MOTION ESTIMATION AT THE DECODER USING MAXIMUM LIKELIHOOD TECHNIQUES FOR DISTRIBUTED VIDEO CODING

Ivy H. Tseng and Antonio Ortega

Signal and Image Processing Institute
Department of Electrical Engineering - Systems
University of Southern California
Los Angeles,CA, 90089-2564
E-mail:{hsinyits, ortega}@sipi.usc.edu

ABSTRACT

Distributed video coding techniques have been proposed to support relatively "light" video encoding systems, where some of the encoder complexity is transfered to the decoder. In some of these systems, motion estimation is performed at the decoder to improve compression performance: a block in a previous frame has to be found that provides the correct side information to decode information in the current frame. In this paper we compare various techniques for motion estimation at the decoder that have been proposed in the literature and we propose a novel technique that exploits all the information available at the decoder using a maximum likelihood formulation. Our experiments show that likelihood techniques provide potential performance advantages when used in combination with some existing methods, in particular as they do not require additional rate overhead.

1. INTRODUCTION

With the increasing availability of mobile cameras and wireless video sensors, new computationally-demanding applications are being proposed for these mobile devices. For example, there is interest in allowing mobile device users to capture video clips and then share them with others by uploading them to a central server. While compression will be needed, due to bandwidth limitations, conventional predictive video coding techniques may be too complex for some of these devices, due to the motion estimation to be performed at the encoder. Thus, a need has emerged for novel video compression techniques that can achieve good performance with a computationally light encoder, possibly shifting some of the complexity to the decoder.

The Slepian-Wolf theorem provides a basic tool to achieve this goal. Let Y and X be two correlated sources. The theorem states that if the joint distribution of X and Y is known and Y is only available at the decoder, X can be encoded at the theoretically optimal rate $H(X|Y)$ [1]. A corresponding theorem for lossy source coding due to Wyner and Ziv [2] has lead to several proposals for practical Wyner-Ziv coding (WZC).

For video compression, we wish to encode pixel blocks in the current frame, which are likely to be correlated to blocks in the previous frame. As shown in [3], in a distributed coding setting, this can be seen as a scenario where the decoder will have access to multiple candidate side information blocks, and will have to decide which side information is best for decoding. Each of these candidate side information blocks are blocks to be found in the previous frame, so that the decoder is performing an operation similar to motion compensation. Since the information sent by the encoder cannot be decoded without side information, identifying the correct side information (i.e., the correct block) typically involves, for *each* candidate side information, (i) using the side information to decode what was transmitted, and (ii) determining whether the decoded data indicates that the side information was correct[1]. This second step can be achieved by letting the encoder send information that can be used to identify the correct side information. As an example, a hash function can be used for this purpose: the encoder will send the result of the hash function for a given block to the decoder, so that the decoder can determine if decoding is correct (i.e., if the hash value it generates matches that sent by the encoder). A more formal definition of the problem can be found in [3], which also provides a key insight: the number of bits needed (e.g., in the form of a hash) to identify the correct side information at the decoder is, under some simplifying assumptions, the same that would be needed if the encoder identified the best side information (e.g., via motion estimation) and sent the location of the corresponding block to the decoder.

Practical methods proposed to date to enable motion estimation at the decoder require some transmission rate overhead. Aaron, Zhong and Girod resort to feedback to deal with model uncertainty (see references in [4]). Additional parity bits are requested from the encoder if the decoder decides the decoding is not reliable. However, this approach leads to increases in decoding delay and it also requires a feedback channel. Aaron, Rane and Girod propose sending an additional hash function as a coarsely quantized and sampled original frame [4]. Puri and Ramchandran use cyclic redundancy checks (CRCs) to validate the correctness of the decoded blocks [5]. In addition to increasing the overall rate, CRC-based approaches can only be applied reliably to a limited range of rates, as will be discussed later in this paper.

Our goal in this paper is to design rate-efficient techniques that will enable the correct reference to be estimated at the decoder. We propose a novel estimation method based on maximum likelihood (ML). Our proposed technique can be seen to complement existing methods. We provide an analysis of our method and existing methods in order to address the trade-off between performance and rate overhead and also discuss the ranges of operating rates that are most suitable for each method.

[1]Note that this formulation does not preclude the encoder sending some motion information to the decoder; if accurate motion information is transmitted, then there will be a single candidate side information, while if only "rough" motion information is sent it will be used to reduce the number of candidate side information blocks.

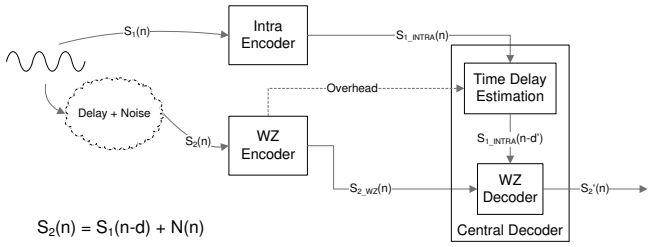

Fig. 1. Time Delay Estimation: The motivation and proof of using maximum likelihood method to search for the correct side information. d' is the estimated time delay.

As in [6], we have a situation where the decoder uses quantized data to estimate the time delay (or in this case the motion displacement) between two data sequences. While in [6] both sources were coded independently with standard quantizers (and thus could be decoded independently), here we show that this estimation can be done reliably, *even if one of the streams is coded using WZC*. The time delay estimation (TDE) problem serves as a motivation and proof of concept of the proposed technique, which in this paper is mostly proposed for decoder motion estimation.

This paper is organized as follows. We first introduce the problem in the context of TDE in Section 2. In Section 3 we propose the ML method and also review other methods used for TDE. In Section 4 the experimental results of TDE are given. We extend the ML method to a video coding environment in Section 5 and provide experimental results and analysis of different methods in Section 6.

2. REFERENCE FINDING AT DECODER IN THE CONTEXT OF TIME DELAY ESTIMATION (TDE)

Consider the scenario illustrated by Figure 1, where there are two sensor nodes, N_1 and N_2, that obtain readings S_1 and S_2 respectively. Here S_1 and S_2 are correlated in the sense that S_2 is a delayed noisy version of S_1: $S_2(n) = S_1(n-d)+N(n)$, where delay d and noise N are both unknown. N_1 sends S_{1_INTRA}, an encoded version of S_1 to the central node. S_{1_INTRA} is encoded independently and can be decoded without requiring any information from N_2; it will be used as side information to decode the information sent by N_2. N_2 has knowledge of the noise statistics (this could be a design parameter, or could have been learned via information exchange between the nodes). S_2 is encoded as S_{2_WZ} using distributed coding techniques based on this correlation and sent to the central decoder.

Since the delay d is unknown at the decoder, and correct decoding of S_{2_WZ} can only be guaranteed when S_{1_INTRA} delayed by d is used as side information, the central decoder will need to estimate the correct d. Note that this problem can be seen as the 1-D counterpart of the problem of motion estimation at the decoder for distributed video coding. Motion vectors represent spatial displacements, while for now we consider temporal displacements for the TDE problem. Also, the residual obtained by subtracting a predictor block (in the previous frame) from current block corresponds to the noise N for the TDE problem.

3. REFERENCE FINDING TECHNIQUES

3.1. Overhead-based Techniques

3.1.1. Correlation

Correlation-based techniques are commonly used for TDE. In our problem formulation, correlation cannot be computed directly at the decoder, since S_{2_WZ} can only be decoded correctly once d has been obtained. Thus, some of the information corresponding to reading S_2 needs to be coded independently so that it can be decoded without any side information from N_1. This method is similar to Aaron, Rane and Girod's hash function [4] in the sense that a subset of original information is sent to the decoder to help locate the correct reference.

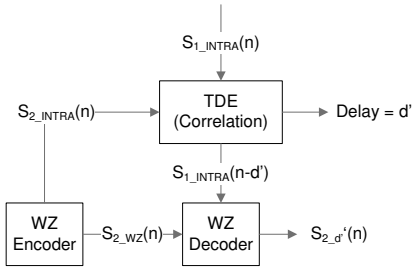

Fig. 2. Correlation: Subset of S_2 is intra encoded to correlate with S_1 to locate the correct reference

For every L-sample data block from S_2, we consider two simple approaches to convey intra-coded information. First we can include m consecutive samples encoded independently as a preamble in each L-sample block; the remaining $L - m$ samples are coded using WZC. We denote this method "preamble sample correlation" (PSC). The second approach would embed independently coded samples every k WZC coded symbols; this will be denoted "embedded sample correlation" (ESC). PSC usually performs better than ESC in terms of estimation accuracy. However, PSC has the drawback that it cannot detect a delay longer than m samples. Let the independently encoded samples in S_2 be denoted $S_{2\_INTRA}(n)$, where $n = 1 \cdots m$ for PSC and $n = 1, k \ldots mk$ for ESC. Let $S_{1\_INTRA}(n)$ be the intra encoded version of $S_1(n)$, $n = 1 \cdots L$. PSC and ESC pick d which maximizes

$$R(d) = \frac{1}{m} \sum_n S_{2\_INTRA}(n) S_{1\_INTRA}(n-d)$$

as the estimated delay.

3.1.2. Cyclic Redundancy Checks (CRCs)

Several proposed practical systems use CRCs, that sent to the decoder along with the WZC encoded data, in order to find the correct reference at the decoder [4][5]. As shown in Fig. 3, at the decoder, each possible delay is tested sequentially. $S_{2\_WZ}(n)$ is first decoded with $S_{1\_INTRA}(n-d')$ as the side information, where d' is one of possible delays. Then the decoder checks if the WZ decoded sequence $S_{2\_d'}(n)$ passes the CRC test. If it does, d' is declared to be the estimated delay; if it fails, $S_{2\_WZ}(n)$ is decoded with respect to the next possible delay.

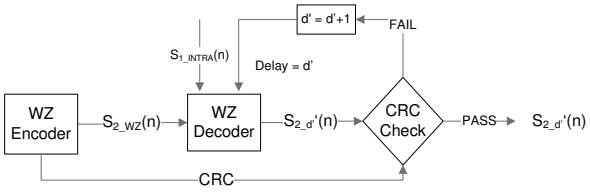

Fig. 3. Cyclic Redundancy Check: The validity of $S_{2\_d'}(n)$ is checked with a CRC test sequentially. $S_{2\_d'}(n)$ is the decoded $S_{2\_WZ}(n)$ with $S_{1\_INTRA}(n-d')$ as the side information

The major drawback of the CRC method is that its performance degrades outside of a certain range of block lengths. This is because even very few bit flips (say, just one bit) in a block of data lead to incorrect CRC values at the decoder. WZC techniques can be designed to limit the probability of decoding errors, but this probability is nonzero. Thus, as the block length increases, so does the probability that at least one sample will be decoded in error, *even if the correct delay, and thus side information, are being used*. Because of this, for longer blocks it becomes more likely that the CRC test will reject every possible candidate. With similar arguments, as the decoding error probability increases (possibly due to high SNR or limitation of transmission rate), the probability that the CRC test will reject every possible candidate also increases. Note also that a CRC test only provides pass/fail information, with no other ordering of the blocks. Thus, when all candidate delays fail the CRC check, the CRC provides no information to indicate which of the the blocks might be a more likely candidate.

To improve the CRC performance, one could partition one long block into n shorter ones so that each shorter block is sent with it own CRC (using shorter blocks could decrease the probability of being rejected by the CRC test for the correct delay). If the same length of CRC is used, the overhead will increase. Conversely, if a shorter CRC is used, the risk then would be that multiple *different* decoded blocks could all pass the CRC test. This again will pose the problem of selecting one among the multiple candidates that meet the condition, which cannot be done by using CRC provided information alone.

Also, in the case that one long block is partitioned, for each block, we retrieve a list of candidate delays, from which a single delay for all the blocks needs to be identified with some suitable rules. As an example, if two smaller blocks, P_1 and P_2, are used, we can determine that a correct delay is identified if both blocks provide consistent information, e.g., if there is only one candidate delay a that is valid for both P_1 and P_2 (we do not fully discuss all cases due to lack of space).

Note that in video applications CRCs may be applied to small data units, e.g., 8×8 pixel blocks or macroblocks, and thus the problems associated with long block lengths may not arise. However when multiple macroblocks share CRC information (in order to reduce the rate overhead) the above mentioned problems may arise and additional tools may be needed to supplement CRC information.

3.2. Maximum Likelihood Techniques

We now propose a novel technique to find the correct reference at the decoder using maximum likelihood estimation (ML). This method is based on the intuition that if we decode $S_{2\_WZ}(n)$ based on the correct side information $S_{1\_INTRA}(n-d')$, with the correct alignment d', the joint statistics of $S_{2\_d'}(n)$ and $S_{1\_INTRA}(n-d')$ should be similar to the original joint statistics of $S_2(n)$ and $S_1(n-d')$. Also, $S_{2\_d'}(n)$ and $S_{1\_INTRA}(n-d')$ should be similar. We define the likelihood that the delay is d' as:

$$L(Delay = d') = Pr(S_{2\_d'}(n)|S_{1\_INTRA}(n-d')),$$

where we apply the same probability model that was selected for the original data:

$$Pr(S_2(n)|S_1(n-d')),$$

i.e., conditional distribution of S_2 given S_1, which should be known at the encoder in order to enable efficient WZC. This likelihood model can be obtained in the training process, learned online or be given as an *a priori* design parameter. Our proposed ML approach involves first decoding $S_{2\_WZ}(n)$ with respect to all possible references (d'). Then for each $S_{2\_d'}$, the likelihood at every sample point is averaged. The decoded data with the highest average likelihood is then chosen as the decoded result.

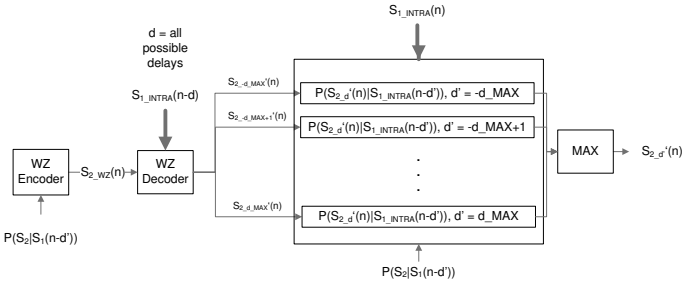

Fig. 4. Maximum Likelihood: $S_{2\_WZ}(n)$ is decoded with all the possible side information candidates. The joint probability model of S_1 and S_2, known at the encoder, is used to compute the likelihood of each decoded candidate value. The one with the maximum likelihood is then chosen as the result.

Block length influences the accuracy of TDE for the proposed method. Longer blocks include more WZC coded samples, and thus better estimation accuracy will be achieved. However, longer blocks will also introduce longer delay in decoding. Moreover, in the video case, displacement information changes locally (i.e., different blocks have different motion) and thus it is not practical in general to group together multiple blocks in order to improve likelihood estimation, as often the blocks will not share common motion.

4. EXPERIMENTAL TDE RESULTS

In our experiments a uniform 8-level scalar quantizer is used. We separate 8 quantization bins into 2 cosets and transmit only the LSB of each symbol as coset index. The maximum possible delay is $\pm$ 15 samples. Each experimental result is attained by at least 10,000 runs of Monte Carlo simulation and at least 10 errors occur for each point. In Fig. 5, we compare the TDE error probability for various mechanisms. The total rate is the total number of bits sent from both N_1 and N_2 for detection. First we note that our proposed ML approach outperforms all other methods, while being the only method that does not require any overhead. CRC works reasonably well but only in a relatively small range of block sizes. Outside of this range of rates its performance can degrade significantly. All the correlation methods require a certain amount of overhead, and their performance degrades as the ratio of overhead samples to total samples decreases.

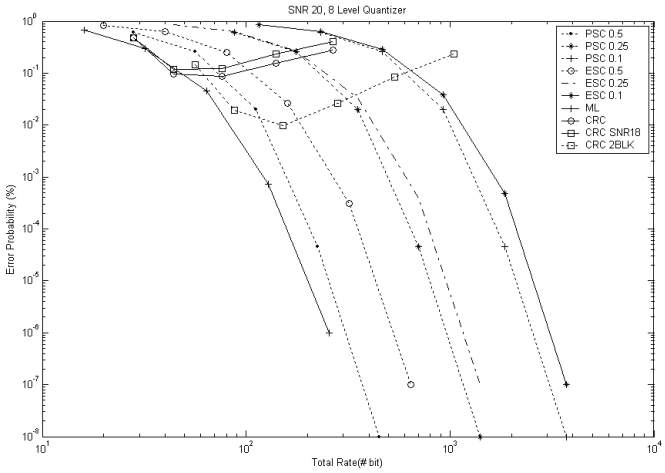

Fig. 5. Error probability of TDE by various methods at 20dB SNR. The number following PSC and ESC represents the ratio of independently encoded symbols to the block length. In CRC approaches, CRC-12 is used. "BLK2" refers to the 2-block case mentioned in Section 3.1.2

In summary these results show that the ML technique is a rate efficient and accurate method for TDE. This method relies, as does WZC in general, on some knowledge of the conditional statistics of S_1 and S_2, so that performance of both TDE and WZC will degrade when there is mismatch between the noise statistics assumed in the design and the actual noise affecting the measurements.

5. MAXIMUM LIKELIHOOD TECHNIQUES FOR MOTION ESTIMATION AT THE DECODER

We now extend our proposed method to the video case. For this we use a transform domain distributed video coding architecutre, as shown in Fig. 6. This is a simplified version of the PRISM system [5]. Instead of aligning two signal streams, here we are trying to find the best predictor block from the previous frame that enables correct decoding of the current macroblock. Let C_i be the current macroblock, C_{i_WZ} be the WZ coded version of C_i, $B_j,\ j = 1 \cdots M$ be all the possible candidate blocks in the previous frame, and C'_{i_j} be decoded C_{i_WZ} when B_j is used as side information. The likelihood of B_j being the best predictor is

$$L(B_j) = Pr(C'_{i\_j}|B_j)$$

The same probability model $Pr(C_i|B_{Ti})$, where B_{Ti} is the true predictor for C_i, for the original data is applied. This model can be generated through training.

Here we assume that DCT coefficients are independent, i.e.

$$Pr(C_i|B_{Ti}) = \prod_k^D Pr(C_i^k|B_{Ti}^k),$$

where C_i^k represents the kth DCT coefficient in block C_i and likewise for B_{Ti}^k. D is the number of DCT coefficients. Also, for transform domain coding, DCT coefficients are quantized before transmission, so here we consider the probability of the difference between the quantized DCT coefficients from the current block and the predictor block.

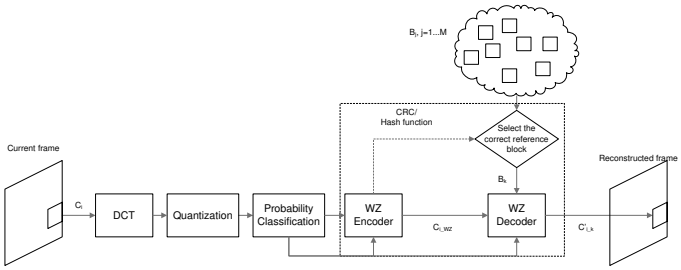

Fig. 6. The architecture of transform domain distributed video coding: A simplified version of PRISM [5]

$$Pr(C_i|B_{Ti}) = \prod_k^D Pr(C_i^k|B_{Ti}^k) = \prod_k^D Pr(Q(C_i^k)|Q(B_{Ti}^k))$$
$$= \prod_k^D Pr_k(\left|Q(C_i^k) - Q(B_{Ti}^k)\right|)$$

Pr_k is the pmf for the kth DCT coefficient.

Although the TDE problem and the motion estimation problem are very closely related, the motion estimation problem differs from the TDE problem in two major aspects. First, in the TDE case, determining the delay between the two sources is more important (as in many cases TDE is used in the context of source localization). Instead, in video coding problems, reconstruction quality is more important (and the accuracy of the estimated motion itself is not as important, as long as good quality can be achieved at the decoder). Thus in our evaluation for the video case we use PSNR, rather than probability of error, as the performance metric in the video case. Second, unlike in Section 4, where we know the exact probability model, in the video case the joint statistics of C_i and B_j are usually hard to estimate and may change over time. Thus, while in Section 3.2 ML method could perform reliable TDE solely based on WZC information, here sending extra intra information is sometimes needed to help locate the best predictor and improve the reconstruction quality. The extra intra information we select to send in our experiments is the DC value of the macroblock. The reason to chose the DC value is because the DC coefficient influences PSNR the most among all frequency coefficients.

6. EXPERIMENTAL RESULTS

In our experiments, 8×8 DCT is used. A different uniform scalar quantizer is used for each of the 64 DCT coefficients (we use the MPEG-1 intra mode quantization table). WZC scheme is coset coding. Two macroblock sizes (8×8 and 16×16) are tested. Test video sequences include "Foreman" and "Hall monitor". Experiments are done on the 11th to 12th frames of the sequence. For the ML method, the probability model is trained on the 1st to 10th frames of the sequence and the true predictor blocks are those with minimum SAD. For the CRC method, the length of CRC code is 12. All the methods are compared with the optimal case, i.e., that where the motion vectors are computed at the encoder and are transmitted to the decoder. The DC coefficients are sent as intra information. The experiment results are shown in Figs. 8 and 9.

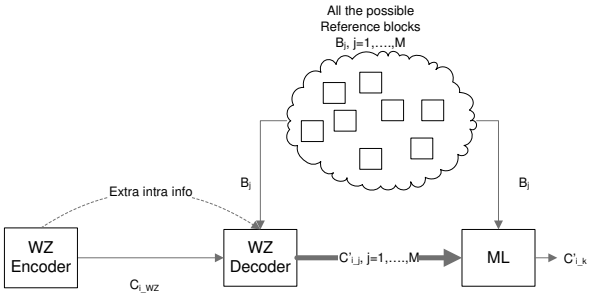

Fig. 7. Due to the lack of knowledge of the exact probability model in the video case, extra intra information may be needed to help locate the best predictor and improve the reconstruction quality.

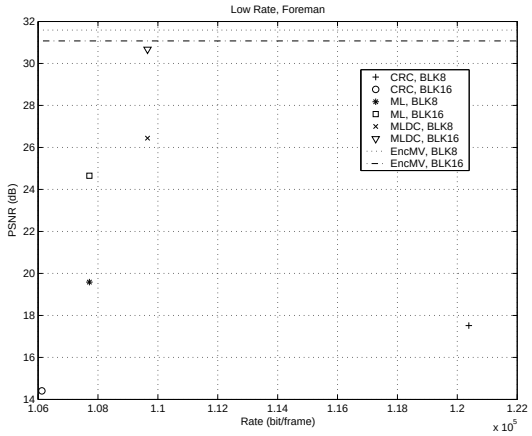

Fig. 8. *Foreman*: PSNR of various methods in low rate. The number of cosets for AC coefficients is 2. The number of cosets for DC coefficients is 2 for CRC and 32 for ML (while the CRC method spends bits on sending the CRC code, ML spends bits to have higher precision of the DC coefficients). The DC coefficients for MLDC is sent in intra mode (8 bits precision). "EncMV" represents the optimal case.

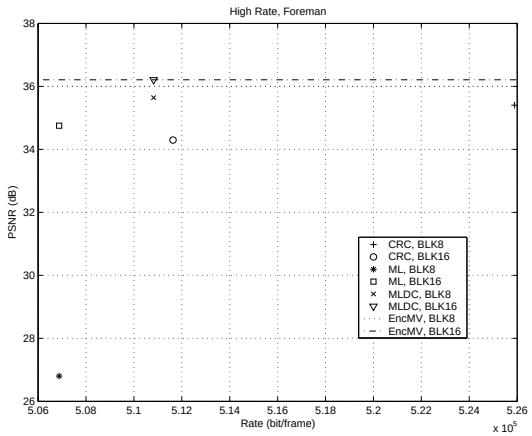

Fig. 9. *Foreman*: PSNR of various methods in higher rate. The number of cosets for AC coefficients is 32. The rest of the experimental settings are the same as those of Fig. 8.

First the experiment results verified the limitation of CRC discussed in Section 3.1.2. When the decoding error probability is high, which translates to a smaller number of cosets and lower rate in our experiment setup, CRC fails to locate side information. The poor performance of CRC, as seen in Fig. 8, is due to the fact that not all the blocks could find a corresponding side information. When the CRC test rejects all candidate side information information blocks, we cannot successfully decode the frame in its entirety and the then PSNR presented here is computed assuming that every pixel of the block which is unable to decode is set to 127. On the other hand, with similar transmission rate, ML could in general give better results. In particular, when bigger macroblocks are used (16×16 pixels) and DC is sent as intra information, ML provides performance close to the optimum level.

We also show, in Fig. 9, that when the decoding error probability is lower (i.e., the number of cosets is bigger and the rate is higher), both CRC and ML can perform well. When bigger macroblock sizes and/or DC intra information are used, ML outperforms CRC. Due to lack of space, we do not include the results for "Hall monitor", but the results are similar to those for "Foreman". When longer test sets are used, the performance degrades; this indicates that in order to use ML techniques in practice it would be necessary to use adaptive probabilistic models.

Also from Fig. 8 and Fig. 9, we can see that under the same parameter settings, bigger block sizes lead to better reconstruction for the ML method. This is consistent with the results of Section 4. In cases where CRC techniques fail to identify the correct side information, ML could be a reliable alternative method provided bigger block sizes can be used and the DC helper information is sent. In this case ML can perform reliable motion estimation at the decoder with slightly lower rate than CRC.

7. REFERENCES

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

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

[3] P. Ishwar, V. M. Prabhakaran, and K. Ramchandran, "Towards a theory for video coding using distributed compression principles," in *Proceeding of IEEE Conference on Image Processing*, Barcelona, Spain, September 2003.

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

[5] R. Puri and K. Ramchandran, "Prism: A "reversed" multimedia coding paradigm," in *Proceeding of IEEE Conference on Image Processing*, Barcelona, Spain, September 2003.

[6] L. Vasudevan, A. Ortega, and U. Mitra, "Application-specific compression for time delay estimation in sensor networks," in *First ACM Conference on Embedded Networked Sensors*, Los Angeles, CA, November 2003, ACM Sensys.