DISTRIBUTED VIDEO CODING WITH PARTICLE FILTERING FOR CORRELATION TRACKING

Lina Stankovic, Vladimir Stankovic, Shuang Wang and Samuel Cheng**

Dept. of Electronic and Electrical Engineering, University of Strathclyde
Royal College Building, G1 1XW, Glasgow, UK
*School of Electrical and Computer Engineering, University of Oklahoma
OK 74135, Tulsa, USA
phone: + (44) 1415482704, fax: + (44) 1415524968, email: lina.stankovic@eee.strath.ac.uk
Web: http://personal.strath.ac.uk/lina.stankovic

ABSTRACT

One of the main challenges of distributed video coding (DVC) is that correlation among source and side information needs to be estimated and well modelled a priori. Since in DVC correlation dynamically changes with the scene, in order to get the full benefit from powerful distributed source code (DSC) designs, predicting and tracking correlation is essential. This paper proposes an adaptive scheme based on integrated particle filtering within the LDPC-based DSC that dynamically tracks the changes in image correlation to enhance belief propagation LDPC decoding. The system maintains its low encoder complexity, showing significant performance improvement compared to the case without dynamic particle filtering tracking.

1. INTRODUCTION

Distributed Video Coding (DVC) or Wyner-Ziv video coding, is one of the earliest and most advanced applications of Distributed Source Coding (DSC) to date. DVC [1, 2] was proposed as a solution for emerging setups, such as video surveillance with tiny cameras and cell-to-cell communications, where low encoding complexity is a must. It brought a paradigm shift from the conventional centralized video coding architecture, where encoding complexity is much higher than decoding complexity, typical of traditional television broadcasting.

Wyner-Ziv (WZ) coding [3], where one source is available at the decoder as side information, is a lossy version of the Slepian-Wolf problem [4] with a distortion constraint. Slepian and Wolf considered lossless separate compression of two discrete sources, and showed that, roughly speaking, there is no performance loss compared to joint compression as long as joint decompression is performed. Wyner and Ziv showed that for a particular correlation where source and side information are jointly Gaussian, there is no performance loss due to the absence of side information at the encoder.

WZ source coding is usually realized by quantization followed by Slepian Wolf (SW) coding of quantization indices based on channel coding. Quantization is used to tune rate-distortion performance, while the SW coder is essentially a conditional entropy coder. The WZ decoder will thus comprise a SW decoder, which makes use of side information to recover the coded information. The SW decoder is followed by a minimum-distortion reconstruction of the source using side information.

Practical SW code design based on conventional channel codes is possible since correlation between the sources to be separately compressed is seen as a virtual communication channel, and as long as this virtual channel can be modeled by some standard communication channel, e.g., Gaussian, channel codes can be effectively employed. Designs [5] based on trellis-coded quantization followed by advanced channel coding, e.g., with turbo codes and low-density parity-check (LDPC) codes come very close to the bounds for two jointly Gaussian sources.

However, in much the same way as the information-theoretical DSC framework [3, 4], the state-of-the-art SW and WZ code designs based on turbo and LDPC codes perform well only when correlation statistics between sources are stationary and known at the encoders and decoder. To overcome this in case of non-stationary sources, in [6], an algorithm is proposed that performs adaptive SW decoding with decoder side information, using particle filtering integrated within an LDPC decoder based on belief propagation. The algorithm of [6] assumes discrete sources and binary symmetric channel correlation whose crossover probability p changes over time. Particle filtering and belief propagation are used to predict and track the change of p, to enhance decoding.

The problem of statistical correlation estimation between the source and side information is particularly important in DVC, since the scene dynamically and unpredictably changes. Thus in DVC we have a problem of WZ coding of non-stationary sources with unknown statistics. However, the designs (with few exceptions) usually simplify the problem by modelling correlation noise, i.e., the difference between the source and side information, as Gaussian or Laplacian random variable and estimate the parameters either based on training sequences or previously decoded data, which imposes certain loss especially for high-motion sequences. Non-stationarity of the scene has been dealt mainly by estimating correlation noise (e.g., on the pixel or block level) from previously decoded data and different initial reliability is assigned to different pixels based on the amount of noise estimated [7, 8, 9].

In this paper, we build on [6], to predict and track varying correlation between source and side information for WZ

 ISSN 2076-1465

decoding of video. Side information is generated using motion compensated interpolation, and belief-propagation decoding is enhanced with particle filtering-based correlation tracking mechanisms. Belief propagation and particle filtering are both performed iteratively on one generalized factor graph, where messages between different regions exchange their information to improve SW decoding. Thus in contrast to previous work, conventional belief propagation-based SW decoding and correlation statistics estimation are considered jointly.

For simplicity, the proposed design is tested on a pixel-based DVC [2] without a feedback channel. Similar decoders can be built for DCT-based designs, feedback channels, and including methods of [7, 8, 9] to start with.

The paper is organized as follows. In the next section, we outline previous methods for correlation estimation in DVC. We explain the concept of adaptive graph-based decoding incorporating particle filtering in Section 3. In Section 4, the proposed system setup is described, followed by experimental results in Section 5. The paper concludes with Section 6.

2. DVC AND CORRELATION MODELLING

DVC exploits the WZ coding principles to avoid computationally-expensive motion search operation at the encoder and shifting it to the decoder side. The fact that in WZ coding, the encoder does not need to know/use side information makes it possible to accomplish predictive coding without encoder motion compensation. In a nutshell, a block of pixels in the current frame is WZ encoded into a stream without any reference to previously encoded data, and the decoder uses all available information to generate a side information block that will be used to WZ decode the compressed stream.

State-of-the-art WZ coding designs based on turbo, LDPC, and trellis-based codes are successfully used for DVC (see [10, 11] and references therein).

Note that a key difference between WZ coding and DVC is that in the former statistics of the sources is known to both the encoder and decoder and does not change over time. In DVC, however, the decoder needs to generate side information using all information it has; regardless of how side information is generated, correlation noise statistics will be unknown and dynamically change over time. Indeed, due to the non-stationarity of real scenes, WZ coding in DVC has to deal with varying correlation statistics.

Estimating correlation statistics has been identified as a key challenge in DVC. Usually, correlation error is modeled as a Gaussian or Laplacian random variable whose parameters are estimated from previously decoded frames. It has been shown however that these models are not accurate enough if there are occluded regions in the scene [7]. In this case, the correlation noise is not white anymore and stationary, but instead would be concentrated in the occluded areas, which are usually at the boundaries of moving objects.

In [8], the correlation noise is always modeled as Laplacian, but to capture the non-stationary nature of the scene, the correlation parameter was varied from pixel to pixel. The noise power is increased if the pixel difference between motion compensated blocks in the two key frames used to generate side information is high; otherwise, it is decreased. The reasoning behind this method is that if the difference between the two key frames is high then we have less confidence in their average and the noise variance is higher. Thus, incorporating this model within SW decoding ensures that the channel code (employed for SW decoding) assigns higher reliability to pixels that have been predicted with higher accuracy, that is, the difference between the key frames is smaller.

Similarly, in [9], the Laplacian distribution is used with the parameter estimated online at the sequence, frame, block, and pixel level from decoded frames at the decoder.

Note that in [7, 8, 9], non-stationarity of a scene is addressed by changing correlation model on-the-fly and supplying the SW decoder with different initial reliability estimates. However, once SW decoding starts, (via, for example, belief propagation) the correlation is fixed.

Since the SW decoding process refines starting beliefs, unifying the process of correlation estimation and decoding into a single iterative process can provide better statistics estimate and consequently improved performance.

3. ADAPTIVE SLEPIAN-WOLF DECODING WITH PARTICLE FILTERING

In this section, we describe an adaptive algorithm for SW decoding via a factor graph, which enables efficient computation of marginal distributions via the belief propagation (BP) algorithm [12]. BP operates on the factor graph, which is a bipartite graph connecting variable nodes (representing unknown variables) and factor nodes. Messages are passed iteratively between connected variable nodes and factor nodes until the algorithm converges or a fixed number of iterations is reached. These messages will represent the influence that one variable has on another. In our case, messages will be inferences or beliefs on source bits and correlation. Thus, we group these types of messages into regions, thus generalizing the BP algorithm.

3.1 Graphical model

First, we construct the graph, then identify variable nodes and factor nodes, and how they are connected. For standard BP decoding of the SW code, the variable nodes are the coded bits and the factor nodes represent the connection among multiple variable nodes. N source bits $x \in \{x_1, x_2, ..., x_N\}$ are compressed by the SW encoder into syndrome bits $s \in \{s_A, s_B, ..., s_M\}$, where $M<N$. The syndrome factor node f_a, $a \in \{A, B, ..., M\}$ is connected to variable nodes x, taking into account constraints imposed by syndrome bits s.

In order to take into account the correlation between x and side information bits $y \in \{y_1, y_2, ..., y_N\}$, we additionally define correlation factor nodes f_i, $i \in \{1, 2, ..., N\}$ as:

$$f_i(x_i) = \begin{cases} 1-p, & \text{if } x_i = y_i \\ p, & \text{otherwise} \end{cases}. \qquad (1)$$

p is the estimated correlation between X and Y, i.e., Y can be considered as the output of X having passed through a Binary Symmetric Channel with crossover probability p. If p

is known a priori and is a constant over time, X can be compressed very close to the SW limit ($H(X|Y)$) using a syndrome-based approach and LDPC codes [13]. However in reality, p is rarely a constant over time and thus using an estimated a priori p for all X, will definitely not achieve performance close to the SW bound.

In order to allow for variability of p over time, variable nodes representing p_i, $i\in\{1, 2,...,N\}$ are added to the graph to connect to factor node f_i. The number of correlation factor nodes f_i connected to each p_i is referred to as the connection ratio. In general, it is usually enough to assume that p will vary slowly over time, and thus it is expected that adjacent variable nodes p_i will not differ much in value. This is represented in the graph by additional p-factor nodes $f_{i,j}$ that connect adjacent variable nodes p_i and p_j, as defined in (2), where λ is a hyper-prior and can be chosen rather arbitrarily.

$$f_{i,j}\left(p_i, p_j\right)=\frac{1}{\sqrt{2\pi\lambda}}\exp\left(-\frac{\left(p_j-p_i\right)^2}{2\lambda}\right). \quad (2)$$

The final constructed graph, illustrated in Fig. 1, comprises three regions: Region 3 is a standard Tanner graph for channel decoding, Region 1 is a bipartite graph capturing correlation factor p variability, and Region 2 connects Regions 1 and 3.

3.2 Message passing algorithm

In this section, message passing based on the BP algorithm between the 3 regions of the above graph is described. Standard BP (sum-product algorithm), generally used for channel decoding can handle only discrete variables. The crossover probabilities p, however, are not discrete, since they can change continuously over time. We therefore resort to particle filtering [14], which is integrated within the standard BP algorithm in order to handle continuous variables. Particle filtering (PF) estimates the a posteriori probability distribution of the correlation variable node p by sampling a list of random particles with associated weights.

The message passing schedule for the generalized factor graph in Fig. 1, incorporating PF, is as follows:

1. Initialise the values of N variable particles in Region 1, $p_i = \hat{p}$, for estimating the noise and a uniform weight *1/N*.
2. Initialise messages sent from factor nodes f_i in Region 2 to variable nodes x_i in Region 3 as in (1), where $p=\hat{p}$.
3. If the decoded estimate has the same syndrome as the received one or maximum number of iterations is reached, export the decoded codeword and finish. If not, go to Step 4.
4. Update variable nodes in Region 3 using standard BP (sum-product algorithm) for channel decoding.
5. Update particles in Region 1 by updating variable nodes using BP.
6. Compute the belief for each variable in Region 3 being $x_i \in \{0,1\}$.
7. Compute the belief (=weight) of each particle for each variable node in Region 1.

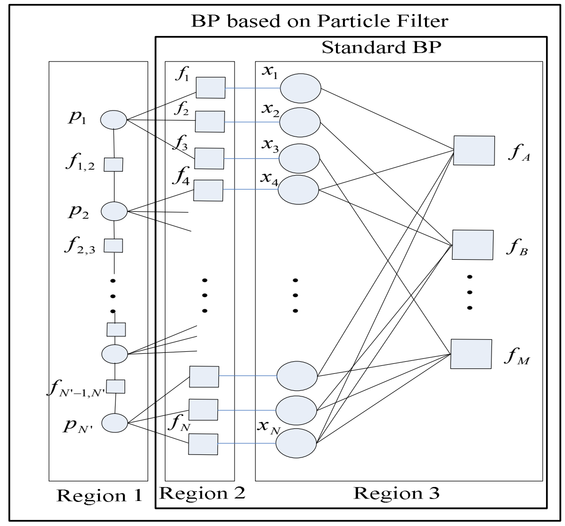

Fig. 1: Generalized factor graph representing constraints between syndromes in Region 3 and correlation constraints in Region 1.

8. Update factor nodes in Regions 3, 1 and 2 in that order.
9. Generate a new codeword based on the belief of variable nodes in Region 3.
10. Go back to Step 3.

Note that when the belief or weight for each particle for each variable node in Region 1 is updated (Step 7), a selection stage is carried out whereby particles with negligible weights are discarded while concentrating on particles with large weight. This is carried out via the systematic resampling algorithm [15]. However, after the resampling step, particles tend to congregate round values with large weight. In order to maintain diversity for future updating steps applying PF in Region 1, our algorithm perturbs the particles by applying the random walk Metropolis-Hastings algorithm, which essentially adds Gaussian noise on the current value p_i for each of the N particles. The weight of each particle is then reset to a uniform weight *1/N* for each particle.

4. SYSTEM SETUP

In order to demonstrate the benefit of correlation tracking in DVC, as a proof-of-concept, we implemented the following pixel-based DVC setup. As in [1, 9, 16], all frames are classified as key frames or Wyner-Ziv frames. Key frames are conventionally intra-coded; Wyner-Ziv frames are first quantized pixel-by-pixel using a uniform scalar quantizer to Q bits/pixel, and then additionally compressed using an LDPC-based SW coder which generates M syndrome bits, yielding an SW compression ratio of *M/Q*.

At the decoder side, side information frame Y is generated using motion-compensated interpolation of previous and future key frames [1,16]. Spatial smoothing [16], via vector-median filtering, is used to improve the result which is then quantized using the same scalar quantizer as at the encoder and is used as side information Y in SW decoding. Each frame is decoded by the adaptive LDPC-based SW decoder described in Section 3. SW decoding is followed by estimation and reconstruction [6].

Note that our implementation does not use a feedback channel, thus, rate control rests with the encoder. However, adding a feedback channel together with rate-compatible SW coding is possible in this set-up.

Our DVC decoder is flexible, providing the following tunable parameters: source and side-information quantization ratio, rate (compression ratio), hyper-prior λ, connection ratio between Regions 1 and 2, inclusion of Metropolis-Hastings random walk or not for perturbing particles following resampling, maximum number of iterations if BP fails to converge, number of particles to model *statistics*, and initial estimate of correlation $\hat{p}$.

5. EXPERIMENTAL RESULTS

In order to verify the effect of correlation tracking across WZ-encoded frames in a video sequence, we tested the above set-up with two video sequences, Car and Coast, with different scene dynamics. Both the average bit-error rate (BER) of the SW decoder and the PSNR of the reconstructed video sequence are calculated. These are plotted versus rate, which refers to the SW rate *M/Q*, i.e., a rate of 1 represents zero compression. For comparison, the benchmark performance of the DVC decoder (as described in Section 4) but without PF-aided correlation tracking, is also determined. In results plots below, solid lines always represent performance of our proposed DVC setup and dotted lines the benchmark performance. During our experiments, we observed that a starting $p=0.13$, $\lambda=0.1$, connection ratio=16 and using Metropolis-Hastings gave the best results.

Figs. 2 and 3 show the BER and PSNR performance, respectively, of Car video sequence for varying quantization levels Q. As expected, the BER for sequences quantized to Q=3 bits is better than for higher Q's and the waterfall section of the curves shifts to the right as the scene differences between X and Y increase, i.e., too many errors introduced by the virtual correlation channel that cannot be tracked. There is a clear improvement, for all Q's, in BER for sequences decoded by the generalized PF-based BP decoder compared to the classical BP decoder with no PF, e.g., enabling an extra 20% compression for the same BER of 10^{-6}.

BER improvement matches PSNR improvement. It is interesting to observe that while the BER improvement due to PF-based correlation tracking is more pronounced for lower Q, PSNR improvement due to PF-based correlation tracking is more pronounced for higher Q. The improvement of reconstructed video quality when using the generalized PF-based BP algorithm can be up to 3dB for this sequence, or 10% more compression for the same PSNR.

Fig. 4 shows a reconstructed frame from the Car sequence, where the proposed PF-based SW decoder removes the artefacts (in the centre of the frame) that could not be removed by the standard benchmark SW decoder.

Figs. 5 and 6 show the relative performance of correlation tracking versus standard WZ decoding for the Coast sequence for varying Q. Both BER and PSNR plots show similar trends to those of the Car sequence.

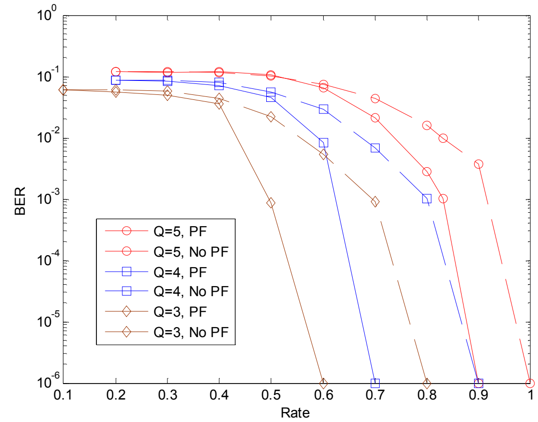

Fig. 2: BER of adaptive SW decoder for Car video sequence for varying number of quantization levels. Solid lines: proposed decoder, dotted lines: benchmark with no PF.

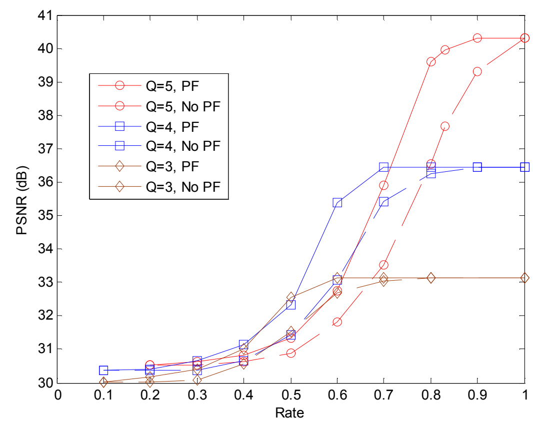

Fig. 3: PSNR for reconstructed Car sequence for varying number of quantization levels. Solid lines: proposed decoder, dotted lines: benchmark with no PF.

(a) (b)

Fig. 4: Reconstructed view of frame 7 of Car sequence for Q=3, rate 0.5 for: (a) benchmark system (b) proposed system.

However, as expected, due to the faster scene changes, the improvement in performance is less pronounced than for the Car sequence. Correlation tracking is robust enough to handle such scene changes and provides about 5% increase in compression for the same PSNR compared to the case of assuming a fixed *p*.

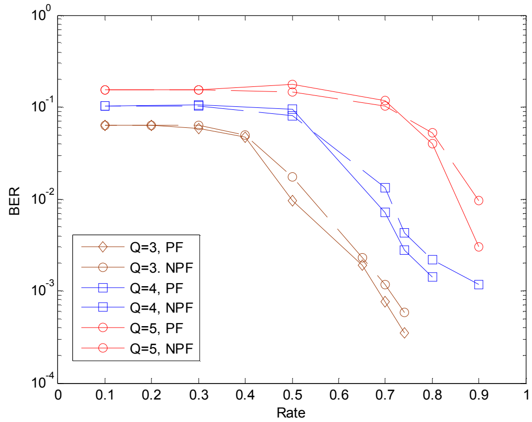

Fig. 4: Coast BER for varying quantization levels.

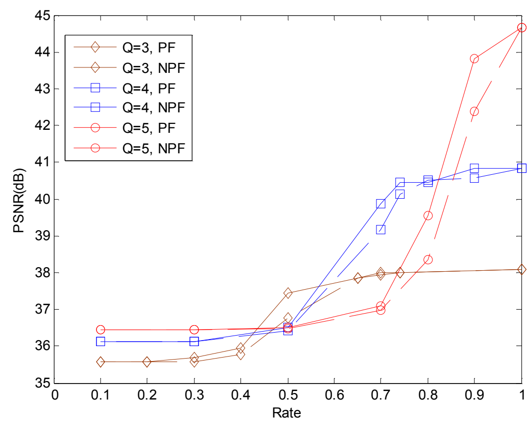

Fig. 5: Coast PSNR for varying quantization levels.

6. CONCLUSIONS

In this paper, we show that it is possible to improve performance of DVC through the use of an adaptive SW decoder incorporating particle filtering to track varying correlation between the source and side information. Our SW decoder is based on a generalized factor graph, comprising three regions: Region 3 which is the SW code bipartite graph decoded with standard belief-propagation algorithm, Region 1 which represents the variable correlation nodes updated via the particle filtering algorithm, and Region 2 which connects Regions 1 and 3. We demonstrate worthiness of this approach for correlation estimation by implementing a simple pixel-based DVC coder with an adaptive BP-based SW decoder incorporating particle filtering for tracking variability in real-time of pixel-by-pixel correlation. BER of SW decoding and PSNR results for two video sequences with varying scene dynamics clearly indicate the improvement due to the integration of particle filtering for correlation tracking within the BP-based LDPC SW decoder.

REFERENCES

[1] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in Proc. *40th Annual Asilomar Conf.*, Pacific Grove, CA, Nov. 2002.

[2] R. Puri and K. Ramchandran, "PRISM: A new robust video coding architecture based on distributed compression principles," in Proc. *40th Annual Allerton Conf.*, Allerton, IL, Oct. 2002.

[3] A.D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 1–10, Jan. 1976.

[4] D. Slepian and J.K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.

[5] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "On multiterminal source code designs," *IEEE Trans. Inform. Theory*, vol. 54, pp. 2278-2302, May 2008.

[6] S. Cheng, S. Wang, and L. Cui, "Adaptive Slepian-Wolf decoding using particle filtering based belief propagation," in Proc. *47th Annual Allerton Conf.*, Allerton, IL, 2009.

[7] P.F.A. Meyer, R.P. Westerlaken, R. Klein Gunnewiek, and R.L. Lagendijk, "Distributed source coding of video with non-stationary side information", in Proc. *VCIP-2005,* vol. 5960, pp. 857-866, Beijing, China, July 2005.

[8] M. Dalai, R. Leonardi, and F. Pereira, "Improving turbo codec integration in pixel-domain distributed video coding," in *Proc. IEEE ICASSP, 2006.*

[9] C. Brites and F. Pereira, "Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, pp. 1177-1190, Sept. 2008.

[10] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. of the IEEE*, vol. 93, pp. 71–83, Jan. 2005.

[11] C. Guillemot, et al., "Distributed monoview and multiview video coding: Basics, problems and recent advances," *IEEE Signal Proc. Magazine*, pp. 67-76, Sept. 2007.

[12] D. J. C. MacKay and R. M. Neal, "Near shannon limit performance of low density parity check codes," *IEE Electronics Letters*, vol. 32, no. 18, 1996.

[13] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Letters*, vol. 6, pp. 440-442, Oct. 2002.

[14] M. S. Arulampalam, S. Maskell, N. Gordon, T. Clapp, D. Sci, T. Organ, and S. A. Adelaide, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Proc.*, vol. 50, no. 2, pp. 174–188, 2002.

[15] M. Bolic, P. M. Djuric, and S. Hong, "Resampling algorithms for particle filters: A computational complexity perspective," *EURASIP Journal on Applied Signal Processing*, vol. 15, no. 5, pp. 2267–2277, 2004.

[16] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," Proc. *5th EURASIP Conf. Speech and Image Proc.*, Smolenice, Slovakia, July 2005.