

Delay-Efficient GOP Size Control Algorithm in Wyner-Ziv Video Coding

Imad Ahmad, Ziad Ahmad, Ibrahim Abou-Faycal

Department of Electrical and Computer Engineering, American University of Beirut
Beirut, Lebanon

ifa04@aub.edu.lb, zfa05@aub.edu.lb, iaf@alum.mit.edu

Abstract—In Wyner-Ziv Video Coding (WZVC) (a special form of distributed video coding) the heavy work of motion estimation is shifted to the decoder. Unlike conventional video systems, such a scheme allowed for new emerging technologies that require low-complexity encoders. A key factor to the performance of WZVC is the quality of the Side Information (SI) present at the decoder which heavily depends on the key frame separation. For this reason, the Group Of Pictures (GOP) size plays a significant role in the coding efficiency of the system. In this paper, we focus on the control of GOP size in transform-domain WZVC with feedback channel. We present a novel algorithm that performs online selection of the GOP size relying on past system behavior. Our proposed algorithm incurs minimal delays to the system making it suitable for real-time applications.

Keywords—Distributed Video Coding (DVC), Wyner-Ziv Video Coding (WZVC), Group Of Pictures (GOP).

I. INTRODUCTION

In conventional video coding, such as MPEG and H.26x, temporal redundancy is exploited on the encoder by the motion estimator. This task imposes complexity on the encoder while maintaining a simple decoder. Such a scheme has been widely adopted in applications such as video streaming and broadcasting where a video is encoded once and decoded several times. However, new technologies have emerged lately that require the perfect opposite distribution of complexity: simple encoder, but complex decoder. These technologies may include wireless video networks, mobile video cameras, and multi-camera surveillance systems. For this purpose, a new direction in the video compression community has appeared recently focusing on Distributed Video Coding (DVC). These systems are based on the Slepian-Wolf [1] and Wyner-Ziv [2] theorems that date back to the 1970s. In particular, Slepian and Wolf suggested that the minimum rate needed to encode two statistically dependent sources with joint encoding and decoding can be similarly achieved by independent encoding while joint decoding. Wyner and Ziv extended this result to include lossy sources with Side Information (SI) present at the decoder. The first practical implementations of DVC systems appeared in 2002 by Puri and Ramchandran [3] and Aaron *et al.* [4]. The latter system has been greatly considered in the literature and we base our study on the transform-domain case of its implementation [5]. This system is referred to as transform-domain Wyner-Ziv Video Coding (WZVC).

In WZVC, frames are grouped into key frames and Wyner-Ziv (WZ) frames. Key frames are intracoded while WZ frames are intercoded. The SI frame is generated at the decoder by performing motion estimation and compensation using

decoded key frames. This SI frame serves as a prediction of the WZ frame; for this reason, the quality of the SI has a great impact on the Rate-Distortion (RD) performance in WZVC. Consequently, the overall coding efficiency is directly related to the distance between the key frames or, equivalently, on the Group Of Pictures (GOP) size. The vast majority of the literature considers a rigid GOP size along the whole video. While this might serve well in videos where expected uniform motion activity is present, a variable GOP size is favored in many other applications where motion content is unknown beforehand.

In conventional video systems motion content can be exploited at the encoder. This feature of conventional video systems had always made GOP size control a straightforward problem. Unfortunately, this is not the case in WZVC. The encoder can no more explicitly exploit motion content with the strong constraint of having the GOP size control module at the encoder. In this sense, the designer should search for a method to extract any leakage of information to the encoder about motion content so that GOP size control can be realizable. In [6] a content adaptive Wyner-Ziv video codec is presented for controlling the GOP size. Exploiting motion content on the encoder is reconsidered but this time with average interpolation techniques. Specifically, motion activity metrics are calculated on the encoder to evaluate the temporal correlation in the video sequence and then hierarchical clustering is used to provide variable GOP sizes. However, we believe that the proposed mechanism is still computationally extensive on the encoder and introduces delays by the clustering process. Another adaptive algorithm for GOP size control is proposed in [7] for WZVC with feedback channel suppression; a different approach is presented to deal with the problem at hand. The mechanism proposed is driven by the mere idea of predicting the coming performance of any GOP size to be set as future choice. Although the delays introduced by evaluating the performance of each GOP size before encoding might be tolerable in codecs without feedback, it might not be desired otherwise. In this paper, we propose a new algorithm for GOP size control in WZVC with feedback channel. We assume that streams of video, as it is usually the case, contain shots that nearly maintain uniform motion characteristics. In this context, we rely on previous system behavior to different GOP sizes and accordingly allow the codec to adapt to the changes in the motion activity.

This paper is organized as follows: In Section II, we

describe the codec that we base our analysis on. In Section III we present the techniques we use to estimate the Peak Signal-to-Noise Ratio (PSNR) of the decoded frames at the encoder. We present our proposed algorithm in Section IV and discuss the experimental results in Section V. We conclude our paper in Section VI and motivate future work on the subject.

II. TRANSFORM-DOMAIN WYNER-ZIV VIDEO CODEC

We adopt in this paper the transform-domain Wyner-Ziv video codec presented in [5]. However, we use Low Density Parity Check (LDPC) codes instead of Turbo codes. Figure 1 illustrates the overall block diagram of the system. Particularly, the encoder divides the incoming frames into key frames and WZ frames. Key frames are intracoded using conventional intraframe coding. On the other hand, WZ frames are transformed using blockwise Discrete Cosine Transform (DCT) and the resulting coefficients are then quantized. Afterwards, bitplanes are extracted and coded using the LDPC encoder and only syndrome bits are sent to the decoder upon request over the feedback channel. At the decoder, an SI frame is generated by performing motion-compensated interpolation relying on decoded key frames and previously decoded WZ frames. The transform SI frame is fed into the LDPC decoder with Laplacian correlation noise model. The LDPC decoder requests syndrome bits until decoding is successful. Using the SI frame and the LDPC decoded frame, both in the transform domain, the transform WZ frame is reconstructed and then Inverse DCT (IDCT) is finally performed to reach the decoded frame.

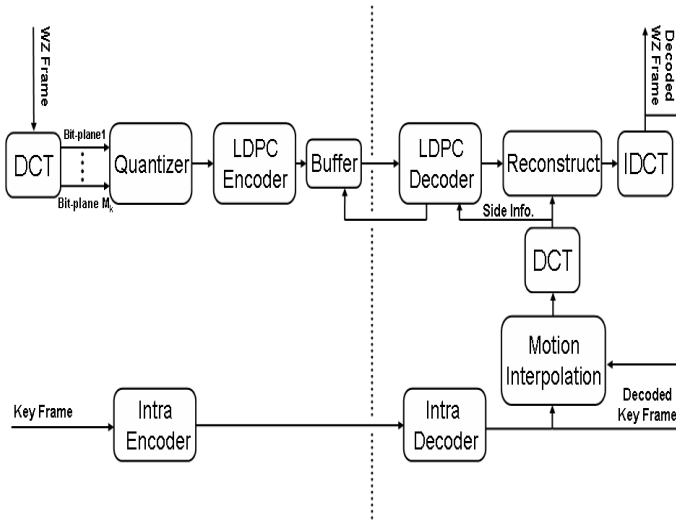


Fig. 1. Basic transform-domain Wyner-Ziv video codec architecture.

III. PSNR ESTIMATION AT THE ENCODER

The presence of the PSNR of the decoded frames at the encoder is essential to our algorithm presented later in Section IV. Fundamentally, to calculate this PSNR value, a copy of

the decoded frame should be available to the encoder. Clearly, this approach is unrealistic. For this reason, we discuss in this section special techniques that we use in order to estimate at the encoder the PSNR for both key frames and WZ frames.

A. Key Frame PSNR Estimation

To estimate the PSNR of a key frame at the encoder, we compute the distortion between the original and the decoded frame using the original frame only. In our codec, we employ motion JPEG for key frame intracoding and thus we use a simple method for computing the Mean-Square Error (MSE) as follows:

$$MSE^{KF} = \frac{1}{N} \sum_{i=0}^{N-1} (c_i^{KF} - \hat{c}_i^{KF})^2, \quad (1)$$

where N is the number of pixels in the frame, i indicates the index of the coefficient in the complete DCT matrix, and c_i^{KF} and \hat{c}_i^{KF} are the DCT coefficients of the original key frame before and after quantization, respectively. Note that we compute the \hat{c}_i^{KF} values using a midpoint reconstruction rule.

B. Wyner-Ziv Frame PSNR Estimation

Now we consider estimation of the PSNR for decoded WZ frames at the encoder. The target here is to find an estimate of the decoded WZ frame and consequently calculate the PSNR with the original frame. First, we generate a rough estimate of the SI frame at the encoder and perform a simple reconstruction rule. For estimating the SI frame, we emulate the interpolation technique performed at the decoder. However, we rely on averaging the frames, which is very simple on the encoder, and we use original frames in place of decoded frames. Next, we use the original WZ frame (after quantization) and the estimated SI frame to construct an estimated decoded frame. In particular, we use the following simple rule for the reconstruction process used at the decoder and introduced in [8]:

$$\tilde{c}_i^{WZF} = \begin{cases} LB_i & \text{if } c_i^{SI} < LB_i, \\ UB_i & \text{if } c_i^{SI} > UB_i, \\ c_i^{SI} & \text{otherwise,} \end{cases} \quad (2)$$

where c_i^{SI} and \tilde{c}_i^{WZF} are the DCT coefficients of the estimated SI frame and reconstructed WZ frame, respectively. LB_i and UB_i are the lower and upper bounds of the quantization bin corresponding to the coefficient at index i in the DCT matrix of the original WZ frame. Specifically,

$$LB_i = (c_i^{QWZF} - \frac{1}{2}) \times QSS_j$$

and

$$UB_i = (c_i^{QWZF} + \frac{1}{2}) \times QSS_j,$$

where c_i^{QWZF} is the quantized DCT coefficient of the original WZ frame at index i and QSS_j is its corresponding quantization step size at band j .

Finally, the MSE is calculated as follows:

$$MSE^{WZF} = \frac{1}{N} \sum_{i=0}^{N-1} (c_i^{WZF} - \tilde{c}_i^{WZF})^2, \quad (3)$$

where c_i^{WZF} is the DCT coefficient at index i of the original WZ frame.

IV. PROPOSED GOP SIZE CONTROL ALGORITHM

In this section we discuss the algorithm that we propose for controlling the GOP size in a transform-domain Wyner-Ziv video codec with feedback channel. The essence of our new idea is to base the future GOP size choice on previous system behavior to a collection of different sizes. We motivate the use of previous system performance by assuming video sequences that contain shots of roughly steady motion content. We see that such a scheme would avoid the complexity burden entailed by estimating the performance of all different GOP modes before encoding, in contrast to what is proposed in [7]. We also believe that our algorithm perfectly suits codecs with feedback channels because it reduces delays for two main reasons: (1) it allows for contiguous coding in principle and (2) uses an increasing window methodology in favor of the surviving GOP size to reduce the number of tests. Moreover, we note that the algorithm extracts the true, *not estimated*, coding rate present at the encoder for evaluating the performance. This advantage makes the size selection less dependent on estimation techniques.

In our proposed algorithm, the system is initiated by using a small set of size N of different GOP sizes, $GOP_0, GOP_1, \dots, GOP_i, \dots, GOP_{N-1}$, for coding as a test phase. The PSNR is estimated for the key frames and WZ frames for each GOP used in the test phase as discussed in Section III while the coding rate is easily computed given that it is known to the encoder. In order to compare the coding performance of the different GOP sizes in the test phase, average PSNR, \overline{PSNR} , and average coding rate, \overline{Rate} , are computed. The GOP size with the best performance (highest ratio) will be selected as future GOP size with a certain window size, w . We define the window size to be the number of times the surviving GOP size, GOP_{surv} , is used until another test phase is considered. To reduce the impact of decision errors size assignment is only increased when multiple tests yield the same decision. More precisely, if the test phase decides on the same GOP size that was selected previously the window size is doubled, otherwise it is reset to 1. The window size of the previous surviving GOP size should be stored and we refer to it as w_{old} . We also make sure that the new window size is bounded by a certain number of frames, $limit$, so that tests can still carry on.

Enumerated, the algorithm operates as follows:

- 1) Set $i = 0$. Initiate the test phase:
 - a) Code using GOP_i and estimate the PSNRs.
 - b) Compute the ratio $R_i = \frac{\overline{PSNR}}{\overline{Rate}}$.
 - c) Set $i = i + 1$. If $i < N$, return to step a).
 - d) $GOP_{surv} = \arg \max_{GOP_i} R_i$, for $0 \leq i < N$. If $GOP_{surv} = \text{previous } GOP_{surv}$, $w = w_{old} \times 2$, otherwise $w = 1$.
 - e) If $w \times GOP_{surv} > limit$, set $w = \lfloor \frac{limit}{GOP_{surv}} \rfloor$.
 - f) Set $w_{old} = w$.
- 2) Code using GOP_{surv} .
- 3) Set $w = w - 1$. If $w = 0$, go to step 1, otherwise go to step 2.

V. EXPERIMENTAL RESULTS

In order to test our proposed algorithm, we run it over the first 150 frames of the *Hallyway* and *Foreman* video sequences. We use the quantization matrix present in Annex K of the JPEG standard [9] with quality (scaling) factors $QF = \{0.5, 1, 2, 4\}$. For the test phase, we consider three different GOP sizes: 2, 3, and 4, successively. We set the window limit to 30 frames.

First we analyze the results of the *Hallyway* sequence. Figure 2 provides the GOP sizes that were set by our proposed algorithm along the sequence for both $QF = 0.5$ and $QF = 1$. Clearly, we can see that the algorithm dominantly chooses size 4 over the other two sizes. We interpret this as a detection of the uniform low motion activity that is present in this test sequence.

Table 1 shows the ratios of the estimated average PSNR to previous average coding rate for test phases 1, 4, and 9. Real ratios are also presented in this table in order to show that the estimation is sufficiently good for our purposes. As expected in a low activity sequence, the estimated ratios might be larger than the true ratios since original WZ frames are used in the estimation. However, we observe that the last argument does indeed affect the values in absolute but typically not in relative to other GOPs in the test phase specially when the values are significantly different.

Table 1. Ratios for the Hallway sequence for $QF = 2$ (* denotes survival GOP).

Test #	GOP	True R_i ($\frac{dB}{bpp}$)	Est. R_i ($\frac{dB}{bpp}$)
1	2	66.61	79.97
	3	82.63	104.53
	4	93.18*	120.21*
4	2	57.07	63.45
	3	64.74	73.73
	4	71.54*	82.34*
9	2	61.51	73.36
	3	74.47	91.73
	4	81.49*	100.99*

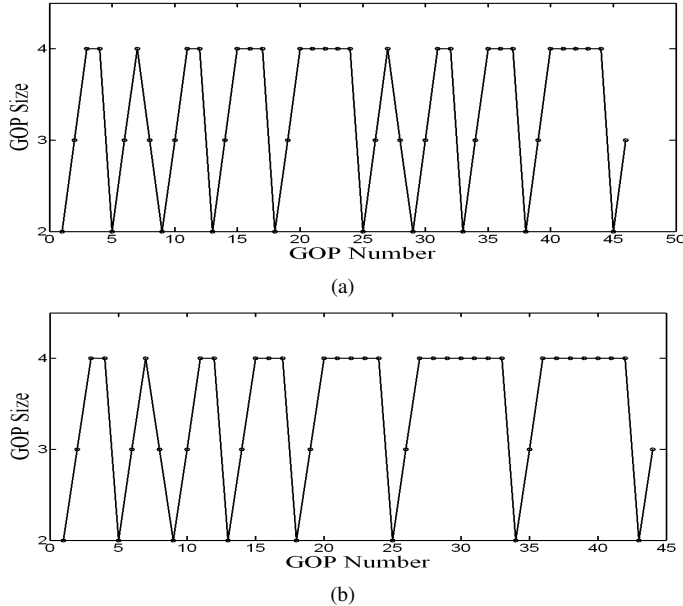


Fig. 2. GOP size variations by the proposed algorithm along the Hallway sequence (first 150 frames) for $QF = 0.5$ (a) and $QF = 4$ (b).

We now examine the RD curve in Figure 3 of the *Hallway* sequence. We notice that the rigid GOP size of 4 greatly outperforms the other rigid sizes. This explains why our algorithm chooses the GOP size of 4 most of the time. The RD curve of our algorithm is also presented on the same figure and we see that it approaches that of the best GOP size, size of 4.

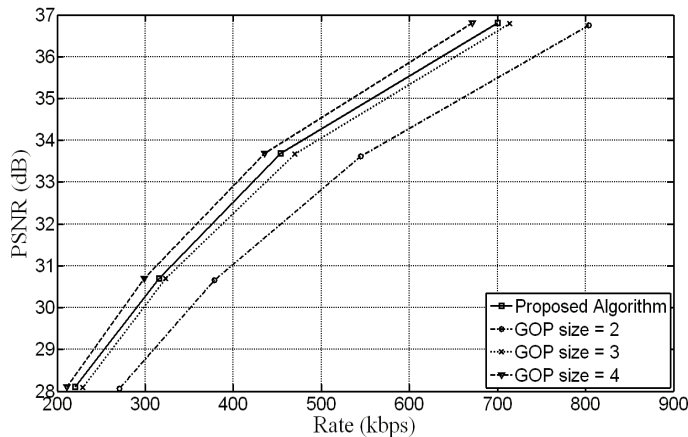


Fig. 3. Rate-Distortion curve for the Hallway sequence.

Next we turn our attention to the results of the *Foreman* sequence. In Table 2, we show the estimated and true ratios for different test phases to ensure that the decisions made by the algorithm are almost similar to when using the true ratios. We notice in Figure 4 that the GOP sizes chosen by the algorithm alternate between the three sizes. We explain this result by examining the RD curve in Figure 5. Although the GOP size

of 2 has rather minute gains at higher rates, the performance of each of the three GOP sizes is very close to one another. For this reason, our algorithm could not settle on a GOP size fit for this video sequence.

Table 2. Ratios for the Foreman sequence for $QF = 2$ (* denotes survival GOP).

Test #	GOP	True $R_i (\frac{dB}{bpp})$	Est. $R_i (\frac{dB}{bpp})$
1	2	51.63	56.94
	3	53.48*	59.78*
	4	50.95	57.53
4	2	48.51	52.89
	3	48.49	53.60
	4	53.50*	60.47*
9	2	56.10	63.73
	3	55.54	64.42
	4	61.52*	72.06*

Consequently, we believe that the choices of the GOP sizes in the test phase should be distant apart; the increment may be greater than 1. We pose such a solution to combat the cases where videos do not maintain a steady survival GOP size and to make the algorithm general. We assume the situation is flexible and does not impose the need for deciding on the strictly "best" GOP size. In this context, we discuss the behavior of the algorithm in these scenarios. This behavior can best be explained by categorizing motion content into three groups as such:

- High motion content that is enough for the algorithm to choose the smallest GOP size most of the time (i.e. needs a GOP size or vary between GOP sizes less than smallest, if possible.)
- Low motion content that is enough for the algorithm to choose the largest GOP size most of the time (i.e. needs a GOP size or vary between GOP sizes larger than the largest.)
- Motion content that matches any GOP size in the set and, if not steady, may vary along the whole set.

It is clear that if the sequence falls in the first or in the second category the algorithm works properly by detecting the best GOP size from the given set. Now, if the sequence falls in the third category the algorithm will not be as effective when successive GOP decisions are different, as in the case of the *Foreman* sequence. Nevertheless, the algorithm will outperform the worst rigid GOP size in all cases. On this ground, we see that the algorithm should perform well if we carefully choose a set of distant GOP sizes (e.g. a set of GOP sizes 2 and 5 will divide the video into parts of either high or low motion activity).

Before closing this section, we shall state important remarks pertaining to the delay reduction of the algorithm. First, we note that the mechanism is based on initiating the coding by a test phase and then deciding on the future GOP size. This offers the system continuous operation without having to put coding on hold to decide on the incoming GOP size. Moreover, we note that the algorithm does not add but a very small

number of computations to the encoder. These relate mostly to the PSNR estimation of the decoded frames that is performed only during the test phase. Add to that, deploying the idea of a growing window makes the process of GOP size selection less repeated whenever possible. As a conclusion, we see that all these factors contribute to mitigating the bogging effect of integrating a GOP size algorithm to Wyner-Ziv video coding systems.

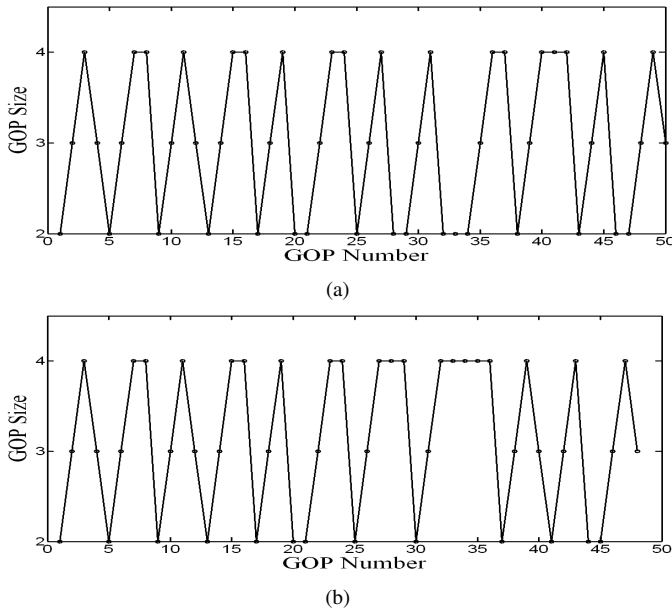


Fig. 4. GOP size variations by the proposed algorithm along the Foreman sequence (first 150 frames) for $QF = 0.5$ (a) and $QF = 4$ (b).

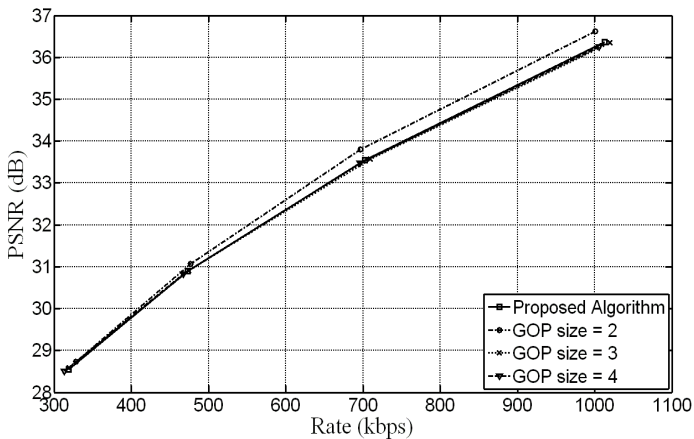


Fig. 5. Rate-Distortion curve for the Foreman sequence.

VI. CONCLUSIONS

In this paper, we have presented a novel delay-efficient GOP size control mechanism for Wyner-Ziv video coding with feedback channel. Our approach is especially beneficial when video streams possess shots that contain uniform motion activity. Our new algorithm controls GOP size relying on previous system behavior by deploying test phases while coding a given video. In addition, we adopted the concept of a window to decrease the number of test phases. As a future work, we aim at making the algorithm more flexible and robust to rapid variations in motion content. We also see that it is important to test and trigger our algorithm to suit specific video applications.

REFERENCES

- [1] J. Slepian and J. Wolf. "Noiseless coding of correlated information sources," *IEEE Trans. on Information Theory*, vol. 19, no. 4, pp. 471-480, 1973.
- [2] A. Wyner and J. Ziv. "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Information Theory*, vol. 22, no. 1, pp. 1-10, 1976.
- [3] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," in *Proc. Allerton Conf. on Communication, Control, and Computing*, Allerton, Illinois, USA, October 2002.
- [4] A. Aaron, B. Setton, and B. Girod, "Towards practical Wyner-Ziv Coding of video," in *Proc. IEEE Int. Conf. on Image Processing*, Atlanta, Georgia, USA, October 2006.
- [5] A. Aaron, B. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Proc. SPIE Int. Conf. on Visual Communications and Image Processing*, San Jose, California, USA, January 2004.
- [6] J. Ascenso, C. Brites, and F. Pereira, "Content adaptive Wyner-Ziv video coding driven by motion activity," in *Proc. IEEE Int. Conf. on Image Processing*, Barcelona, Spain, September 2003.
- [7] C. Yaacoub, J. Farah, and B. Pesquet-Popescu, "New Adaptive Algorithms for GOP Size Control with Return Channel Suppression in Wyner-Ziv Video Coding," *International Journal of Digital Multimedia Broadcasting*, vol. 2009, Article ID 319021, 11 pages, 2009.
- [8] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed Video Coding," in *Proc. IEEE: Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, pp. 71-83, January 2005.
- [9] ITU-T, I. JTC1, *Digital compression and coding of continuous-tone still images*, ISO/IEC 10918-1 ITU-T Recommendation T.81 (JPEG).