Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2009, Article ID 171257, 11 pages
doi:10.1155/2009/171257

Research Article

Side-Information Generation for Temporally and Spatially Scalable Wyner-Ziv Codecs

Bruno Macchiavello,[1] Fernanda Brandi,[1] Eduardo Peixoto,[1] Ricardo L. de Queiroz,[1] and Debargha Mukherjee[2]

[1] *Departamento de Engenharia Elétrica, Universidade de Brasilia, 70.910-900 Brasilia, DF, Brazil*
[2] *Hewlett Packard Labs, Palo Alto, CA 94304, USA*

Correspondence should be addressed to Bruno Macchiavello, bruno@image.unb.br

Received 1 May 2008; Revised 8 October 2008; Accepted 15 January 2009

Recommended by Frederic Dufaux

The distributed video coding paradigm enables video codecs to operate with reversed complexity, in which the complexity is shifted from the encoder toward the decoder. Its performance is heavily dependent on the quality of the side information generated by motio estimation at the decoder. We compare the rate-distortion performance of different side-information estimators, for both temporally and spatially scalable Wyner-Ziv codecs. For the temporally scalable codec we compared an established method with a new algorithm that uses a linear-motion model to produce side-information. As a continuation of previous works, in this paper, we propose to use a super-resolution method to upsample the nonkey frame, for the spatial scalable codec, using the key frames as reference. We verify the performance of the spatial scalable WZ coding using the state-of-the-art video coding standard H.264/AVC.

1. Introduction

The paradigm of distributed source coding (DSC) is based on two information theory results: the theorems by Slepian and Wolf [1] and Wyner and Ziv [2] for lossless and lossy codings of correlated sources, respectively. It has recently become the focus of different video coding schemes [3–12]. A review on DSC applied to video coding, that is, distributed video coding (DVC) can be found elsewhere [13]. Even though it is believed that a DVC algorithm will never outperform conventional video schemes in rate-distortion performance [13], DVC is a promising tool in creating reversed complexity codecs for power-constrained devices. Currently, digital video standards are based on predictive interframe coding and discrete cosine block transform. In those, the encoder typically has high complexity [14] mainly due to the need for mode search and motion estimation in finding the best predictor. Nevertheless, the decoder complexity is low. On the other hand, DVC enables reversed complexity codecs, where the decoder is more complex than the encoder. This scheme fits the scenario where real-time encoding is required in a limited-power environment, such as mobile hand-held devices.

A common DVC architecture is a transform domain Wyner-Ziv codec [4, 13], where periodic key frames are encoded with a conventional intraframe encoder, while the rest of the frames—called Wyner-Ziv (WZ) frames—are encoded with a channel coding technique, after applying the discrete cosine transform and quantization. This codec can be seen as DVC with temporal scalability because the WZ-encoded frames can represent a temporal enhancement layer. At the decoder, the key frames are used to generate prediction of the current WZ frame, called side information (SI), which is fed to the channel decoder. The SI for the current WZ frames can be generated using motion analysis on neighboring key and previously decoded WZ frames, thus exploring temporal correlations. As in much of the prior work [4, 11, 15], this architecture uses a feedback channel to implement a Slepian-Wolf codec.

A different approach is a mixed resolution framework that can be implemented as an optional coding mode in any existing video codec standard, as proposed in previous

works [16–18]. In that framework, the encoding complexity is reduced by lower resolution encoding, while the residue is WZ encoded. That spatially scalable framework does not use a feedback channel and considers more realistic usage scenarios for video communication using mobile power-constrained devices. First, it is not necessary for the video encoder to always operate in a reversed complexity mode. Thus, this mode may be turned on only when available battery power drops. Second, while complexity reduction is important, it should not be achieved at a substantial cost in bandwidth. Hence, the complexity reduction target may be reduced in the interest of a better rate-distortion trade-off. Third, since the video communicated from one mobile device may be received and played back in real time on another mobile device, the decoder in a mobile device must support a mode of operation where at least a low-quality version of the reversed complexity bit stream can be decoded and played back immediately, with low complexity. Off-line processing may be carried out for retrieving the higher quality version. In the temporal scalability approach, the only way to achieve this is to drop the WZ frames, resulting in unnecessarily low frame rates.

It is well known that the performance of those or any other WZ codec is heavily dependent on the quality of the SI generated at the decoder. In this work, we compare the performance and complexity reduction of different SI estimators. For a temporally scalable codec, we introduce a new SI generator that models the motion between two key frames, in order to predict the motion among key frames and a WZ frame. We compare our results with a common SI estimator for a WZ codec with temporal scalability [19, 20], such a codec tries to model the motion vectors of the current WZ frame using the next and previous decoded key frames.

A more accurate SI generator for a DVC codec with temporal scalability was presented elsewhere [21, 22]. There the SI generator uses forward and bidirectional motion estimation, motion vector refinement, spatial motion smoothing techniques, and it adapt the motion vector to fit into the frame grid. Compared to that technique [21, 22], the SI generation proposed in this paper is less complex, and does not modify the reference or the motion vector. Nevertheless, it is less efficient, being outperformed by the more complex algorithm [21, 22]. However, similar tools like spatial motion smoothing and motion vector refinement, as described elsewhere [21, 22], can be used along with the proposed technique in order to increase the overall performance at the cost of a more complex decoding. For the mixed resolution framework, we improve SI generation, as a continuation of previous work [18]. This method is based on superresolution using key frames [23]. The main idea is to restore the high-frequency information from an interpolated block of the low-resolution encoded frame. This SI generation can be done iteratively, using the SI generated from a previous iteration to improve the quality of the current frame being generated. Other works have used iterative SI generation techniques [24–26]. All of them assume key frames are intracoded, and the intermediate frames are entirely WZ coded. In [25], previously decoded bit planes are used to improve SI. In [24], a motion-based algorithm is presented, however the SI is generated by aggressively replacing low-resolution (LR) blocks by blocks from the key frames.

Here, the rate-distortion (RD) performance of the proposed SI generation methods along with a coding time comparison is presented. We also present the RD performance of the spatial scalable coder and compare it to conventional coding. Such a coder is based in previous studies for optimal coding parameter selection [27] and correlated statistic estimation [28]. The results of the temporal scalable coder in the transform domain are known, and normally outperform simple intracoding, but underperforms zero-motion vector coding, depending on the sequence [29]. The entire tests were implemented using the state-of-the-art standard H.264/AVC as the conventional codec.

The paper is organized as follows; the WZ architectures are described in Section 2. In Section 3, the different schemes for generation of the side information are detailed, and in Section 4 simulation results are presented. Finally, Section 5 contains the conclusions of this work.

2. Wyner-Ziv Coding Architectures

In order to compare SI generation methods, we consider two different Wyner-Ziv coding architectures: a transform domain Wyner-Ziv (TDWZ) codec and a spatially scalable Wyner-Ziv (SSWZ) codec.

2.1. Transform Domain Wyner-Ziv Codec. The TDWZ codec architecture [4, 13] allows for temporal scalability. At the encoder only some frames, denoted as key frames, are conventionally encoded, while the rest are entirely WZ coded. At the decoder, the key frames can be instantly decoded by a conventional decoder, while the WZ layer can be optionally used to increase the temporal resolution of the sequence. The architecture is shown in Figure 1. The WZ frames are coded by applying a discrete cosine transform (DCT), whose coefficients are quantized, sliced into bit planes and sent to a Slepian-Wolf coder. Typically, the Slepian-Wolf coder is implemented using turbo codes or LDPC codes, where only the parity bits are stored in a buffer. The code is punctured and bits are transmitted in small amounts upon a decoder request, via the feedback channel.

Complexity reduction is initially obtained with temporal downsampling, since only the key frames are conventionally encoded. However, if the key frames were to be encoded as *I*-frames, a more significant complexity reduction can be achieved, since there will be no motion estimation at the encoder side. Note that if the key frames are selected as the reference frames and the WZ frames are the nonreference frames, then the key frames can be coded as conventional *I*-, *P*-, or reference *B*-frames, without drifting errors. This not only increases the performance in terms of *RD*, but also increases the complexity since motion estimation may be used for the key frames as well.

At the decoder, the SI generator uses stored key frames in order to create its best estimate for the missing WZ frames. Motion estimation and temporal interpolation techniques are typically used. Typically, the previous and next key frames

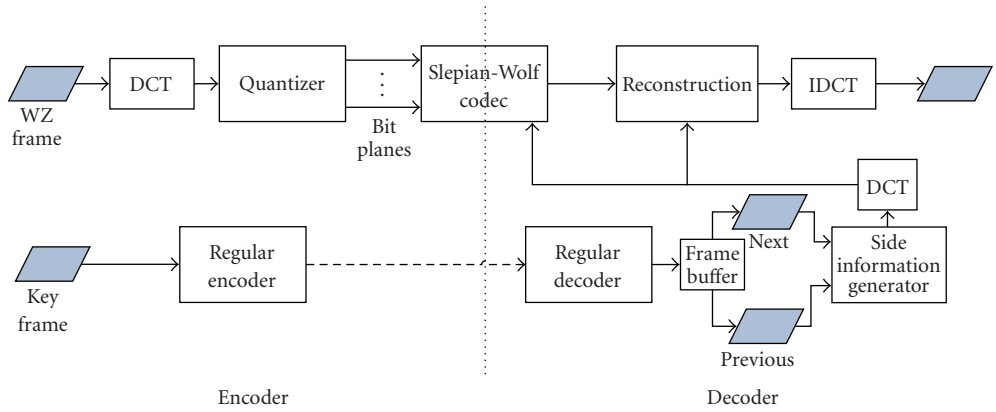


Figure 1: Transform domain Wyner-Ziv codec architecture.

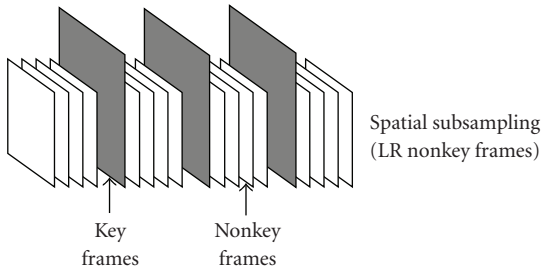


Figure 2: Illustration of key frames in spatial scalable video.

of the current WZ frame are used for SI generation, although some works use two previously decoded frames [30]. This SI is used for channel decoding and frame reconstruction in the decoding process of the WZ frame. A better SI means fewer errors, thus requesting fewer bits from the encoder. Therefore, the bit rate may be reduced for the same quality. Hence, a more accurate SI can potentially yield a better performance of the TDWZ codec.

2.2. Spatially Scalable Wyner-Ziv Codec. The mixed resolution framework [16–18] used by the SSWZ codec can be implemented as an optional coding mode in any existing video codec standard (results using H.263+ can also be found in previous works [16–18, 28]).

In that framework, the reference frames (key frames) are encoded exactly as in a conventional codec as *I*-, *P*- or reference *B*-frames, at full resolution. For the nonreference *P*- or *B*-frames, called nonreference WZ frames or nonkey frames, the encoding complexity is reduced by LR encoding, as illustrated in Figure 2.

The architecture of a SSWZ encoder is shown in Figure 3. The nonreference frames (WZ frames) are decimated and encoded using decimated versions of the reconstructed reference frames in the frame store. Then, the Laplacian residual, obtained by taking the difference between the original frame and an interpolated version of the LR layer reconstruction, is WZ coded to form the enhancement layer. Since the reference frames are conventionally coded, there are no drift errors. The number of nonreference frames and the decimation factor may be dynamically varied based on the complexity reduction target.

At the decoder (Figure 4), high-quality versions of the nonreference frames are generated by a multiframe motion-based mixed superresolution mechanism [18]. The interpolated LR reconstruction is subtracted from this frame to obtain the side information Laplacian residual frame. Thereafter, the WZ layer is channel decoded to obtain the final reconstruction. Note that for encoding and decoding the LR frame, all reference frames in the frame store and their syntax elements are first scaled to fit the lower resolution of nonreference LR coded frame. The channel code used is based on memoryless cosets. A study for optimal coding parameter selection for coset creation can be found elsewhere [17, 27, 28]. There, a mechanism to estimate the correlated statistics from the coded sources is described.

3. Side-Information Generation

In this section, we detail two different methods for side information generation. The first technique generates a temporal interpolation of a frame for a TDWZ codec, being significantly different from previous SI generation algorithms. In the SE-B algorithm [19, 20] the motion vectors, obtained from bidirectional motion estimation between the previous and next key-frames, are halved. Then, motion compensation is done by changing the reference block (see Figure 5). Other methods [21, 22] adapts the motion vectors to fit into the grid of the SI frames, to avoid blanks and overlaps areas. The proposed technique keeps both the reference and the motion vector, using a simple technique to deal with overlaps and blanks areas.

The second SI generation method proposed in this work creates a superresolved version of a LR frame for a SSWZ codec. This new method outperforms previous works [18].

3.1. Motion-Modeling Side-Information Estimator. The proposed method models the motion between two key frames F_{n-1} and F_{n+1} as linear. Thus, the motion between F_{n-1} and the current frame F_n is assumed to be half of the motion

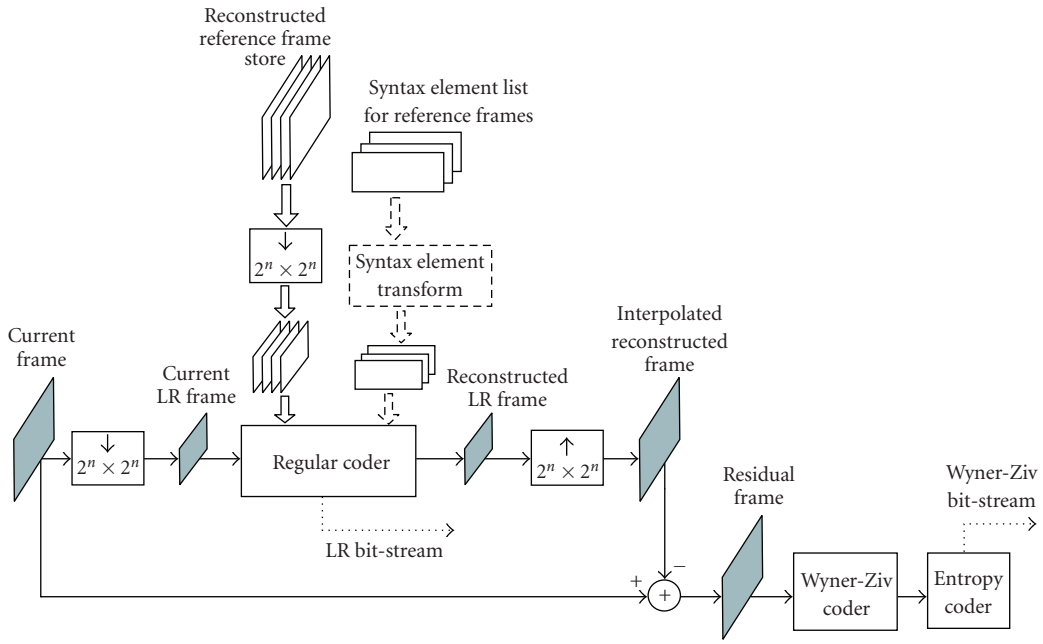


Figure 3: Encoder of the WZ-mixed resolution framework.

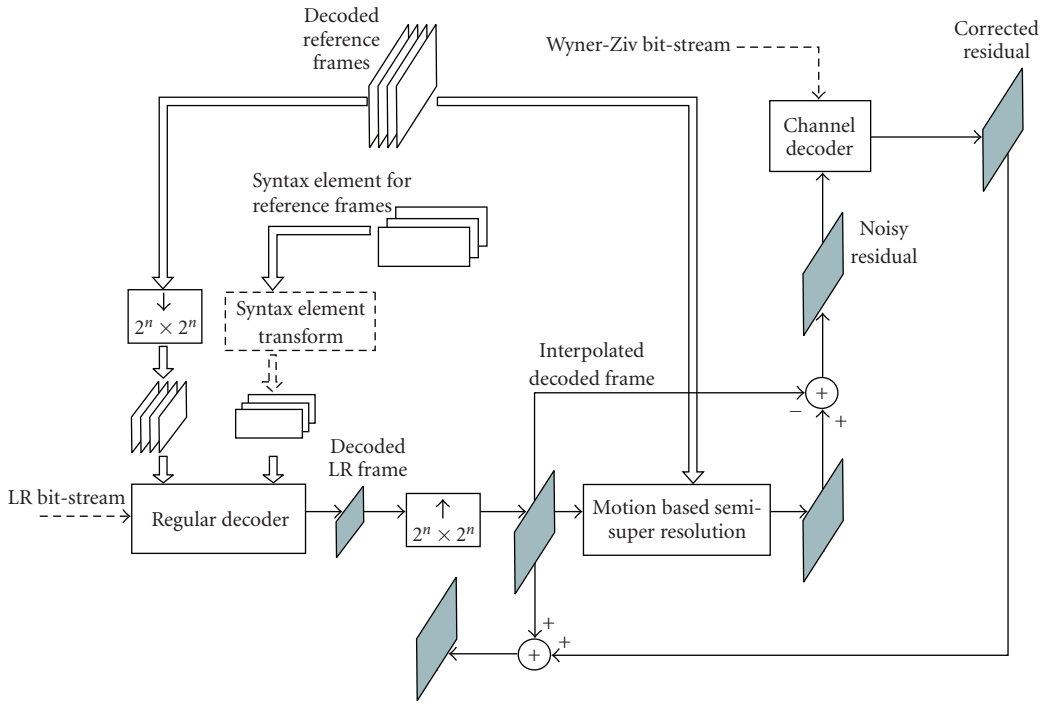


Figure 4: Decoder of the WZ-mixed resolution framework.

between F_{n-1} and F_{n+1}. For a given macroblock in F_{n+1}, it searches the reference F_{n-1} to find the best match for a 16×16 block, named, the reference block. This reference block is kept and translated by $MV_F/2$. This approach leads to two phenomena that did not happen in the SE-B method: overlapping and blank areas. There are three cases for any given pixel:

(i) it is uniquely defined by a single motion vector;

(ii) it is defined by more than one motion vector (an overlapping occurred);

(iii) it is not defined by any motion vector (it is left blank).

In order to perform motion compensation, we need to assign a motion vector or filling process for every pixel. The first case is trivial. For the second case, when more than one option for a pixel exists, a simple average might solve the

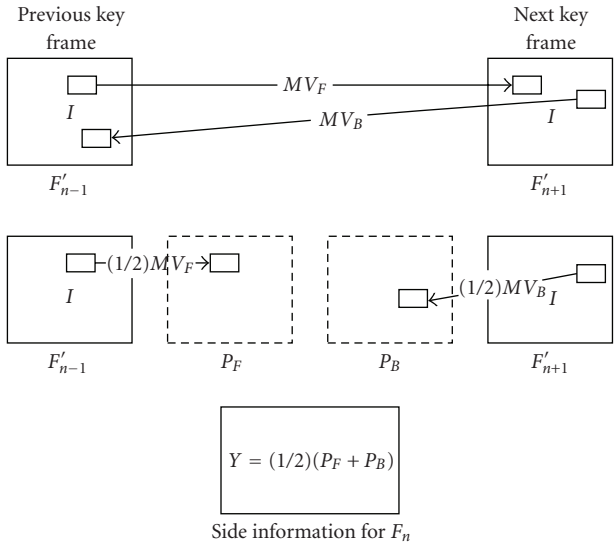
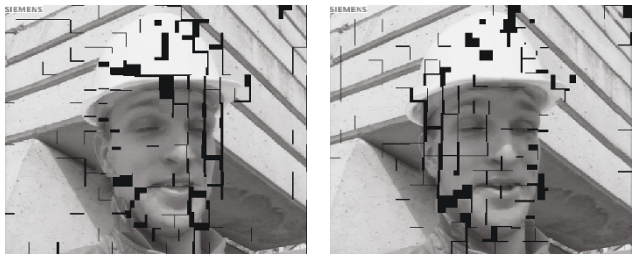


Figure 5: Illustration of SE-B.



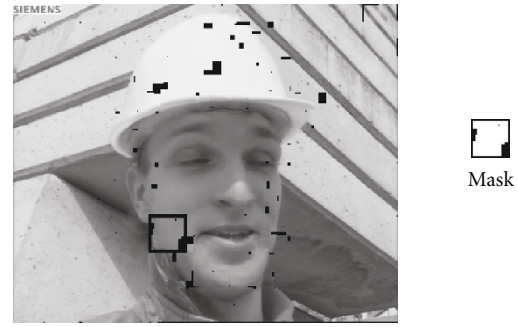
(a) SI with $MV_F/2$ (b) SI with $MV_B/2$

Figure 6: Generating the SI frame.

problem. The last case is more challenging, since no motion vector points to a pixel. One could use the colocated pixel in the previous frame. However, it may not be very efficient since it might be that the motion vector of that block is not zero.

Figure 6(a) shows the second frame of the Foreman CIF sequence using $MV_F/2$. In this case, the key frames were coded with H.264 INTRA with quantization parameter Qp = 18. The overlapping areas were averaged and, as expected, there are some blank areas. In Figure 6(b) it is shown the same frame using $MV_B/2$. There are also some blank areas, but most of them are in different places.

So, combining the frame generated by the forward estimation with the one generated by backward estimation results in a frame with less blank areas, which is depicted in Figure 7(a). After the motion estimation and compensation, and after averaging the overlapping areas, the SI frame might still contain some blank areas. At this point, there is enough information available about the current frame to perform motion estimation using the current SI frame and the previous frame F_{n-1}. The current frame is divided into blocks of 32 × 32 pixels. Then, if there is a blank area in a macroblock, motion estimation is performed for this macroblock. The blank area is not considered when



(a) (b)

Figure 7: (a) Combining the frames in Figure 6. (b) A mask use to perform motion estimation using the current SI frame.

calculating the sum of absolute difference (SAD), that is, a mask with the blank areas is used in the motion estimation process in order to compute only the nonblank areas. Once the new reference block is found, its pixels are used to fill the blank area in the current macroblock. An example of a mask is shown in Figure 7(b), used in the region marked in Figure 7(a).

In order to improve the method, bidirectional motion estimation is performed. To fill the blank areas, a reference block is searched in both the previous and next frames. The result for this single frame is shown in Figure 8.

Note that, in the proposed method, the reference block found using the motion estimation process is kept and translated to the SI frame by a motion vector that is half the original motion vector. In SE-B, the reference block is changed while the motion vector is kept. In another technique [22], the reference block is also kept, but, in order to prevent the uncovered and overlapping areas, motion vectors are changed to point to the middle of the current block in the SI frame. In the proposed method, however, both the motion vector and the reference block are kept. Also, the proposed algorithm is focused on improving the motion estimation based on the key frames. This technique can be used along with spatial motion smoothing techniques and motion vectors refinements [21, 22].

In the unlikely case of blocks wherein most or all of the pixels are blank, one can, for example, use colocated pixels for compensation. These cases are rare and can be avoided with careful choices of the sizes of the blocks and of the motion vector search window.

3.2. Super-resolution Using Key Frames. At the decoder, in the SSWZ codec, the SI is iteratively generated. However, the first iteration is different form the other ones and represents an important contribution of this paper. In the first iteration, similar to an example-based algorithm [31], we seek to restore the high-frequency information of an interpolated block through searching in previous decoded key frames for a similar block, and by adding the high-frequency of the chosen block to the interpolated one.

Note that the original sequence of frames at a high resolution has both key frames and nonkey frames (WZ frames).

Figure 8: Final SI frame of the motion-modeling SI estimator. PSNR = 33.13 dB (the key frames used to generated this SI frame had 38.09 dB and 38.16 dB).

The framework encodes the WZ frames at a lower resolution and the key frames at regular resolution. At the decoder, the video sequence is received at mixed resolution.

The decoded WZ frames have lost high-frequency content due to decimation and interpolation. Our algorithm tries to recover the lost high frequency content using temporal information from the key frames. Briefly, in the first iteration, the algorithm works as follows.

(i) First, we interpolate the WZ frames to the spatial resolution of the key frames to obtain all the decoded frames at the desired resolution.

(ii) Then, the key frames are filtered with a low-pass filter and the high frequency content is obtained as a difference between the original key frames and their filtered version.

(iii) A block matching algorithm is used, with the interpolated nonkey frame as source and the filtered key frames as reference, in order to find the best predictor for each block of the nonkey frame.

(iv) The corresponding high frequency content of the predictor block is added to the block of the WZ frame, after scaling it by a confidence factor.

The past and future reference frames in the frame store of the current WZ frame are low-pass filtered. The low-pass filter is implemented through downsampling followed by an up-sampling process (using the same decimator and interpolator applied to the WZ frames). At this point, we have both key and nonkey frames interpolated from a LR version. Next, a block-matching algorithm is applied using the interpolated decoded frame. The block-matching algorithm works as follows.

Let a frame $F = B + H$, where B is the decimated and interpolated (filtered) version of F, while H is the residue, or its high frequency. For every 8×8 block in the interpolated decoded frame, the best sub-pixel motion vectors in the past and future filtered frames are computed. If the corresponding best predictor blocks are denoted as B_p and B_f in the past and future filtered frames, respectively, several predictor candidates are calculated as

$$B = \alpha B_p + (1 - \alpha) B_f, \tag{1}$$

where α assumes values between 0 and 1. In our implementation we use $\alpha = \{0.0, 0.25, 0.5, 0.75, 1.0\}$. Then, if the SAD of the best predictor of a particular macroblock is lower than a threshold T, the corresponding high-frequency of the matched block (i.e., H_p and H_f) of the key frame is added to the block to be superresolved. In other words, we add

$$\alpha H_p + (1 - \alpha) H_f. \tag{2}$$

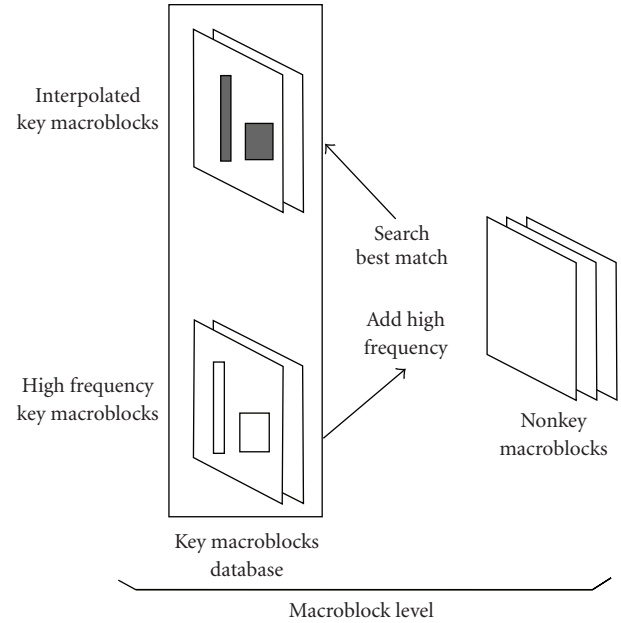


Figure 9: After searching for a best match in the database, we add the corresponding high-frequency to the block to be superresolved.

Figure 9 illustrates the process.

Differently from previous works [16–18], we are adding high frequency content. We want to avoid adding noise in cases where a match is not very close. Hence, we use a confidence factor to scale the high-frequency contents before being added to the LR block.

We assume that the better the match, the better the confidence we have and the more high frequency we add. For example, the confidence factor can be calculated based on the minimum SAD obtained from the block matching algorithm and the rate (R_n) spent by the coder in order to encode the current block. If the minimum SAD calculated during the block matching algorithm has a high value; it is unlikely that the high frequency of the key frame block would exactly match the lost high frequency of the nonkey frame block. Then, it is intuitive to think that a lower minimum SAD gives us more confidence in our match. Besides, if at the encoder side, a large bit-rate is spent to code a particular block, it is likely to be because no good match in the reference frames was found. Thus, the higher the bit-rate, the lower

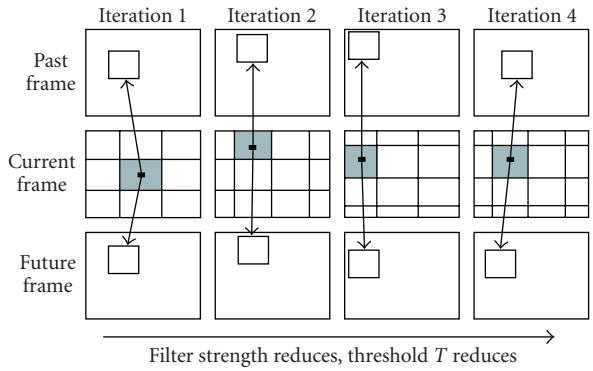


Figure 10: SI generation for nonreference WZ frames. Threshold reduces, and the grid is shifted from iteration to iteration.

the confidence. The confidence is reflected as a scaling factor that multiplies each pixel of the high frequency block, before adding it to the block to be superresolved. For example, one scaling metric can be

$$c = 1 - \left(\frac{\min\left(\mathrm{SAD} + \lambda R_n, T\right)}{T}\right), \tag{3}$$

where λ is a Lagrange multiplier. Note, that if SAD = R_n = 0 then $c = 1$. This means that all the high-frequency content will be added. On the other hand, if $(\mathrm{SAD} + \lambda R_n) = T$ then $c = 0$, so no high frequency is added. The values of T and λ can be empirically found using different test sequences. In our implementation $T_i = \{500, 80, 60, 20, 5\}$, where i indicates the number of the iteration as we will describe next. And the factor λ depends on the QP used. For example, we used $\lambda = k(1.2)^{(QP-12)/3}$ in our H.264/AVC implementation.

We can iteratively super-resolve the frames as in previous works [16–18, 28] by replacing the operation just described. However, after the first iteration, parameters may change. From iteration to iteration the strength of the low-pass filter should be reduced (in our implementation the low-pass filter is eliminated after one iteration). The grid for block matching is offset from iteration to iteration to smooth out the blockiness and to add spatial coherence. For example, the shifts used in four passes can be (0, 0), (4, 0), (0, 4) and (4, 4) (see Figure 10). It is important to note that after the first iteration we already have a frame with high frequency content. Hence, after the first iteration the SI generation is similar to the work presented at [18], where the entire block is replaced by the unfiltered matched block on the key frames, instead of just adding high-frequency. In other words, after the first iteration we replace $B + H$ rather than adding H. Then, after the first iteration the threshold T is drastically reduced, and continues to be gradually reduced so that fewer blocks are changed at later iterations.

4. Results and Simulations

All the SI generation methods were implemented on the KTA software implementation of H.264/AVC [32]. In our entire tests, we use fast motion estimation, the CAVLC entropy coder, no rate-distortion optimization, 16 × 16-pixel search range and spatial direct mode type for B-frames.

For the TDWZ codec, we set the coder to work in two different modes: *IZIZI* and *IZPZP*. That is, in the first mode, all the key frames are set to be coded as conventional I-frames. In the second mode, the key frames are set to be P-frames, with the exception of the first frame. In both cases, Z refers to the WZ frame. Since the goal is SI comparison, the WZ layer for the TDWZ is not really generated. For the WZ frames, DCT transform, quantization and bit plane creation are computed only to be included as overhead coding time. The SSWZ codec was set to work in *IbIbI, IbPbP* and *IpPpP* modes, where b represents the nonreference B-frames coded at quarter resolution and p is a disposable nonreference P frame [14] also encoded at quarter resolution

In Table 1, we present the average results for encoding 299 frames of each of seven CIF sequences: Mobile, Silent, Foreman, Coastguard, Mother and Daughter, Soccer and Hall Monitor. The average total encoding time for different *QPs* of all the key frames, and overhead for the WZ frames, is presented in Table 1. In there, ME means the coding time spent during motion estimation. For the TDWZ codec the overhead for coding the Z frames is included except for channel coding. Note that the *IZPZP* mode is about 7 to 8 times more complex than *IZIZI* mode, because of motion estimation on the key frames. However, a better RD performance is expected for the *IZPZP* mode. For the SSWZ codec the results for the encoding time include the overhead for creating the WZ layer using memoryless cosets as explained in [16, 17, 27]. For the case of *IbIbI* mode, the encoder is about 3 times slower than the temporal scalable codec working in *IZIZ* mode. For the other tests, we note that the spatially scalable coder complexity, working on *IbPbP* or *IpPpP* mode, is comparable to the temporally scalable coder working in *IZPZP* mode. The latter encodes about 20% faster than the SSWZ encoder at *IbPbP* mode. All the coding tests were made on an Intel Pentium D 915 Dual Core, with 2.80 GHz and 1 GB DDR2 of RAM, Windows OS.

Table 1 also shows results for the conventional H.264/AVC codec working in *IBPBP* and IP_dPP_dP modes without rate-distortion optimization. The B-frames are nonreference frames and P_d indicates a disposable nonreference P-frame. It can be seen that all WZ frameworks spend less encoding time than conventional coding. As expected the TDWZ codec with the key frames encoded as I-frames yields the faster encoding.

Even though the focus of a DVC codec is the reduction in encoding complexity, an evaluation of the decoding time is important to understand the complexity of the entire system. In Table 2 we present the average SI generation time for a single frame of the tested sequences. Note that our implementations are not optimized. Time should be considered only for decoding complexity comparison between the different SI techniques. An optimized implementation should be able to generate SI faster, in all cases. The SE-B [19] and the motion-modeling method used 16 × 16 blocks and search area of 16 pixels. Note that the proposed method did not add to much decoding complexity in comparison with the simple SE-B algorithm. For the spatial scalable coder, the

Table 1: Average encoding time for the temporally scalable WZ codec (TDWZ), spatial scalable codec (SSWZ) and conventional H.264/AVC. TOTAL = total coding time in seconds, ME = motion estimation coding time in seconds.

		IZIZ	*IZPZP*
TDWZ codec		19.28 (ME: 0)	142.08 (ME: 109.32)
	IbIbI	*IbPbP*	*IpPpP*
SSWZ codec	64.33 (ME: 33.75)	178.23 (ME: 136. 81)	164.18 (ME: 126.83)
		IBPBP	IP_dPP_dP
Conventional AVC codec		307.6 (ME: 236.81)	258.21 (ME: 199.44)

Table 2: Average SI generation time in frame per second.

	TDWZ codec	SSWZ codec
SE-B	1.02	—
Motion-Modeling	1.42	—
Semi-super resolution	—	6.03

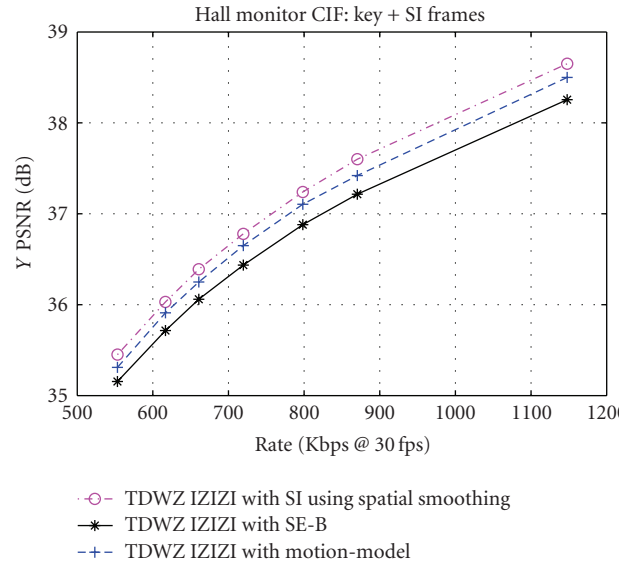


Figure 11: Results for SI generation for the luminance component of Hall Monitor CIF sequence.

time required to create one SI frame using the semi-super resolution process for the same block size was around 1.2 seconds. However, as described above, for the semi-super resolution method is better to use an 8×8 block size for block matching. The search area was set to 24 pixels. With these conditions, the required time to create an SI frame was approximately 6 seconds.

Even though an important issue in WZ coding is reduction in encoding complexity, it should not be achieved at a substantial cost in bandwidth. In other words, a WZ coder should not yield too much loss in RD performance in comparison with conventional encoding. As previously mentioned, the SI generation plays an important part in determining the overall performance of any WZ codec. In Figure 11, we compare the RD performance, for CIF resolution sequence, of: (i) our implementation of the SE-B algorithm [19, 20], (ii) SI generation with spatial smoothing [21] and (iii) the proposed motion-modeling method. The PSNR curves correspond to 299 frames (key frames and SI frames, no parity bits are transmitted).

The real performance of the WZ codecs depends on the enhancement WZ layer. However, it is assumed that a better SI can potentially improve the performance of a WZ codec. Figure 11 compares key plus SI frames for the TDWZ codec in *IZIZI* mode for a low-motion sequence. Note that, in Figure 12, both PSNR and rate are given for the luminance component only. It can be seen that the motion-modeling algorithm outperforms the SE-B algorithm, without significantly increasing the SI generation time (see Table 2). However, it underperforms the one with frame interpolation and spatial smoothing. The performance differences are in line with the respective increase in complexity. The spatial smoothing could also be incorporated into the other two components to increase both the performance and the decoding complexity. Note that for low motion sequence, the SI generation methods that use temporal frame interpolation have good performance; since it is possible to generate an accurate prediction of the motion among the key frames and the frame being interpolated.

In Figure 12 a similar comparison is done, for the superresolution process, also using intra key frames (*IbIbI* mode). In this case, the semi-super resolution process outperforms previous techniques at a cost of higher encoding complexity. Note that the Soccer sequence presents high motion; therefore it is harder to make an accurate temporal interpolation of the frame. In these cases, the Si generated by the superresolution process should potentially achieve better results.

In Figure 13, we compare the performance for the different SI methods in different coding modes. It compares key and SI frames for the TDWZ codec in *IZIZI* and *IZPZP* modes, using the two implemented SI generators: the SE-B estimator and the motion-modeling estimator. It also shows results for the SSWZ codec in *IbIbI* and *IbPbP* modes. PSNR results are computed for the luminance component only, but the rate includes luminance and chrominance components. It can be seen that, for the TDWZ coder, the motion-modeling method consistently outperforms the SE-B method. Also, as expected, the SSWZ codec has the better overall RD performance, at a cost of a higher coding time. In this figure, a better RD performance will simply indicate a better SI, since no parity bits were transmitted.

It is known that the TDWZ codec normally outperforms intracoding, but it is worse than coding with zero motion vectors [29]. Since the SSWZ is less common in the literature, in Figures 14 through 16 we show results for the SSWZ codec including the enhancement layer formed by memoryless cosets with the coding parameters mechanism and correlated statistics estimation described in

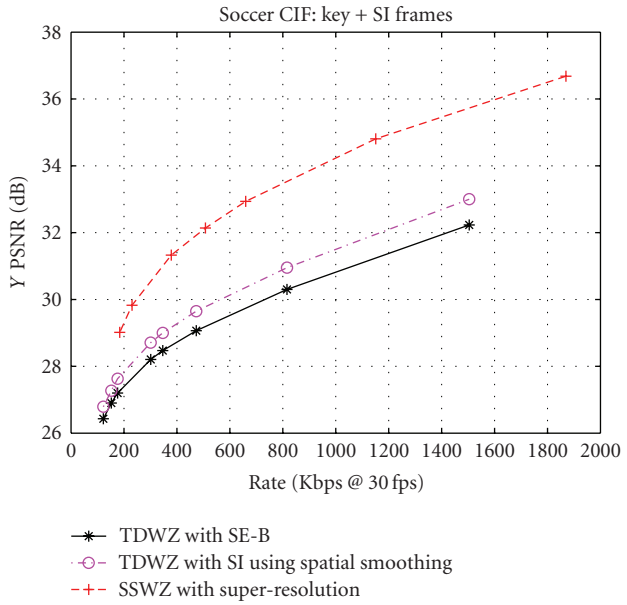


Figure 12: Results for SI generation for the luminance component of Hall Monitor CIF sequence.

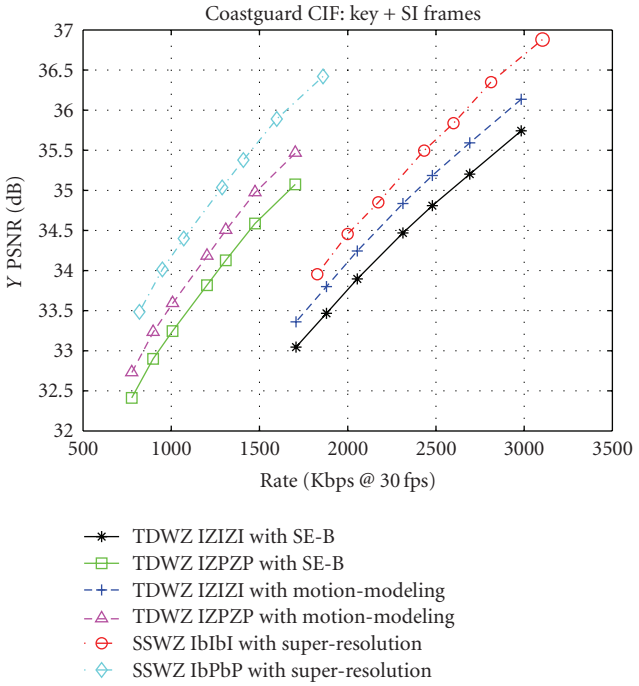


Figure 13: Results for SI generation for Foreman CIF sequence.

[17, 18, 27, 28]. We compare (i) conventional H.264/AVC codec working in *IBPBP* or IP_dPP_dP mode, with 2 reference frames, search range of 16 pixels and CAVLC entropy encoder, (ii) the SSWZ codec after three iterations (in *IbPbP* or *IpPpP* modes) with similar coding settings, and (iii) conventional coding in *IBPBP* or IP_dPP_dP modes but with a search range of zero (i.e., zero motion vector coding).

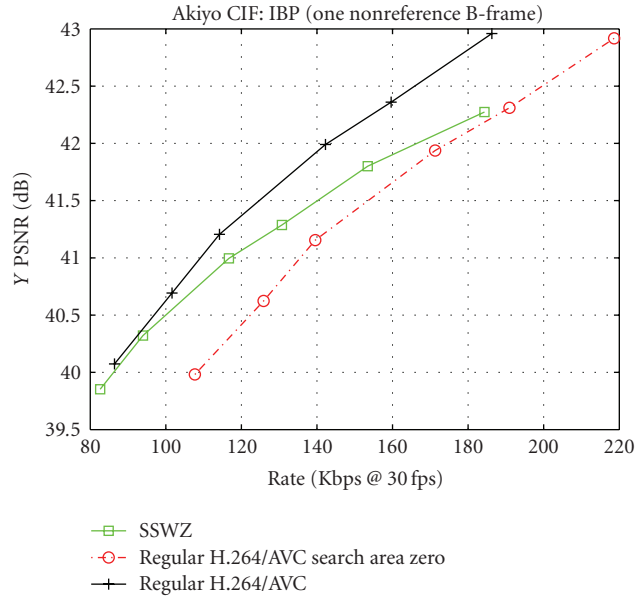


Figure 14: Results of SSWZ codec for Akiyo CIF sequence.

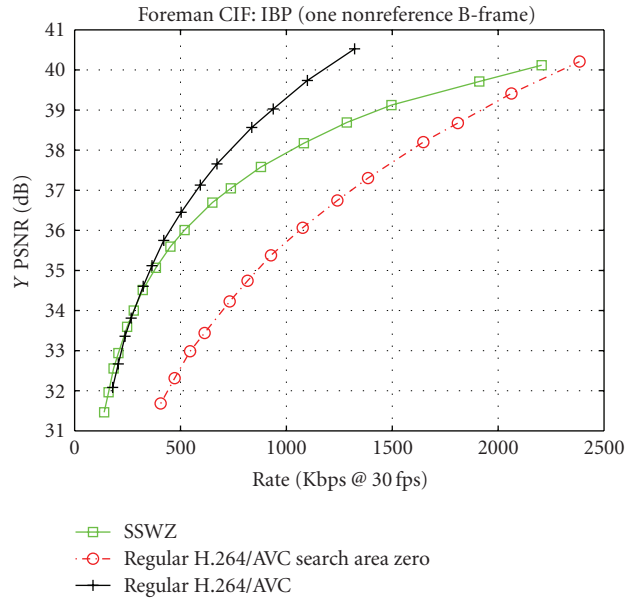


Figure 15: Results of SSWZ codec for Foreman CIF sequence.

It can be seen that the WZ coding mode is competitive. The SSWZ codec outperforms conventional coding with zero motion vectors at most rates. The gap between conventional coding and WZ coding, with similar encoding settings, is larger at high rates. However, as can be seen in the Mother and Daughter CIF sequence, the WZ mode may outperform the conventional H.264 at low rates. In fact, the SSWZ can potentially yield better results for low rates in low motion sequences, than conventional coding. This can be explained because the SSWZ uses multi resolution encoding that can be seen as an interpolative coding scheme which is known for their good performance of low bit-rates. Other interpolative

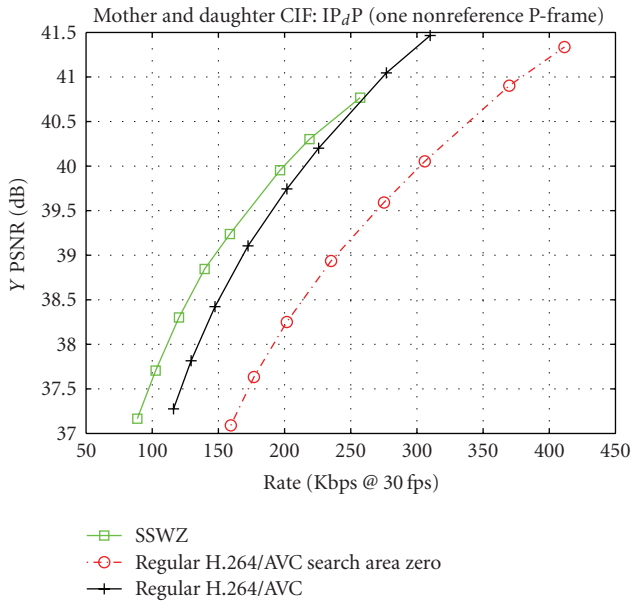


Figure 16: Results of SSWZ codec for Mother and Daughter CIF sequence.

coding schemes have been used in image compression with better performance than conventional compression for low rates [33]. Therefore, it is possible to have a WZ codec operating with a 40%–50% reduction in encoding complexity (see encoding time for conventional *IBPBP* coding mode and SSWZ *IbPbP* coding mode in Table 1), and still produce better results than conventional coding for certain rates. Also, the SSWZ is not using a feedback channel, the correlation statistics are estimated [28]. Thus, a more robust estimation may significantly improve the performance. A specially designed entropy codec can encode the cosets more efficiently.

5. Conclusions

In this work, we have introduced two new SI generation methods, one for a temporally scalable Wyner-Ziv coding mode and another one for a spatially scalable Wyner-Ziv coding mode. The first SI generation method, proposed for the temporally scalable codec, models the motion between two key frames as linear. Thus, the motion between one key frame and the current WZ frame, with a GOP size of 2, will be half of the motion between the key frames. An algorithm for solving the problem of overlapping and blanks was proposed. The results show that this SI method has a better performance than the SE-B estimator [19], while being significantly simpler than frame interpolation with spatial motion smoothing and motion vector refinement [22]. However, the later outperforms the proposed technique. Nevertheless, spatial motion smoothing and motion vectors refinement tools can also be incorporated in the present framework potentially increasing its performance. The SI generation for the spatial scalable codec uses a confidence value to scale the amount of high-frequency content that is added to the block to be superresolved. It works better than the previous techniques [16–18]. This SI method helps a spatial scalable Wyner-Ziv to achieve competitive results.

Also, a complexity comparison using coding time as benchmark was presented. The temporal scalable codec with key frames coded as "intra" frames is considerably less complex than any other WZ codec. However, it has the worst RD performance (considering key frames and SI). The WZ coding mode with spatially scalability is about 20% more complex than the temporal scalable codec using *P*-frames as key frames in both cases. In the other hand, the spatial scalable coder is more competitive and may outperform a conventional codec for low-motion sequences at low rates. Thus, in certain conditions, the spatial scalable framework allows reversed complexity coding without a significant cost in bandwidth.

We can conclude that a spatial scalable WZ codec produces RD results closer to conventional coding than the temporal scalable WZ codec. However, a complete WZ codec may be able to have both coding modes, since the temporal scalable mode can achieve lower complexity.

Acknowledgment

This work was supported by Hewlett-Packard Brasil.

References

[1] J. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions 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 Transactions on Information Theory*, vol. 2, no. 1, pp. 1–10, 1976.

[3] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," in *Proceedings of the Data Compression Conference (DCC '99)*, pp. 158–167, Snowbird, Utah, USA, March 1999.

[4] A. Aaron, S. D. Rane, E. Setton, and B. Girod, "Transform-domain Wyner-Ziv codec for video," in *Visual Communications and Image Processing 2004*, vol. 5308 of *Proceedings of SPIE*, pp. 520–528, San Jose, Calif, USA, January 2004.

[5] R. Puri and K. Ramchandram, "Prism: a new robust video coding architecture based on distributed compression principles," in *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing*, pp. 1–10, Allerton, Ill, USA, October 2002.

[6] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," in *Visual Communications and Image Processing*, vol. 5308 of *Proceedings of SPIE*, pp. 83–91, San Jose, Calif, USA, January 2004.

[7] Q. Xu and Z. Xiong, "Layered Wyner-Ziv video coding," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3791–3803, 2006.

[8] H. Wang, N.-M. Cheung, and A. Ortega, "A framework for adaptive scalable video coding using Wyner-Ziv techniques," *EURASIP Journal on Applied Signal Processing*, vol. 2006, Article ID 60971, 18 pages, 2006.

[9] M. Tagliasacchi, A. Majumdar, and K. Ramchandran, "A distributed-source-coding based robust spatio-temporal scalable video codec," in *Proceedings of the 24th Picture Coding*

Symposium (PCS '04), pp. 435–440, San Francisco, Calif, USA, December 2004.

[10] X. Wang and M. T. Orchard, "Desing of trellis codes for source coding with side information at the decoder," in *Proceedings of Data Compression Conference (DCC '01)*, pp. 361–370, Snowbird, Utah, USA, March 2001.

[11] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proceedings of the Data Compression Conference (DCC '02)*, pp. 252–261, Snowbird, Utah, USA, April 2002.

[12] M. Ouaret, F. Dufaux, and T. Ebrahimi, "Codec-independent scalable distributed video coding," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '07)*, vol. 3, pp. 9–12, San Antonio, Tex, USA, September 2007.

[13] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[14] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[15] A. M. Aaron, S. Rane, R. Zhang, and B. Girod, "Wyner-Ziv coding for video: applications to compression and error resilience," in *Proceedings of the Data Compression Conference (DCC '03)*, pp. 93–102, Snowbird, Utah, USA, March 2003.

[16] D. Mukherjee, "A robust reversed complexity Wyner-Ziv video codec introducing sign-modulated codes," Tech. Rep. HPL-2006-80, HP Labs, Palo Alto, Calif, USA, May 2006.

[17] D. Mukherjee, B. Macchiavello, and R. L. de Queiroz, "A simple reversed-complexity Wyner-Ziv video coding mode based on a spatial reduction framework," in *Visual Communications and Image Processing 2007*, vol. 6508 of *Proceedings of SPIE*, pp. 1–12, San Jose, Calif, USA, January 2007.

[18] B. Macchiavello, R. L. de Queiroz, and D. Mukherjee, "Motion-based side-information generation for a scalable Wyner-Ziv video coder," in *Proceedings of IEEE International Conference on Image Processing (ICIP '07)*, vol. 6, pp. 413–416, San Antonio, Tex, USA, September 2007.

[19] Z. Li and E. J. Delp, "Wyner-Ziv video side estimator: conventional motion search methods revisited," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 1, pp. 825–828, Genova, Italy, September 2005.

[20] Z. Li, L. Liu, and E. J. Delp, "Rate distortion analysis of motion side estimation in Wyner-Ziv video coding," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 98–113, 2007.

[21] C. Brites, J. Ascenso, J. Q. Pedro, and F. Pereira, "Evaluating a feedback channel based transform domain Wyner-Ziv video codec," *Signal Processing: Image Communication*, vol. 23, no. 4, pp. 269–297, 2008.

[22] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *Proceedings of the 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, pp. 1–6, Smolenice, Slovakia, June-July 2005.

[23] F. Brandi, R. L. de Queiroz, and D. Mukherjee, "Super resolution of video using key frames and motion estimation," in *Proceedings of the IEEE International Conference on Image Processing (ICIP '08)*, pp. 321–324, San Diego, Calif, USA, October 2008.

[24] X. Artigas and L. Torres, "Iterative generation of motion-compensated side information for distributed video coding," in *Proceedings of IEEE International Conference on Image Processing (ICIP '05)*, vol. 1, pp. 833–836, Genova, Italy, September 2005.

[25] J. Ascenso, C. Brites, and F. Pereira, "Motion compensated refinement for low complexity pixel based distributed video coding," in *Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS '05)*, pp. 593–598, Como, Italy, September 2005.

[26] W. A. R. J. Weerakkody, W. A. C. Fernando, J. L. Martínez, P. Cuenca, and F. Quiles, "An iterative refinement technique for side information generation in DVC," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '07)*, pp. 164–167, Beijing, China, July 2007.

[27] D. Mukherjee, "Optimal parameter choice for Wyner-Ziv coding of laplacian sources with decoder side-information," Tech. Rep. HPL-2007-34, HP Labs, Palo Alto, Calif, USA, 2007.

[28] B. Macchiavello, D. Mukherjee, and R. L. de Queiroz, "A statistical model for a mixed resolution Wyner-Ziv framework," in *Proceedings of the 26th Picture Coding Symposium (PCS '07)*, Lisbon, Portugal, November 2007.

[29] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The discover codec: architecture, techniques and evaluation," in *Proceedings of the 26th Picture Coding Symposium (PCS '07)*, pp. 1–4, Lisbon, Portugal, November 2007.

[30] L. Natário, C. Brites, J. Ascenso, and F. Pereira, "Extrapolating side information for low-delay pixel-domain distributed video coding," in *Proceedings of the 9th International Workshop on Visual Content Processing and Representation (VLBV '05)*, pp. 16–21, Sardinia, Italy, September 2005.

[31] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[32] J. Jung and T. K. Tan, "KTA 1.2 software manual," VCEG-AE08, January 2007.

[33] B. Zeng and A. N. Venetsanopoulos, "A JPEG-based interpolative image coding scheme," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '93)*, vol. 5, pp. 393–396, Minneapolis, Minn, USA, April 1993.