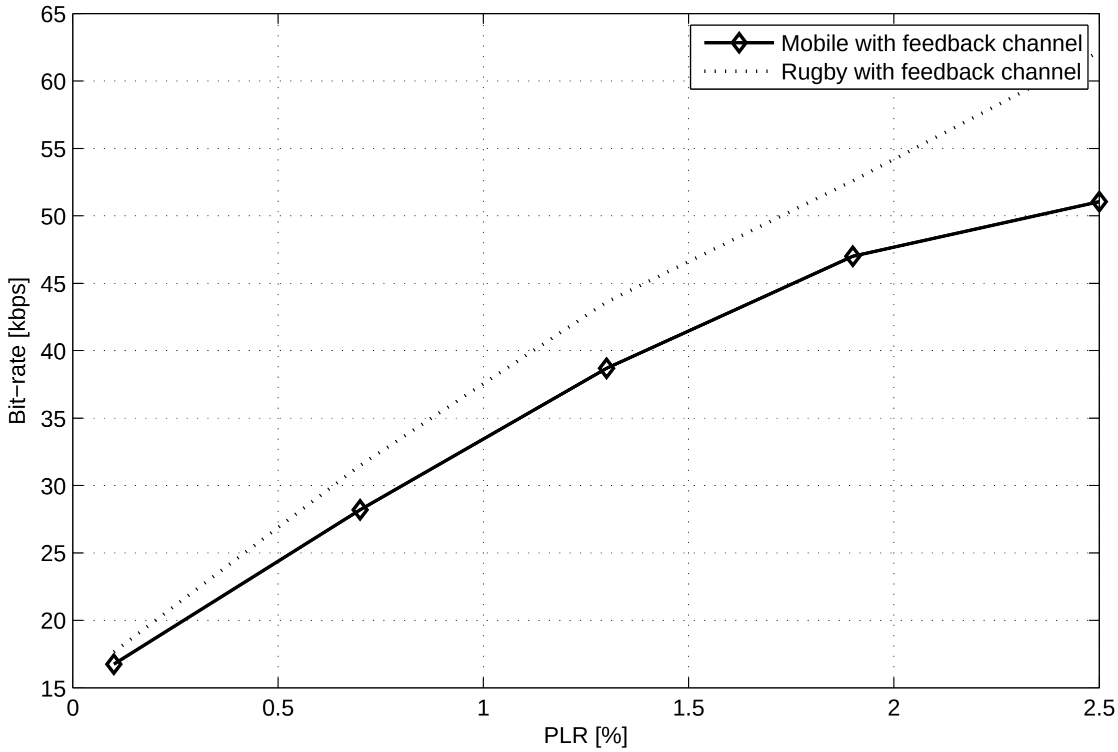


Fig. 9. Dependence of the RR bit-rate from the PLR.

the "bursty" nature of the channel errors, a high PLR is likely to produce video deteriorations which are poorly concealed (especially in intra-coded frames); moreover, due to motion prediction, these errors are likely to propagate to subsequent frames, and also more bits are requested to encode the RR information, since the rate requested for this information depends on how much the received sequence (and thus the extracted feature vector) differs from the original one (10).

VI. Conclusions and future research directions

In this paper we proposed a reduced-reference quality assessment algorithm to compute an approximation of the VSSIM metric by means of distributed source coding tools. Our results show that good correlation values (higher than 0.85) exist between the VSSIM computed in full-reference modality and the RR one. Moreover, we have explored how to allocate the bit-rate for the RR information by proposing a simple rate allocation algorithm. Due to the non-stationary nature of channel errors, the overhead introduced by removing the feedback channel is relevant and thus in practice solutions where a feedback channel is available have to be preferred. In order to estimate the video quality for sequences transmitted over channels whose PLR is lower than 1% (i.e. in most modern IP networks), our system requires that the RR information sent over the RR channel is coded spending about 20 kbits/s, if an expected correlation with the true VSSIM of 0.85 is tolerated. This communication rate is nowadays achievable ubiquitously thanks to advances in network technologies: thus our system represents a practical solution for accurately estimating the perceived quality of a video stream.

Further improvements to the system could be obtained by accurately modeling the probability density function of the features vector $\mathbf{f}$ in the rate allocation module and by adopting a more accurate and general distortion estimation algorithm for the term σ_z^2 in equation (10).

REFERENCES

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

[2] ITU-T, *Final Draft International Standard, ISO-IEC FDIS 14 496-10*, 2003, Information Technology - Coding of audio-visual objects - Part 10: advanced video coding.

[3] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, June 2000.

[4] N. Färber, K. Stuhlmüller, and B. Girod, "Analysis of error propagation in hybrid video coding with application to error resilience," in *IEEE International Conference Image Processing*, Kobe, Japan, October 1999.

[5] I. E. G. Richardson, *Video Codec Design*, John Wiley & Sons, 2002.

[6] B. Girod, "What's wrong with mean-squared error?," *MIT Press Cambridge, MA, USA*, 1993.

[7] Z. Wang, H.R. Sheikh, and A.C. Bovik, *The Handbook of Video Databases: Design and Applications*, chapter 41: Objective video quality assessment, pp. 1041–1078, CRC Press, 2003.

[8] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion meausure," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 121–132, 2004.

[9] "The Video Quality Expert Group web site," http://www.its.bldrdoc.gov/vqeg.

[10] P. Corriveau and A. Webster, "Final report from the video quality experts group on the validation of objective models of video quality assessment, phase II," Tech. Rep., Video Quality Expert Group, July 2003.

[11] HR Wu and M. Yuen, "A generalized block-edge impairment metric for video coding," *IEEE Signal Processing Letters*, vol. 4, no. 11, pp. 317–320, 1997.

[12] Z. Wang, AC Bovik, and BL Evan, "Blind measurement of blocking artifacts in images," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, Canada, September 2000, vol. 3.

[13] P. Marziliano, F. Dufaux, S. Winkler, T. Ebrahimi, SA Genimedia, and S. Lausanne, "A no-reference perceptual blur metric," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, USA, September 2002, vol. 3.

[14] J. Caviedes, S. Gurbuz, P. Res, and NY Briarcliff Manor, "No-reference sharpness metric based on local edge kurtosis," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, USA, September 2002, vol. 3.

[15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structual similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, April 2004.

[16] M. Masry, S.S. Hemami, and Y. Sermadevi, "A scalable wavelet-based video distortion metric and applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 260–273, 2006.

[17] R. Bernardini, M. Naccari, R. Rinaldo, M. Tagliasacchi, S. Tubaro, and P. Zontone, "Rate allocation for robust video streaming based on distributed video coding," *Signal Processing: Image Communication*, vol. 23, no. 5, pp. 391–403, 2008.

[18] Q. Li and Z. Wang, "A no-reference perceptual blockiness metric," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, NV, USA, April 2008.

[19] T. Yamada, Y. Miyamoto, M. Serizawa, and H. Harasaki, "Reduced-reference based video quality-metrics using representative-luminance values," in *Third International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, AZ, USA, 2007.

[20] A. R. Reibman, V. A. Vaishmpayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 327–334, April 2004.

[21] T. Yamada, Y. Miyamoto, and M. Serizawa, "No-reference video quality estimation based on error-concealment effectiveness," in *IEEE Packet Video*, Lausanne, Switzerland, November 2007.

[22] A.A. Webster, C.T. Jones, M.H. Pinson, S.D. Voran, and S. Wolf, "An objective video quality assessment system based on human perception," *SPIE Human Vision, Visual Processing, and Digital Display IV*, vol. 1913, pp. 15–26, 1993.

[23] S. Wolf and M.H. Pinson, "Spatial-temporal distortion metric for in-service quality monitoring of any digital video system," *Proc. SPIE*, vol. 3845, pp. 266–277, 1999.

[24] O. Sugimoto, R. Kawada, M. Wada, and S. Matsumoto, "Objective measurement scheme for perceived picture quality degradation caused by MPEG encoding without any reference pictures," *Proc. SPIE*, vol. 4310, pp. 932, 2000.

[25] M.C.Q. Farias, S. Mitra, M. Carli, and A. Neri, "A comparison between an objective quality measure and the mean annoyance values of watermarked videos," in *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, USA, September 2002.

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

[27] Z. Wang and E.P. Simoncelli, "Reduced-reference image quality assessment using a wavelet-domain natural image statistic model," *Human Vision and Electronic Imaging X Conference, San Jose, CA, January*, pp. 17–20, 2005.

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

[29] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell System Technical Journal*, vol. 39, pp. 1253–1266, 1960.

[30] T.-K. Chua and D. C. Pheanis, "QoS evaluation of sender-based loss-recovery techniques for VoIP," *IEEE Netw.*, vol. 20, no. 6, pp. 14–22, December 2006.

[31] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.

[32] N. Färber B. Girod, "Feedback-based error control for mobile video transmission," *Proceedings of the IEEE*, vol. 87, pp. 1707–1723, October 1999.