Progressively Refined Wyner-Ziv Video Coding for Visual Sensors

NIKOS DELIGIANNIS and FREDERIK VERBIST, Vrije Universiteit Brussel
JÜRGEN SLOWACK and RIK VAN DE WALLE, Ghent University
PETER SCHELKENS and ADRIAN MUNTEANU, Vrije Universiteit Brussel

Wyner-Ziv video coding constitutes an alluring paradigm for visual sensor networks, offering efficient video compression with low complexity encoding characteristics. This work presents a novel hash-driven Wyner-Ziv video coding architecture for visual sensors, implementing the principles of successively refined Wyner-Ziv coding. To this end, so-called side-information refinement levels are constructed for a number of grouped frequency bands of the discrete cosine transform. The proposed codec creates side-information by means of an original overlapped block motion estimation and pixel-based multihypothesis prediction technique, specifically built around the pursued refinement strategy. The quality of the side-information generated at every refinement level is successively improved, leading to gradually enhanced Wyner-Ziv coding performance. Additionally, this work explores several temporal prediction structures, including a new hierarchical unidirectional prediction structure, providing both temporal scalability and low delay coding. Experimental results include a thorough evaluation of our novel Wyner-Ziv codec, assessing the impact of the proposed successive refinement scheme and the supported temporal prediction structures for a wide range of hash configurations and group of pictures sizes. The results report significant compression gains with respect to benchmark systems in Wyner-Ziv video coding (e.g., up to 42.03% over DISCOVER) as well as versus alternative state-of-the-art schemes refining the side-information.

Categories and Subject Descriptors: I.4.2 [Image Processing and Computer Vision]: Compression (Coding); E.4 [Coding and Information Theory]: Data Compaction and Compression, Error Control Codes
General Terms: Design, Algorithms, Performance
Additional Key Words and Phrases: Wyner-Ziv coding, hash-driven distributed video coding, low-cost encoding, visual sensors, successive refinement

ACM Reference Format:
DOI: http://dx.doi.org/10.1145/2530279

A preliminary version of this article was presented at the 2012 European Signal Processing Conference (EUSIPCO) and appears in the conference proceedings [Deligiannis et al. 2012c]. This work is supported by the Fund for Scientific Research (FWO) Flanders, by the projects G.0391.07 and G.0047.12, and by the Flemish Institute for the Promotion of Innovation by Science and Technology (IWT).
Authors’ addresses: N. Deligiannis, F. Verbist, P. Schelkens, and A. Munteanu, Department of Electronics and Informatics, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium, and iMinds, Gaston Crommenlaan 8 (b102), 9050 Ghent, Belgium; email: {ndeligia, fverbist, peschelke, acmuntea}@etro.vub.ac.be; J. Slowack and R. Van de Walle, ELIS Department, Multimedia Lab, Ghent University, Gaston Crommenlaan 8, 9050 Ghent, Belgium, and iMinds, Gaston Crommenlaan 8 (b102), 9050 Ghent, Belgium; email: j.slowack, Rik.Vandewalle}@ugent.be.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2014 ACM 1550-4859/2014/01-ART21 $15.00
DOI: http://dx.doi.org/10.1145/2530279

ACM Transactions on Sensor Networks, Vol. 10, No. 2, Article 21, Publication date: January 2014.
1. INTRODUCTION

Wireless sensors are employed to collect information from their surroundings, carry out menial processing and transmit the gathered data to a central base station. Wireless multimedia sensor networks (WMSNs) [Akyildiz et al. 2007] consist of sensors with video capturing functionality that can complement the human ability to safeguard areas, detect events or explore hard-to-reach (or even hostile) environments. For example, the integrated mobile surveillance system from Tseng et al. [2007] comprises multiple static wireless sensors along with a number of mobile sensors. Appropriate algorithms enable the system of Tseng et al. [2007] to detect particular intrusion events and identify sources of invasion. In contrast to conventional surveillance solutions that usually gather a large amount of data from wired cameras, such a WMSN-based system notably reduces the computational demands and manpower resources. In an alternative scenario, WMSNs can support health or elderly care by, for instance, spotting unattended wandering or actions of patients, or by providing fall detection, while at the same time preserving the privacy of the patients. In this context, energy-efficient and inexpensive low-resolution recording sensors, like the ones from Camilli and Kleihorst [2011], could severely reduce the total implementation cost, thereby lowering the strain on state and health care finances. Along a different trend, highly specialized visual sensor applications centered on the human body and intended for clinical diagnoses or observation receive growing attention. In particular, wireless capsule endoscopy [McCaffrey et al. 2008], where videos of the digestive track are acquired using a camera-pill, shows an important positive diagnosis rate compared to other diagnostic methods. Maximizing battery life, while providing high-resolution and high-quality video, poses a significant challenge.

From a coding perspective, the restricted size and energy supply of wireless visual sensors confine the hardware and processing power demands of the employed video codec. Moreover, to minimize the volume of information that a sensor node transmits, high compression performance is required. Traditional video codecs, for instance, the H.26x standards [Wiegand et al. 2003], efficiently exploit the available correlation through complex spatial or temporal prediction techniques at the encoder. Although such an approach vastly increases compression efficiency, it leads to complex video encoders, thereby rendering such codecs unsuitable for visual sensors. Based on the theory by Slepian and Wolf [1973] and Wyner and Ziv [1976], distributed source coding (DSC) enables efficient compression by exploiting the source statistics only at the decoder. Applying DSC principles [Ishwar et al. 2003], distributed video coding (DVC) systems [Puri and Ramchandran 2002; Girod et al. 2005; Artigas et al. 2007], possess simple encoders but sophisticated decoders. Moreover, DVC systems can elegantly integrate error resilience [Xu et al. 2007]. Therefore, DVC systems have been recognized as viable solutions for WMSNs and are regarded to play a key role in future WMSN protocols [Xiong et al. 2004], [Akyildiz et al. 2007]. Furthermore, in the context of medical imaging, our prior work in [Deligiannis et al. 2012b] has demonstrated the high potential of DVC in wireless capsule endoscopy.

To apply DSC principles in an effective mono-view video coding paradigm, the video sequence is split in two correlated sources of information. One source is coded using conventional intra-frame video coding techniques, and then used at the decoder to generate a side-information signal. This side-information signal is subsequently employed to Wyner-Ziv decode the other source of information. Despite the research invested in DVC during the last decade, current systems still fail to match the coding performance of their traditional video coding counterparts. One of the main reasons for this shortcoming is the deficiency of Wyner-Ziv video decoders to generate efficient motion-compensated predictions [Li et al. 2007] to serve as side-information.
Common DVC schemes [Girod et al. 2005; Artigas et al. 2007] employ motion-compensated interpolation (MCI) to generate a temporal prediction at the decoder. However, MCI performs motion estimation without access to the original frame by assuming a linear motion model, which falls short in case of irregular motion [Li et al. 2007]. In DVC, coding large GOPs comes with the profit of reduced encoding complexity by coding more frames under the Wyner-Ziv paradigm. Alas, the compression performance of state-of-the-art MCI-based DVC architectures [Artigas et al. 2007] deteriorates when the GOP size increases. To increase the quality of the created side-information, alternative DVC schemes employ hash-based motion estimation [Aaron et al. 2004; Ascenso et al. 2010; Martinian et al. 2006; Deligiannis et al. 2009b; Deligiannis et al. 2012b, 2012a; Verbist et al. 2011], where appropriate hash information is sent to the decoder to support motion estimation.

Although side-information creation techniques based on bidirectional prediction, as in Artigas et al. [2007], Verbist et al. [2011], Deligiannis et al. [2012b], Ye et al. [2009], and Martins et al. [2009], provide the finest DVC performance, they require a reference frame from a subsequent temporal instance to be available at the decoder. As a consequence, bidirectional prediction introduces structural delay, which is a potential bottleneck for low-delay sensor applications. Moreover, since the frame acquisition order is different from the encoding order, the DVC encoder has to buffer the intermediate frames in every GOP. This increases the memory footprint of the encoder and opposes the low-cost requirements of sensor nodes. Current DVC approaches targeting low-delay, are based on motion-compensated extrapolation (MCE) [Girod et al. 2005; Li et al. 2007]. Unfortunately, MCE-based systems notably underperform compared to their bidirectional counterparts [Girod et al. 2005; Li et al. 2007].

Driven by the principles of successively refined WZ coding [Steinberg and Merhav 2004], this work presents a novel successively refined Wyner-Ziv video coding architecture for WMSNs, which provides high rate-distortion (RD) performance and low encoding complexity. Based on our prior work [Deligiannis et al. 2012a, 2012b], initial side-information is constructed using hash-based motion estimation, thereby avoiding blind motion estimation at the decoder, as in MCI-based DVC schemes, for instance, Girod et al. [2005] and Artigas et al. [2007]. However, advancing over our prior art [Deligiannis et al. 2009b, 2012a, 2012b], the proposed scheme follows a novel successive refinement strategy, which improves the side-information quality over several distinct levels built for a number of grouped DCT coefficient bands. The proposed progressive refinement method introduces several novel tools compared to contemporary side-information refinement techniques, for instance, Adikari et al. [2006], Varodayan et al. [2008], Ye et al. [2009], and Martins et al. [2009]. The major component of the presented codec is a new successively refined overlapped block motion estimation (OBME) technique at the decoder, which performs pixel-based multihypothesis prediction at each refinement level. Our novel successively refined OBME provides high RD performance with a limited number of refinement levels, hence constraining the decoding complexity and structural latency induced by repeated motion estimation. Furthermore, the proposed codec supports multiple temporal prediction structures including a low-delay unidirectional and a new hierarchical unidirectional prediction structure. The latter preserves low coding delay but still supports temporal scalability. Finally, this work provides an analysis of the complexity and encoder power consumption of the proposed system.

Experimentation shows that the proposed successive refinement scheme significantly improves the compression performance compared to the non-successively refined version of our system. The proposed codec also achieves remarkable compression gains compared to non-refined DVC systems, namely, DISCOVER and the hash-based system of Ascenso et al. [2010], of up to 42.03% and 27.92% rate reduction in Bjøntegaard.
measure [Bjøntegaard 2001], respectively. Moreover, compared to the recent successively refined transform-domain Wyner-Ziv (TDWZ) systems of Ye et al. [2009] and Martins et al. [2009], the proposed codec brings notable RD gains of up to 30.66% and 25.21% in Bjøntegaard rate reduction, respectively. Additionally, the proposed codec advances over our previous hash-based Wyner-Ziv video coding schemes of Deligiannis et al. [2009b, 2012a], with respective Bjøntegaard rate savings [Bjøntegaard 2001] of up to 51.55% and 11.41%.

The rest of this article is structured as follows. Section 2 reports the relevant literature and underlines the original contributions in this work. The architecture of the proposed codec is discussed in Section 3. A detailed explanation of the hash formation and compression method is presented in Section 4, whereas Section 5 expands on the different temporal prediction structures supported by the proposed system. The architectural details of the decoder related to generating side-information in a successively refined manner are clarified in Section 6. Section 7 reports a thorough experimental evaluation of the system, while Section 8 draws the conclusions of this work.

2. RELATED WORK AND CONTRIBUTIONS

Successively refined Wyner–Ziv coding encodes a source in multiple refinement stages, with different side-information and distortion conditions at each stage. Let a source $X$ be successively coded by a coarse and a refinement stage and let $R_1$, $\Delta R$, $D_1$, $D_2$, and $Y_1, Y_2$ be the rates, the distortion levels, and the side-information of the coarse and the refinement stage, respectively. If the side-information at the different stages forms a Markov chain, that is, $X \leftrightarrow Y_2 \leftrightarrow Y_1$, Steinberg and Merhav [2004] characterized the set of all achievable quadruples $(R_1, \Delta R, D_1, D_2)$. When $D_1 \geq D_2$, the successively refined Wyner-Ziv RD function $R_{X|Y_1, Y_2}^{SWRZ}(D_1, D_2)$ is bounded by

$$R_{X|Y_2}^{WZ}(D_2) \leq R_{X|Y_1, Y_2}^{SWRZ}(D_1, D_2) \leq R_{X|Y_1}^{WZ}(D_2),$$

where $R_{X|Y_1}^{WZ}(D_2), R_{X|Y_2}^{WZ}(D_2)$ are the RD functions of individual Wyner–Ziv coding with side-information $Y_1$ and $Y_2$, respectively.

When the side-information is identical for all the refinement stages, Steinberg and Merhav [2004] proved that Wyner-Ziv scenarios involving doubly symmetric binary or quadratic Gaussian sources are successively refinable, that is, $R_{X|Y_2}^{WZ}(D_2) = R_{X|Y_1}^{SWRZ}(D_1, D_2)$. Moreover, Cheng and Xiong [2005] broadened the Wyner-Ziv successive refinability property to sources defined by the sum of independent Gaussian noise and arbitrarily distributed side-information.

Initial works proposed successive side-information refinement in pixel-domain DVC architectures [Artigas and Torres 2005; Ascenso et al. 2005; Adikari et al. 2006; Weerakkody et al. 2007]. More intricate, Fan et al. [2010] proposed a multistage side-information refinement technique by performing Wyner-Ziv coding of a frame in several spatial resolution layers. However, since the compression performance of pixel-domain Wyner-Ziv video coding is rather low, state-of-the-art DVC solutions operate in the transform-domain. Successive refinement of the side-information through unsupervised learning of the motion vectors combined with LDPC decoding was proposed in Varodayan et al. [2008]. A motion estimation refinement stage based on the decoded DC components of the WZ frames was presented in Badem et al. [2008]. Alternatively, Ye et al. [2009] decoded and reconstructed all DCT bands of the WZ frame but added a post-processing procedure to update the side-information, prior to the final coefficient reconstruction. Repeated side-information generation after the reconstruction of every DCT coefficient band was put forward in Martins et al. [2009]. Side-information refinement after decoding of the DC band only, which did not require the use of an overcomplete DCT, was proposed in Deligiannis et al. [2011]. Joint successive
refinement of the side-information and the correlation channel estimation was proposed by [Deligiannis et al. 2012c].

The major contribution of this work is the introduction of a novel successively refined Wyner-Ziv video coding architecture that delivers state-of-the-art RD performance in DVC. The proposed system features several key novelties.

First, the proposed system generates initial side-information using hash-based OBME and subsequent pixel-based multihypothesis prediction, as in our previous systems in [Deligiannis et al. 2011, 2009b, 2012a, 2012c]. Similar to Deligiannis et al. [2011], the hash information corresponds to a downscaled and low-quality H.264/AVC Intra coded version of the original WZ frames. However, conversely to our previous works, the proposed system features a set of down- and up-sampling filters, each of which corresponds to a different trade-off between the involved computational complexity and the RD performance. Moreover, this work analyses the effect of the downscaling factor on the compression performance. Additionally, a proof of optimality of the employed pixel-based multihypothesis motion compensation scheme is included.

Second, the proposed Wyner-Ziv video coding architecture implements a novel strategy to refine the side-information during decoding. To this end, the decoding process is executed in distinct refinement levels that coincide with congregated frequency bands in a $4 \times 4$ DCT block. Every time a refinement level is decoded, a partially decoded WZ frame is available to the decoder and is used to improve the side-information for the next refinement level. By balancing the RD performance with the supplemental computational complexity and induced latency, an optimal strategy to assemble the DCT bands into refinement levels is devised. This approach opposes the scheme in Martins et al. [2009], where the side-information is refined whenever a DCT band is decoded, as well as our previous architecture in Deligiannis et al. [2011], where refinement is carried out only once, that is, after decoding the DC band.

Third, unlike Varodayan et al. [2008] and Badem et al. [2008], side-information refinement is performed after reverting the partially decoded WZ frame to the pixel-domain, thereby avoiding the need for an overcomplete DCT and the significant corresponding computational load at the decoder. We highlight that an in-loop deblocking filter is applied to the partially decoded frame in order to enhance the consistency of the motion field. Such in-loop deblocking is absent from the systems of Deligiannis et al. [2011, 2012c], Ye et al. [2009], and Martins et al. [2009].

Fourth, our system features a novel pixel-based multihypothesis prediction scheme, based on a successively refined version of OBME performed using the partially decoded WZ frames and the references. Instead of merely repeating the basic OBMEC as in Deligiannis et al. [2012c], this novel technique comprises a number of new tools to generate high-quality side-information at consecutive refinement levels. Specifically, as more refinement levels have been completed, the quality of the partially decoded WZ frames increases. This is exploited (i) by gradually decreasing the number of overlapping blocks over consecutive refinement levels and (ii) by introducing a selection procedure at every refinement level that determines for which overlapping blocks motion estimation requires further refinement. Both measures decrease the overall OBME complexity without undermining the prediction performance. Finally, (iii) prior to multihypothesis prediction, a screening procedure is applied to reject spurious outlier predictors. The devised refinement strategy delivers state-of-the-art RD performance, even when a moderate number of refinement levels are employed.

Fifth, an additional original aspect of the Wyner-Ziv video codec presented in this work is the availability of several temporal prediction structures to the decoder. Specifically, besides traditional hierarchical bidirectional temporal prediction, our system also supports a unidirectional prediction structure to target low-delay decoding. Conversely to MCE-based methods [Girod et al. 2005; Li et al. 2007], the hash-based nature of our
system allows for flexible modification of the temporal prediction structure by altering the appropriate reference frame(s). Furthermore, our system is equipped with a new hierarchical unidirectional temporal prediction structure, providing low-delay decoding while simultaneously supporting temporal scalability, a characteristic of particular importance in the context of heterogeneous WMSNs.

Experimental evaluation substantiates that the proposed hash-based Wyner-Ziv codec significantly outperforms state-of-the-art non-successively refined DVC codecs, namely, DISCOVER [Artigas et al. 2007], the hash-based system in Ascenso et al. [2010] and our previous hash-based DVC systems in Deligiannis et al. [2009b] and Deligiannis et al. [2012a]. Moreover, experimental results confirm that the proposed successively refined Wyner-Ziv system surpasses the state-of-the-art successively refined Wyner-Ziv systems of Ye et al. [2009] and Martins et al. [2009]. Apart from delivering superior compression performance over competing schemes, the proposed Wyner-Ziv architecture simultaneously offers features of vital importance to WMSNs, namely, low-cost encoding, low-delay and scalability.

3. THE PROPOSED SUCCESSIVELY REFINED WYNER-ZIV VIDEO CODEC

This section presents the architecture of the proposed hash-driven successively refined Wyner-Ziv (HSRWZ) video codec, the block diagram of which is given in Figure 1.

3.1. Encoding

The incoming video frames, captured by the sensor’s camera, are divided into group of pictures (GOP). The first frame in each GOP, called key frame $K$, is intra-frame coded. To this end, the H.264/AVC Intra codec, configured using the Main profile as in Artigas et al. [2007], is employed. H.264/AVC Intra is selected as it is one of the most efficient intra coding schemes; however, any alternative intra-frame video codec, that is, Motion JPEG, can be employed to code the key frames. The remaining frames in each GOP, which are named WZ frames and denoted by $W$, are encoded in two parts. First, an auxiliary hash is coded after which a WZ bit-stream is generated.

The hash is composed of a low-resolution, coarsely intra-coded description of each WZ frame. The hash formation and compression procedures, which are explicitly detailed in Section 4, are carefully designed to retain low sensor node (encoder) complexity, while providing high compression performance.
Together with the hash data, a Wyner-Ziv bit-stream is formed for each WZ frame. Conversely to other hash-driven TDWZ schemes, for example, [Ascenso et al. 2010], the proposed Wyner-Ziv encoder encodes the original WZ frame, rather than its difference with the reconstructed hash information. This approach maintains low complexity and memory demands at the encoder [Deligiannis et al. 2012b]. Moreover, in order to keep the encoder as simple as possible, the proposed codec does not classify blocks for which a hash information is required, as in Ascenso et al. [2010].

The proposed video codec follows the principles of successively refined Wyner-Ziv coding [Steinberg and Merhav 2004]. Initially, the WZ pixel values are transformed using the $4 \times 4$ separable integer transform of H.264/AVC [Wiegand et al. 2003], which has similar properties as the DCT. Subsequently the obtained DCT coefficients, belonging to 16 distinct DCT bands, that is, $b = \{0, 1, \ldots, 15\}$, are quantized with $2^{16}$ levels using predefined quantization matrices (QMs) as in Artigas et al. [2007] and Deligiannis et al. [2012a]. The quantized symbols of the DCT coefficients are split into bit-planes, forming binary codewords. Note that the proposed system divides the WZ frames in several quality scalability layers (QLs); namely, as the number of decoded bands and bit-planes (per band) increases, the quality of the decoded WZ frame improves. The proposed codec takes advantage of this property by supporting a successively refined side-information creation scheme. Specifically, the quantized coefficients are divided into $L$ side-information refinement levels (SIRLs), that is, $l = \{0, 1, \ldots, L - 1\}$. The coefficients of the DC band form the coarsest side-information stage, denoted by $l = 0$, while each additional SIRL, $l = \{1, 2, \ldots, L - 1\}$, is constructed by congregating the AC bands on a diagonal in a $4 \times 4$ DCT block; see Section 6.1 for explicit details. Per SIRL $l$, the binary codewords of the quantized DCT coefficients belonging to $l$, are fed to an LDPC Accumulate (LDPCA) coder [Varodayan et al. 2006], to realize Slepian-Wolf coding. The derived syndrome bits per WZ bitplane are stored in a bit-plane buffer and transmitted upon the decoder’s request using a feedback channel.

### 3.2. Decoding

At the decoder, the key frames are H.264/AVC Intra decoded and stored in the reference frame buffer. The hash bit-stream is decoded and the reconstructed hash is interpolated in order to be restored to the original frame size. Previously decoded frames and the interpolated hash frame are then used to produce the initial version of the side-information $Y_0$ for the WZ frame; see Section 6.2. This initial side-information is first transformed using the DCT and subsequently used to enable the LPDCA decoding of the bit-planes of the DC coefficient band. To this end, the online correlation channel estimation (CCE) algorithm in Deligiannis et al. [2012a] estimates the conditional statistics of the DC coefficients of the original WZ frame given the DC band coefficients of the side-information frame. We highlight that the algorithm in Deligiannis et al. [2012a] is built in a progressive manner and the correlation estimation is executed again per decoded bit-plane of the band, thus enabling bit-plane-by-bit-plane progressively refined CCE (see CCE loop in Figure 1).

After LDPCA decoding, the derived bit-planes are grouped per quantization index and minimum mean square error (MMSE) [Kubasov et al. 2007] reconstruction is carried out to obtain the decoded DC coefficients. Thereafter, the decoded DC band coefficients are assembled together with the coefficients of the transformed side-information $Y_0$ at the positions of the remaining SIRLs, that is, $l = \{1, 2, \ldots, L - 1\}$ and the inverse discrete H.264/AVC approximation of the DCT is performed to derive the partially decoded WZ frame belonging to the coarsest refinement level.

Next, our novel side-information refinement approach, discussed in Section 6.3, is applied using the partially decoded WZ frame and reference frames from the buffer. The proposed method is run recursively, that is, per SIRL $l = \{1, 2, \ldots, L - 1\}$. The
The proposed HSRWZ codec uses the progressively refined (partially decoded) WZ frame from the previous SIRL, that is, $l - 1$, to produce improved side-information $Y_l$. Using the updated side-information $Y_l$, the quantized coefficients belonging to the SIRL $l$ are LDPCA decoded and MMSE reconstructed. Notice that using this updated $Y_l$ the coefficients belonging to all previous SIRLs, that is, $\{0, 1, \ldots, l - 1\}$, are MMSE reconstructed again, thereby progressively improving the quality of the decoded frame. This strategy corresponds to a side-information refinement loop at the decoder, where every WZ frame is decoded in distinct stages, comprising the different SIRLs (see SIRL loop in Figure 1). Remark that within every SIRL, the decoder produces the necessary soft information from the CCE algorithm in Deligiannis et al. [2012a]. Per decoded bit-plane of a band, the correlation is estimated again, constituting the CCE loop in Figure 1. Hence, similar to Deligiannis et al. [2012c], the proposed HSRWZ features two nested refinement loops, refining both the side-information and the CCE.

In this fashion, as more information is WZ decoded, better versions of the partially decoded WZ frame become available to the decoder. As a consequence, the quality of the side-information $Y_l$ improves for every SIRL $l$. This means that the source $W$ and the side-information $Y_l$ at every SIRL $l = \{0, 1, \ldots, L - 1\}$ can be modeled as a stochastically degraded channel, forming a Markov chain, $W \leftrightarrow Y_{l-1} \leftrightarrow \cdots \leftrightarrow Y_0$. According to inequality (1) in Section 2, the proposed HSRWZ system yields increased RD performance compared to the equivalent non-successively refined WZ system.

After all the DCT bands of every SIRL are decoded, the inverse integer H.264/AVC approximation of the DCT is performed one last time. Finally, after applying an appropriately tuned version of the H.264/AVC deblocking filter, the reconstructed WZ frame is ready for display and stored in the reference frame buffer. As explained in Section 6.5, the deblocking filter is also deployed inside the side-information refinement loop to alleviate the blockiness of the partially decoded WZ frame at every SIRL, thereby enhancing the motion-compensated prediction.

4. HASH FORMATION AND COMPRESSION

In this section, the formation, encoding and decoding of the hash information is described in detail.

4.1. Hash Encoder

At the encoder-side, the original WZ frame’s luma component is first downscaled by a factor $d$, $d \in \mathbb{N}/\{0, 1\}$. After the downscaling operation, the downscaled luma component is encoded using very low bit-rate H.264/AVC Intra coding. The encoder employs a high quantization parameter $QP_H$ and fast rate-distortion optimized mode decision to limit the computational complexity. Analogous to the key frames, the coding of the hash is not confined to H.264/AVC Intra coding. Hence, any intra codec can be employed to this end.

In the proposed system, downscaling can be performed in two different ways. To minimize the encoder’s computational complexity, straightforward down-sampling can be applied as in Deligiannis et al. [2011], at the risk of introducing undesirable aliasing artifacts. Alternatively, a one-dimensional low-pass filter $g(u)$ with a cut-off frequency of $\pi/d$ can be applied to the rows and columns of the input data to band-limit the signal prior to down-sampling, thereby suppressing aliasing artifacts at the expense of higher complexity. Three different downscaling filters, with different computational complexity are supported in our implementation. The first filter is a fixed triangular filter with 3 taps $g(u) = [1/4, 2/4, 1/4]$ and a cut-off frequency of $\pi/2$, which means that it is specifically tailored to handle downscaling by $d = 2$. However, the filter can be employed for any $d = 2^\gamma$, $\gamma \in \mathbb{N}_0$ by repeated downscaling by 2, each time applying filtering and dyadic down-sampling. In terms of complexity, the calculation of every
filtered sample requires only two integer additions and two bit-shifts. Hence, the added complexity over straightforward down-sampling is relatively limited.

The two other downscaling filters, supported in our implementation, are the so-called Lanczos filters. These filters are derived by windowing the infinite-length impulse response of the ideal low-pass filter with cut-off frequency $\pi/d$, as given by $g_{\text{ideal}}(u) = 1/d \cdot \text{sinc}(u/d)$, $u \in \mathbb{Z}$ [Oppenheim et al. 1999], using a Lanczos window [Duchon 1979] $\omega_a(u)$, which is the central lobe of a scaled sinc function, that is,

$$\omega_a(u) = \begin{cases} 
\frac{\sin(\pi \cdot u/a)}{\pi \cdot u/a}, & |u| < a \\
0, & |u| \geq a 
\end{cases} \quad (2)$$

where the parameter $a$ is a positive non-zero integer value which determines the size of the window. The filters’ coefficients are then derived as

$$g_a(u) = g_{\text{ideal}}(u) \cdot \omega_a(u) = \frac{1}{d} \cdot \text{sinc}\left(\frac{u}{a}\right) \cdot \text{sinc}\left(\frac{u}{3 \cdot d}\right), \quad |u| < a. \quad (3)$$

As discussed in Turkowski and Gabriel [1990], the filters derived in this way have been shown to perform very well for image rescaling tasks. In our implementation, either $a = 2 \cdot d$ or $a = 3 \cdot d$ is chosen, which corresponds to the so-called Lanczos2 and Lanczos3 filters [Turkowski and Gabriel 1990]. Furthermore, as in Turkowski and Gabriel [1990], the filter taps obtained from Equation (3) are normalized to ensure unit DC gain, thereby preserving the average luminance after scaling. It is clear that the improved quality generally obtained using the Lanczos filters comes at the expense of a higher encoder/sensor complexity. Specifically, the Lanczos filters have $2 \cdot a - 1$ taps, of which $2 \cdot (a/d - 1)$ are zero. As such, calculating a single filtered value requires $2 \cdot a \cdot (1 - 1/d) + 1$ floating-point multiplications and $2 \cdot a \cdot (1 - 1/d)$ floating-point additions. Note however that, although a generalized floating-point implementation is used here, integer approximations of these filters for particular values of $d$, yielding similar performance, exist as well [Turkowski and Gabriel 1990].

4.2. Hash Decoder

At the decoder-side, the H.264/AVC hash bit-stream is decoded and the reconstructed hash is upscaled to the original WZ frame’s resolution using the technique presented in Deligiannis et al. [2012b]. In short, a Lanczos windowed version (with $a = 3 \cdot d$) of the ideal upsampling filter is employed, namely,

$$h(u) = h_{\text{ideal}}(u) \cdot \omega_a(u) = \text{sinc}\left(\frac{u}{d}\right) \text{sinc}\left(\frac{u}{3 \cdot d}\right), \quad |u| < 3 \cdot d. \quad (4)$$

where $h_{\text{ideal}}(u) = \text{sinc}(u/d)$, $u \in \mathbb{Z}$, is the ideal filter response with cut-off frequency $\pi/d$ and gain $d$ [Oppenheim et al. 1999]. To reduce the computational complexity, the process of up-sampling generally followed by interpolation filtering to generate the upscaled frame has been replaced by an equivalent polyphase filterbank with interpolation filters $h_{\gamma}(u)$, $0 \leq \gamma < d$. The filters $h_{\gamma}(u)$, $0 \leq \gamma < d$ are straightforwardly derived from $h(u)$ as $h_{\gamma}(u) = h(u \cdot d + \gamma)$. As in Turkowski and Gabriel [1990] and similar to the Lanczos downscaling filters, the resulting filter taps are normalized in our implementation to obtain unit DC gain. Note that $h_0(n) = 1$, so that the input samples in the corresponding phase are preserved by the upsampling process.

5. PREDICTION STRUCTURES

Utilizing the decoded and interpolated hash $W$ or the partially decoded WZ frame and up to two reference frames $R_n, \ n = \{0, 1\}$, the decoder of the proposed HSRWZ codec
supports three temporal prediction structures, namely, a hierarchical bidirectional, a unidirectional, and a hierarchical unidirectional motion prediction structure.

5.1. Hierarchical Bidirectional Motion Prediction Structure

In the hierarchical bidirectional (HIB.B) prediction structure—see Figure 2(a)—a structure commonly utilized by DVC systems, for instance, Girod et al. [2005], Artigas et al. [2007], Martins et al. [2009], and Ascenso et al. [2010], the frames in a GOP of size $G$ are divided into $\log_2 G + 1$ temporal levels (TLs), with the coarsest level $TL_0$ consisting of the key frames. Per WZ frame $W_i$, bidirectional motion estimation is performed based on the available hash or the partially decoded frame, using a previous and a future—in display order—decoded frame as references. Such hierarchical prediction implies that the two reference frames always belong to coarser temporal scalability levels.

The hierarchical structure enables temporal scalability of $\log_2 G + 1$ levels, a feature of great importance for heterogeneous sensor networks, in which resources and demands are constantly fluctuating. Moreover, since reference frames from both temporal directions are considered, the HIB.B structure delivers the finest prediction accuracy, especially in case of irregular motion content.

Nevertheless, the hierarchical bidirectional structure introduces a delay of at least one GOP size, in view of the fact that decoding the WZ frame of $TL_1$ is postponed until both key frames of level $TL_0$ are decoded. Depending on the utilized GOP size and the acquisition frame rate, this delay can be prohibitive for some sensor applications. Additionally, this structure increases the memory requirements of the encoder, and
consequently the hardware and power consumption demands of the sensor, since the intermediate WZ frames have to remain stored prior to encoding.

5.2. Unidirectional Motion Prediction Structure
In order to offer low-delay and low-memory distributed video encoding, the proposed codec supports a unidirectional (IP..P) prediction structure, as shown in Figure 2(b). In this configuration, unidirectional motion estimation is carried out per WZ frame, $W_i$, using the two previous frames in display order as references. For the first WZ frame in the GOP only one reference frame is considered, that is, the previous key frame.

Since the IP..P structure enables motion prediction only from past reference frames, its prediction accuracy is inferior compared to bidirectional prediction, especially in the case of high motion content. On the other hand, the temporal distance of reference frames in the IP..P structure is constant, while in the HIB..B structure the temporal distance of references in the coarse temporal levels increases with the GOP length. Hence, the difference in prediction quality between the hierarchical bidirectional and the unidirectional structure diminishes with increasing GOP length.

5.3. Hierarchical Unidirectional Motion Prediction Structure
Although the previously presented unidirectional prediction structured enables low-delay and low-memory compression, it does not support temporal scalability. To simultaneously facilitate both features, the proposed HSRWZ scheme supports a novel hierarchical unidirectional (HIP..P) prediction structure, as shown in Figure 2(c). It is important to emphasize that the proposed hierarchical unidirectional structure, which is inspired by the hierarchical inter frame prediction utilized in H.264/AVC [Wan et al. 2009], has never been introduced before in any contemporary DVC scheme. In the presented HIP..P structure, unidirectional motion estimation is performed per WZ frame $W_i$, using up to two past reference frames that belong to the same or to a coarser TL. For the first WZ frame in each TL only one reference frame is available, that is, the key frame of the GOP. Hence, for a GOP size of 2, the proposed HIP..P prediction structures boils down to the IP..P structure. Moreover, similar to the HIB..B structure, the temporal distance of reference frames at the coarser TLs increases with the GOP size. Therefore, when the GOP size increases the prediction quality of the HIP..P structure diminishes compared to the IP..P structure. This is the price to pay for temporally scalable, low-delay and low-memory WZ video coding.

6. SUCCESSIVELY REFINED WYNER-ZIV DECODING
This section details the steps of our novel successively refined Wyner-Ziv video decoder. In short, the proposed strategy decodes the WZ frames in distinct stages, called SIRLs. At the coarsest SIRL, that is, $l = 0$, the decoder produces initial side-information $Y_0$ using the interpolated hash and up to two references frames $R_n$, $n = \{0, 1\}$, assigned according to one of the prediction structures in Section 5. Then, at every extra SIRL $l \in \{1, 2, \ldots, L - 1\}$, the proposed technique produces a refined version $Y_l$ of the side-information using the partially decoded WZ frame $\tilde{W}_{l-1}$, the side-information $Y_{l-1}$ from the previous SIRL, that is, $l - 1$, and the reference frames $R_n$.

6.1. Construction of Refinement Levels
The optimal number of SIRLs in the proposed system is a trade-off between RD performance, forced motion estimation and compensation operations, the required soft channel decoding computations, and the imposed structural delay. In particular, the more SIRLs are introduced, the more decoding complexity is associated with the generation of side-information, mainly due to reinitiated motion estimation and compensation. On the other hand, the more the quality of the side-information increases, the more...
the RD performance of the codec improves. In addition, increasing the quality of side-information reduces the complexity of Slepian-Wolf decoding, since the LDPCA decoder requires less soft decoding runs to decode the WZ information. It is also noteworthy that increasing the number of SIRLs \( L \) introduces additional structural latency in the codec. Namely, as the proposed system deploys a feedback channel for optimal rate control, the decoding of every next refinement level has to wait until the information from all the previous refinement levels has been sequentially decoded.

In this regard, examination has revealed that side-information refinement per DCT coefficient’s bit-plane significantly increases the motion estimation complexity and the codec’s structural latency, while lacking adequate support for an accurate motion search. Similarly, side-information refinement per decoded DCT band, as in Martins et al. [2009] and Abou-Elailah et al. [2011], notably increases the codec’s structural delay due to sequential feedback-channel-based decoding.

In order to meet the optimal refinement trade-off, the proposed HSRWZ codec constructs SIRLs \( l = \{0, 1, \ldots, L - 1\} \) by grouping the frequency bands on every diagonal in a \( 4 \times 4 \) DCT transform block\(^1\). In this context, the highest number of SIRLs corresponds to \( L = 6 \), as in Figure 3(a). As shown in Section 7.2, further grouping of the highest AC coefficients, as in Figure 3(b), does not notably influence the RD performance, while the reduced number of SIRLs (i.e., \( L = 4 \)) limits the structural latency of HSRWZ decoding and promotes the practical applicability of the scheme.

### 6.2. Initial Side-Information Generation

To decode the DC coefficient band of the WZ frame, that is, the SIRL \( l = 0 \), the initial side-information generation comprises hash-based overlapped block motion estimation (OBME) followed by probabilistic motion compensation (PMC), as in Deligiannis et al. [2011]. Hash-based OBME delivers side-information of higher quality with respect to MCI-based systems [Ascenso et al. 2006], [Artigas et al. 2007], in particular when the video material contains complex motion [Deligiannis et al. 2011].

For completeness of the presentation, we hereby provide an overview of the hash-based side-information generation method in Deligiannis et al. [2011]. However, this section also includes a novel proof of optimality of the employed PMC technique, which was not presented in Deligiannis et al. [2011].

First, prior to the initial OBME, the reference frames \( R^n, n = \{0, 1\} \), undergo the same down-sampling and interpolation operation as employed during the formation of the hash frame (see Section 4), yielding their filtered versions \( \tilde{R}^n, n = \{0, 1\} \). This improves the consistency of the initial motion vectors.

---

\(^1\)This grouping is inspired by the distribution of the frequencies in the \( 4 \times 4 \) DCT transform block.

\(^2\)The highest coefficient band, that is, the band composed of the coefficients at position \( (3,3) \) in each \( 4 \times 4 \) DCT transform block, is not WZ coded based on the WZ quantization matrices [Artigas et al. 2007].

ACM Transactions on Sensor Networks, Vol. 10, No. 2, Article 21, Publication date: January 2014.
For clarity, in the remainder of the article we adhere to the following notations. Let \( \hat{W}(i, j) \), \( \hat{R}_n(i, j) \) denote the pixel values at position \((i, j)\), within the decoded and interpolated WZ hash frame and the filtered versions \( \hat{R}_n \), of the reference frames, respectively. Analogously, \( Y(i, j) \), \( R_n(i, j) \) represent the pixel values at position \((i, j)\), in the side-information frame \( Y \) and the unfiltered reference frames \( R_n \), respectively. Furthermore, denote by \( Y_p(r, s) \), \( \hat{R}_n^p(r, s) \), \( \hat{W}_p(r, s) \), and \( \hat{R}_n^p(r, s) \), the pixel values at position \((r, s)\), \( 0 \leq r, s \leq B - 1 \) inside the block of size \( B \times B \) pixels with top-left coordinates \( p = (p_1, p_2) \) in the \( Y, R_n, \hat{W}, \hat{R}_n \) frames, respectively.

To perform OBME, every interpolated hash frame \( \hat{W} \) is divided into overlapping blocks of \( B \times B \) pixels, denoted by \( \hat{W}_p \), using an overlap step size of \( \varepsilon \) pixels with \( \varepsilon \in \mathbb{Z}_+ \), \( 1 \leq \varepsilon < B \) [Deligiannis et al. 2009b, 2011]. In this way, a total number of \((H \cdot V)/\varepsilon^2\) overlapping blocks is created, where \( H, V \) are the horizontal and vertical dimensions of the frame. For every overlapping block \( \hat{W}_p \), integer-pel motion estimation is executed, within a specific search range, in each of the filtered reference frames \( \hat{R}_n \). The employed block matching criterion minimizes the sum of absolute differences (SAD) between \( \hat{W}_p \) and the best matching block \( \hat{R}_n^{p+w} \), where \( \mathbf{v}^n = (v_1^n, v_2^n) \) is the derived motion vector per filtered reference frame \( \hat{R}_n \). When two reference frames are used (see Section 5), OBME has identified two candidate motion vectors \( \mathbf{v}^{n=0}, \mathbf{v}^{n=1} \), per overlapping block \( \hat{W}_p \). The algorithm then retains the best temporal motion vector, that is, \( \mathbf{v}^T = (\mathbf{v}^{n=0} \text{ or } \mathbf{v}^{n=1}) \), per overlapping block \( \hat{W}_p \). This selection minimizes the SAD, or in case the SAD values are equal, the motion vector with the minimum displacement is selected. In this way, dubious motion vectors, are sorted out, thereby improving the quality of the produced side-information. The derived motion field is then applied to the unfiltered reference frames \( R_n, n \in \{0, 1\} \) to generate the initial side-information.

As a result of OBME, each pixel \( Y(i, j) \) in the side-information frame belongs to several overlapping blocks \( Y_{p_k}, k = \{0, 1, \ldots, K - 1\} \). Per overlapping block, the best temporal predictor \( R_{p_k + \mathbf{v}^T} \) has been determined. Therefore, each pixel in the side-information frame has a number of candidate temporal pixel predictors, \( R_{p_k + \mathbf{v}^T}(r_k, s_k) \), being the co-located pixels in every best matching temporal predictor block [Deligiannis et al. 2011]. The temporal predictors for every pixel are combined (by means of PMC) to create a single prediction to serve as side-information.

In the following, we derive the PMC method that is optimal in the mean squared error (MSE) sense. Let \( \psi_{(i, j)} = (\psi^0, \psi^1, \ldots, \psi^{(i, j)} \ldots) \) denote the \( Z \) distinct values of the \( K \) candidate temporal predictors \( R_{p_k + \mathbf{v}^T}(r_k, s_k), k = \{0, 1, \ldots, K - 1\} \), for the side-information pixel \( Y(i, j) \). That is, each predictor value \( \psi^k \) appears \( N_k \) times in the set \( R_{p_k + \mathbf{v}^T}(r_k, s_k) \), \( k = \{0, 1, \ldots, K - 1\} \) counting \( K \) candidate temporal predictors. Also, denote by \( W, Y, \psi \) the random variables (RVs) corresponding to the WZ frame, the side-information, and the candidate predictor pixel values, respectively. Moreover, let \( f_{W|\psi}(w|\psi = \psi^k) \) represent the conditional probability density function (PDF) of the source RV given the candidate predictor value \( \psi^k \). This conditional PDF is typically modeled by a peaked distribution (e.g., Laplacian) centered on the candidate predictor value \( \psi^k \) [Deligiannis et al. 2009a]. The following holds:

**Lemma 1**. The expected value of the elements of \( \psi_{(i, j)} \) corresponds to the MSE optimal side-information value \( y_{opt} \) for the original WZ sample \( w \).

**Proof.** The expected mean squared error when predicting an original sample value \( w \) by \( y \), given the collection of conditional PDFs \( f_{w|\psi}(w|\psi = \psi^k) \), as defined by the
elements of $\psi_{(i, j)}$, is expressed by

$$MSE(y) = \sum_{\zeta=0}^{Z-1} p_{\Psi}(\Psi = \psi_\zeta) \int_{-\infty}^{+\infty} (w - y)^2 f_{W|\Psi}(w | \Psi = \psi_\zeta) \, dw,$$

(5)

where $p_{\Psi}(\Psi)$ represents the marginal distribution of the candidate predictor RV. By putting the partial derivative $\partial MSE(y)/\partial y$ to zero and solving for $y$ we derive

$$\sum_{\zeta=0}^{Z-1} p_{\Psi}(\Psi = \psi_\zeta) \int_{-\infty}^{+\infty} 2(w - y)(w - y) f_{W|\Psi}(w | \Psi = \psi_\zeta) \, dw = 0$$

(6)

$$\Rightarrow \sum_{\zeta=0}^{Z-1} p_{\Psi}(\Psi = \psi_\zeta) \int_{-\infty}^{+\infty} y f_{W|\Psi}(w | \Psi = \psi_\zeta) \, dw = \sum_{\zeta=0}^{Z-1} p_{\Psi}(\Psi = \psi_\zeta) \int_{-\infty}^{+\infty} w f_{W|\Psi}(w | \Psi = \psi_\zeta) \, dw.$$

(7)

Bearing in mind that $f_{W|\Psi}(w | \Psi = \psi_\zeta)$ is modeled by a peaked distribution centered on the candidate predictor value $\psi_\zeta$, (7) leads to

$$\Rightarrow y_{opt} = \sum_{\zeta=0}^{Z-1} \psi_\zeta p_{\Psi}(\Psi = \psi_\zeta),$$

(8)

which ends the proof. $\square$

Hence, from Lemma 1, we have $y_{opt} = \sum_{\zeta=0}^{Z-1} N_{\zeta} \psi_\zeta / K$. Therefore, during compensation, the estimated value of a pixel $Y_0(i, j)$, in the side-information frame $Y_0$, is calculated as the mean value of the predictor pixel values determined by OBME, namely,

$$Y_0(i, j) = \frac{1}{K} \sum_{h=0}^{K-1} R_{p_i + v_j}(r_h, s_k)$$

(9)

After decoding the DC frequency band with the initial side-information, the decoder handles the finer refinement stages, as depicted in Figure 4. At each consecutive level, our novel refinement process encompasses the following components.

6.3. Successively Refined Side-Information Generation

In order to produce high-quality side-information for the finer SIRLS, that is, $l = \{1, \ldots, L - 1\}$, the proposed architecture features a novel version of the OBME concept, referred to as successively refined OBME (SR-OBME).

In contrast to alternative methods, [Badem et al. 2008; Varodayan et al. 2008], which refine the side-information in the transform-domain, the proposed refinement technique operates in the pixel-domain. In this manner, we avoid a complex overcomplete DCT representation of the reference frames, which would add undesirable complexity to the decoder. Therefore, at SIRL $l \in \{1, 2, \ldots, L - 1\}$ in order to generate $Y_l$, the DCT transformed side-information of the previous level, that is, $Y_{l-1}$, is updated with the decoded coefficients from all the previous SIRLS $\{0, 1, \ldots, l - 1\}$ and an inverse DCT is performed, yielding the partially decoded WZ frame $\tilde{W}_{l-1}$.

6.3.1. Overlapping Step Increment. In the subsequent step the partially decoded WZ frame $\tilde{W}_{l-1}$ is divided into overlapping blocks and SR-OBME is performed using the reference frames $R^n$, $n = \{0, 1\}$. Contrary to the initial side-information creation (see
Section 6.2), the reference frames are not low-pass filtered in the refinement loop, since the high frequencies in the partially decoded frames are successively restored.

In OBME [Deligiannis et al. 2009b; Deligiannis et al. 2011] the value of the overlapping step $\varepsilon$ controls the trade-off between the complexity and prediction quality of the presented OBME method. In particular, on the one hand, a small overlapping step $\varepsilon$ yields a large number of overlapping blocks, and in turn an accurate estimation of the true motion field. However, decreasing the value of $\varepsilon$ increases the computational complexity of the OBME technique. On the other hand, a large $\varepsilon$ reduces the prediction quality as the number of overlapping blocks, and in turn the number of predictors per pixel, diminish. Then again, a large $\varepsilon$ has a beneficial influence on the complexity of a refinement operation, since less block comparisons are required.

The quality of the partially decoded frame improves as more and more SIRLs have been completed. As a result a less dense motion field is required to further reduce the motion estimation uncertainty. Therefore, this is exploited in the proposed SR-OBME method by controlling the value of $\varepsilon$ at every consecutive refinement level. In this way, the overall complexity is constrained, while the prediction performance is not jeopardized. Specifically, the overlapping step in the proposed SR-OBME is successively increased from one SIRL to the next, as follows:

$$
\begin{align*}
\varepsilon_l &= \varepsilon_{l-1} + l + 1, & \text{if } \varepsilon_l < B \\
\varepsilon_l &= \varepsilon_{l-1}, & \text{otherwise}
\end{align*}
$$

(10)

where $B$ is the dimension of each block. The condition in (10) ensures that the blocks are still overlapping at every SIRL. This means that pixel-based multihypothesis MSE-optimal PMC, as in Section 6.2, can be applied after SR-OBME at each SIRL.

6.3.2. OBME with Selection of Overlapping Blocks for Refinement. Per SIRL $l = \{1, 2, \ldots, L - 1\}$, each overlapping block $\hat{W}_p^{l-1}$ in the partially decoded WZ frame $\hat{W}^{l-1}$ is compared with the co-located block $Y_p^{l-1}$ in the corresponding side-information frame $Y^{l-1}$. If the mismatch is high, meaning that the decoder has “corrected” a significant
amount of DCT-domain information, the motion vector of this overlapping block requires further refinement at the current SIRL \( l \). Specifically, using the SAD criterion and a predefined threshold value \( T_A \), the motion vector of an overlapping block \( \tilde{W}_l^{p-1} \) is refined at the current SIRL if

\[
SAD_p = \sum_{r=0}^{B-1} \sum_{s=0}^{B-1} |\tilde{W}_l^{p-1}(r, s) - Y_l^{p-1}(r, s)| \geq T_A.
\] (11)

Otherwise, the motion vector defined at the previous SIRL for this overlapping block is considered accurate. Empirically, \( T_A = 250 \) has been identified as an appropriate threshold value, both in terms of low computational load and high RD performance.

6.3.3. Motion Compensation with Candidate Predictor Filtering. Since in SR-OBME the blocks used for motion estimation are overlapping (see Section 6.3.1), every pixel \( Y_l(i, j) \) in the refined side-information frame \( Y_l \) belongs to several overlapping blocks, that is, \( Y_l(i, j), k = \{0, 1, \ldots, K-1\} \). For each overlapping block, of which the motion vector was refined in the SIRL \( l \) (see Section 6.3.2), SR-OBME has redetermined the best temporal predictor \( \tilde{R}^{a}_{p_k+v_k^l} \) in the reference frames. For the overlapping blocks for which the motion was not refined, the best predictor is defined by the collocated value in block \( \tilde{W}_l^{p-1} \) in the partially decoded frame. In this way, every pixel in the refined side-information \( Y_l(i, j) \) has a number of candidate pixel predictors, denoted by \( P^l_p(r_k, s_k), k = \{0, 1, \ldots, K-1\} \), which are classified in two types. Namely, if the motion vector for the overlapping block \( Y_l^{p_k} \) was refined, then \( P^l_p(r_k, s_k) = \tilde{R}^{a}_{p_k+v_k^l}(r_k, s_k), n = \{0, 1\} \). Otherwise, if the motion vector was not refined, then \( P^l_p(r_k, s_k) = \tilde{W}_l^{p-1}(r_k, s_k) \). Observe that for each pixel \( Y_l(i, j) \) all the nonrefined pixel predictors \( \tilde{W}_l^{p-1} \) actually correspond to the pixel value of the partially decoded frame \( \tilde{W}_{l-1} \) at position \( (i, j) \).

As more refinement levels are decoded, the partially decoded frame \( \tilde{W}_{l-1} \) ever more resembles the original WZ frame. This decreases the credibility of temporal predictors \( P^l_p(r_k, s_k) = \tilde{R}^{a}_{p_k+v_k^l} \) that significantly deviate from the corresponding pixel value in the partially decoded frame. Hence, prior to pixel-based compensation, the pixel predictors taken from the reference frames undergo a filtering process in order to cast out dubious outliers, which might add spurious noise in the refined side-information. To define an appropriate selection process, we note that a limited filtering range would potentially exclude valuable candidate temporal pixel predictors. On the other hand, a wide filtering range might include outliers. Therefore, per SIRL \( l = \{1, 2, \ldots, L-1\} \), the filtering range for each motion-compensated pixel is adaptively defined based on the distance between the pixel value in the partially decoded frame \( \tilde{W}_{l-1} \) and in the side-information frame \( Y_{l-1} \).

In particular, every candidate pixel predictor, \( P^l_p(r_k, s_k) = \tilde{R}^{a}_{p_k+v_k^l}(r_k, s_k) \) for the side-information pixel \( Y_l(i, j) \) is considered to be trustworthy only if

\[
|\tilde{W}_{l-1}(i, j) - \tilde{R}^{a}_{p_k+v_k^l}(r_k, s_k)| \leq \Delta_{\min, \tilde{W}_{l-1}(i, j)} \times F_A, \quad (12)
\]

where \( \tilde{W}_{l-1}(i, j) \) and \( Y_{l-1}(i, j) \) are the pixel values in the partially decoded frame and the side-information from the previous refinement level, respectively, \( \Delta_{\min} \) is a predefined value ensuring a minimum filtering range and \( F_A \) is a predefined multiplication factor. If Equation (12) is not satisfied, the temporal pixel predictor is considered an outlier and is replaced by \( \tilde{W}_{l-1}(i, j) \). The absolute difference \( |\tilde{W}_{l-1}(i, j) - Y_{l-1}(i, j)| \) gradually decreases as more SIRLs are completed. Hence, the coefficients \( \Delta_{\min}, F_A \) have been
properly set to ensure a suitable filtering range. After experimentation, the best RD performance, was obtained by putting $\Delta_{\text{min}} = 15$ and $F_A = 4$.

After candidate temporal predictors filtering, every pixel $Y_l(i, j)$ in the refined side-information, is compensated in an MSE-optimal manner by the mean value of the predictor pixel values, namely,

$$Y_l(i, j) = \frac{1}{K} \sum_{k=0}^{K-1} P_k^l(r_k, s_k).$$  \hspace{1cm} (13)

6.4. Decoded Coefficients Refinement

The refined side-information frame $Y_l$ is DCT transformed and subsequently employed in bit-plane-per-bit-plane progressively refined CCE [Deligiannis et al. 2012a]. Based on the correlation estimate the required soft-input information is extracted to LDPCA decode the bit-planes of the quantized WZ DCT coefficients belonging to SIRL $l$. After LDPCA decoding, the DCT transformed side-information frame $Y_l$ is used, together with the obtained correlation estimate, to optimally MMSE reconstruct the DCT coefficients of the current SIRL. The updated side-information frame $Y_l$ is also used to refine the reconstruction of all the coefficients that belong to the previous levels, that is, $\{0, 1, \ldots, l - 1\}$. To this end, CCE [Deligiannis et al. 2012a] and MMSE reconstruction are executed again for the bands in the previous SIRLs, using the updated $Y_l$ and the already decoded bit-planes.

In this fashion, the reconstruction of all the decoded coefficients is progressively improved upon as the side-information is refined. This in turn yields a better partially decoded frame $\tilde{W}_l$, which enables an even more accurate refinement of the side-information $Y_{l+1}$ at the next SIRL. This iterative approach yields an improved final reconstruction at the last SIRL $L - 1$, where all the quantized coefficients, that have been decoded at all SIRLs, are refined using the lastly updated side-information $Y_{L-1}$.

6.5. In-loop Deblocking Filter

Due to quantization, the DCT-based coding of the WZ frames might introduce blocking artifacts into the partially or final decoded frame. These blocking artifacts are not only visually displeasing (in the final decoded WZ frame), but they might hamper the SR-OBME used in the successive refinement process as well. To remedy this problem, the adaptive deblocking filter [List et al. 2003] of the H.264/AVC standard [Wiegand et al. 2003] is applied in the refinement loop.

The H.264/AVC deblocking filter [List et al. 2003] first derives a global strength parameter $B_f = \{0, 1, 2, 3, 4\}$ for each block boundary based on the properties of the two neighboring blocks under consideration. These properties include the type of the encompassing macroblock (intra/inter), the presence of non-zero residual data, the similarity of the motion vectors, etc. The actual filtering strength for each line of pixels crossing a block boundary is determined based on the global strength $B_f$ and the quantization parameter $QP$. High $B_f$-values correspond to situations likely to produce blocking artifacts. Therefore, the actual filter strength increases with increasing $B_f$. Similarly, the higher the $QP$, the more likely prominent blocking artifacts become and the stronger the applied filtering.

In our HSRWZ codec, the H.264/AVC deblocking filter is employed according to the following procedure. In essence, the filter can be completely controlled by providing the parameters $B_f$ and $QP$. To determine an appropriate value for $B_f$, we remark that the decoding process of our codec is similar to the case where all blocks are intercoded in H.264/AVC. Indeed, each block in the side-information signal more closely resembles the typical result of motion-compensated prediction than that of intra-prediction.

ACM Transactions on Sensor Networks, Vol. 10, No. 2, Article 21, Publication date: January 2014.
As a consequence, values $B_f = 4$ and $B_f = 3$ can be ruled out as these apply to cases where one of the blocks is intra coded. Furthermore, the WZ information can be seen to correspond to residual data being added to a motion-compensated prediction in H.264/AVC inter coding. Based on this reasoning, the global strength is set to $B_f = 2$, the value selected in H.264/AVC when both neighboring blocks belong to inter-coded macroblocks and one of the blocks has coded residuals [List et al. 2003]. The optimal value for the remaining parameter $QP$ was determined by training using a rich set of video sequences, independent from those used in the experimental evaluation, after which the value $QP = 30$ was retained.

7. PERFORMANCE EVALUATION

The performance of the proposed HSRWZ codec is evaluated and compared against several state-of-the-art codecs. Firstly, the use of straightforward down-sampling to form the hash information is justified. Secondly, the proposed refinement strategy is examined and thirdly, the impact of the hash down-sampling factor on the performance is discussed. Fourthly, the compression capacity of the proposed codec employing different motion prediction structures is investigated. Fifthly, RD performance comparisons are conducted between the proposed codec and numerous relevant state-of-the-art video codecs, namely, the DISCOVER [Artigas et al. 2007] codec, the hash-based DVC system of Ascenso et al. [2010], our previous hash-based codecs in Deligiannis et al. [2009b, 2012a] and the conventional H.264/AVC Intra codec. Additionally, the proposed codec is compared against the successively refined TDWZ systems of Ye et al. [2009] and Martins et al. [2009]. Finally, this section elaborates on the encoding and decoding complexity of the proposed codec.

Regarding the OBME configuration, the motion estimation algorithm was assigned an overlap step size $\varepsilon = 2$, the size of the overlapping blocks was set to $B = 16$ and exhaustive motion search was performed at integer-pel accuracy within a search range of $\pm 20$ pixels. Experiments were carried out on all frames of the Carphone, Foreman, Ice and Soccer sequences, at QCIF resolution and a frame rate of 15Hz. These test video sequences exhibit varied camera and motion attributes. Specifically, Carphone and Foreman are characterized by medium motion content, whereas the latter includes a complex camera panning. Ice exhibits rather high motion content with a large portion of homogeneous areas (ice). The Soccer sequence contains complex motion content with accelerations and frequent camera panning. Four GOP sizes are considered, that is, GOP of size 2, 4, 8, and 16. The results are depicted in terms of the Bjøntegaard Delta (BD) [Bjøntegaard 2001] metric, which is based on the bit-rates and average luma PSNR values of four RD points, corresponding to QM 1, 5, 7 and 8 of [Artigas et al. 2007]. Even though the proposed codec supports both luma and chroma coding, results are only provided for the luma component to allow a fair comparison with prior art [Artigas et al. 2007], [Ascenso et al. 2010]. Per RD point, the H.264/AVC Intra quantization parameters (QPs) for the key and the hash frames are chosen to maintain quasi-constant decoded frame quality.

7.1. Straightforward Hash Down-sampling Assessment

This section studies the impact of the supported downscaling approaches for hash formation on the RD performance of the proposed HSRWZ codec. Table I summarizes the Bjøntegaard rate and PSNR deltas of the obtained RD performance when using the other supported down-sampling methods, namely, straightforward down-sampling (denoted by “Straight”), the triangular filter, and the Lanczos2 filter, compared to the performance achieved by using the Lanczos3 filter. Remark that in all the evaluated approaches decoder-side hash interpolation is performed using the Lanczos3 interpolation filter.
Table I. Bjøntegaard Deltas on the RD Performance of the Proposed Codec When Using Several Hash Down-sampling Methods Compared to the Performance Obtained Using the Lanczos3 Downscaling Filter

<table>
<thead>
<tr>
<th>Method</th>
<th>Foreman Sequence</th>
<th>Soccer Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GOP 2</td>
<td>GOP 8</td>
</tr>
<tr>
<td></td>
<td>ΔR (%)</td>
<td>ΔPSNR (dB)</td>
</tr>
<tr>
<td>Straight</td>
<td>0.55</td>
<td>-0.035</td>
</tr>
<tr>
<td>Triangular</td>
<td>0.54</td>
<td>-0.034</td>
</tr>
<tr>
<td>Lanczos2</td>
<td>0.22</td>
<td>-0.013</td>
</tr>
</tbody>
</table>

Table II. Bjøntegaard Improvements on the RD Performance of the Presented Codec With \(d = 2\) By Using the Proposed Side-Information Refinement Technique Employing Several Refinement Levels With Respect to the Performance Obtained Without Refinement

<table>
<thead>
<tr>
<th>Refinement Levels Number</th>
<th>Foreman Sequence</th>
<th>Soccer Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GOP 2</td>
<td>GOP 8</td>
</tr>
<tr>
<td></td>
<td>ΔR (%)</td>
<td>ΔPSNR (dB)</td>
</tr>
<tr>
<td>2 Levels</td>
<td>-5.57</td>
<td>0.353</td>
</tr>
<tr>
<td>3 Levels</td>
<td>-7.14</td>
<td>0.461</td>
</tr>
<tr>
<td>4 Levels</td>
<td>-7.22</td>
<td>0.466</td>
</tr>
<tr>
<td>5 Levels</td>
<td>-7.28</td>
<td>0.470</td>
</tr>
<tr>
<td>6 Levels</td>
<td>-7.32</td>
<td>0.473</td>
</tr>
<tr>
<td>Full Refinement</td>
<td>-8.20</td>
<td>0.533</td>
</tr>
</tbody>
</table>

Experimental results are depicted for the Foreman and Soccer sequences, using a hash downscaling factor \(d = 2\), employing an HIB..B prediction structure, and considering GOPs of sizes 2 and 8. The results corroborate that the best RD performance is acquired when the Lanczos3 filter is employed. Yet, the loss in compression performance due to employing the three alternative methods is minor. Therefore, since the impact on the RD performance of the system does not dwarf the computational complexity incurred by the use of downscaling filters, straightforward down-sampling is preferred.

### 7.2. Refinement Strategy Evaluation

In the following, we study the influence of the proposed successive refinement methodology on the performance of our HSRWZ codec. Table II assesses the compression performance of the proposed refinement technique vis-à-vis the codec’s configuration without refinement; namely, when the entire WZ frame information is coded in one level using the initial side-information. Comparative tests are carried out for the Foreman and Soccer sequences at a GOP of size 2 and 8. The proposed system employs an HIB..B prediction structure and the hash downscaling factor is set to \(d = 2\). The number of SIRLs varies from two, that is, one side-information refinement operation is performed after decoding the DC coefficient of each DCT transform block, to six SIRLs. In addition, the performance obtained by refining the side-information upon decoding of each coefficient band—this strategy is denoted as “Full Refinement” in Table II—is also included.

The experimental results corroborate the benefit of successive refinement on the RD performance. This is attributed to the progressive improvement of the side-information produced at the decoder, which in turn reduces the Slepian-Wolf rate to decode the coefficients of the current refinement level and improves the reconstruction of the already Slepian-Wolf decoded coefficients. The obtained RD gains increase with...
we now evaluate the impact of the parameters \( T_A \) and \( \Delta_{min} \), \( F_A \) on the performance of the proposed refinement strategy. The proposed HSRWZ codec with four SIRLs is configured with \( d = 4 \) and the HIB..B prediction structure. Table IV tabulates the BD rate and PSNR between the proposed system using several \( T_A \) threshold values and the system with \( T_A = 0 \), that is, the system that performs side-information refinement for every overlapping block. For each \( T_A \) threshold, Table IV also reports the reduction of the average decoding execution time, that is, the average decoding speedup. The latter is given by \( \Delta T = (T_{\text{without, } T_A} - T_{\text{with, } T_A})/T_{\text{without, } T_A} \) (\%), where \( T_{\text{with, } T_A} \) and \( T_{\text{without, } T_A} \) are the decoding execution times (averaged over all four RD points) of the proposed codec with and without using a \( T_A \) threshold. It can generally be observed that increasing the value of \( T_A \) reduces the decoding complexity at the expense of lower RD performance. This is reasonable as increasing the \( T_A \) threshold leads to a lower amount of motion-compensated overlapping blocks in the refinement process. However, a high \( T_A \) value
Table V. Bjøntegaard Deltas on the RD Performance of the Proposed Codec by Using Different $(\Delta_{\min}, F_A)$ Pairs in the Candidate Predictor Filtering Method With Respect to the Performance Obtained Without Filtering

<table>
<thead>
<tr>
<th>$(\Delta_{\min}, F_A)$</th>
<th>Foreman GOP2 $\Delta R$ (%)</th>
<th>Foreman GOP8 $\Delta R$ (%)</th>
<th>Foreman GOP2 $\Delta$PSNR (dB)</th>
<th>Foreman GOP8 $\Delta$PSNR (dB)</th>
<th>Soccer GOP2 $\Delta R$ (%)</th>
<th>Soccer GOP8 $\Delta R$ (%)</th>
<th>Soccer GOP2 $\Delta$PSNR (dB)</th>
<th>Soccer GOP8 $\Delta$PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5, 1)</td>
<td>15.12</td>
<td>17.01</td>
<td>−0.86</td>
<td>−0.96</td>
<td>6.37</td>
<td>9.28</td>
<td>−0.31</td>
<td>−0.45</td>
</tr>
<tr>
<td>(10, 2)</td>
<td>8.78</td>
<td>9.52</td>
<td>−0.52</td>
<td>−0.63</td>
<td>3.72</td>
<td>6.43</td>
<td>−0.19</td>
<td>−0.32</td>
</tr>
<tr>
<td>(15, 4)</td>
<td>−1.67</td>
<td>−3.14</td>
<td>0.09</td>
<td>0.16</td>
<td>−3.84</td>
<td>0.24</td>
<td>−3.95</td>
<td>0.31</td>
</tr>
<tr>
<td>(20, 6)</td>
<td>−0.87</td>
<td>−1.59</td>
<td>0.05</td>
<td>0.08</td>
<td>−1.94</td>
<td>0.12</td>
<td>−3.31</td>
<td>0.22</td>
</tr>
</tbody>
</table>

may have a negative impact on the decoding complexity, for instance, Foreman GOP2 $T_A = 1500$. This is due to the fact that as the quality of the side-information drops more Slepian-Wolf decoding iterations are required, which leads to higher decoding complexity. All in all, the results in Table IV show that the selected value $T_A = 250$ results in a negligible RD performance drop of up to 0.014 dB, while improving the decoding speed by more than 17% in all cases.

Concerning the configuration of the parameters involved in the candidate predictor filter in Section 6.3.3, Table V reports the compression performance for different $(\Delta_{\min}, F_A)$ pairs with respect to the version without filtering. The results corroborate that a restricted filtering range reduces the RD efficiency due to excluding valuable temporal pixel predictors in the refinement process. Conversely, a broad filtering range, for instance, $(\Delta_{\min}, F_A) = (20, 6)$ leads to a limited performance improvement owing to including untrustworthy outliers in the predictor set. Table V shows that a filtering range determined by $(\Delta_{\min}, F_A) = (15, 4)$ yields the best performance.

7.3. Hash Downscaling Factor Assessment

Figure 5 illustrates the compression performance of the proposed system with and without side-information refinement for two hash downscaling factor values, that is, $d = 2$ or $d = 4$. The proposed system employs the HIB+ prediction structure and the results for the Foreman and Soccer sequences at GOPs of sizes 2 and 8 are depicted. To benchmark, the RD performance of DISCOVER has also been included.

When side-information refinement is switched off, the results show that the proposed codec achieves significantly higher RD performance when using $d = 2$. Using $d = 2$ brings average BD rate savings of 10.43% and 17.05% over the codec’s configuration using $d = 4$, in Foreman, GOP2 and 8, respectively. In Soccer, the average BD rate gains are 3.74% and 4.72% for a GOP of size 2 and 8, respectively. Intuitively, when $d$ is increased, less accurate hash information is conveyed to the decoder, thus leading to lower side-information quality. Although reducing the hash frame dimensions diminishes the hash rate overhead, the overall compression performance of the system drops. Moreover, the impact of reducing $d$ on the RD performance of the proposed codec increases when the length of the GOP increases, since it becomes more difficult to generate accurate side-information. Also, the effect of the $d$ factor on the performance of the proposed codec is strongly influenced by the video content. In particular, in case of video content with plenty of spatial details (e.g., Foreman), increasing the $d$ factor reduces the RD performance more compared to the case of video content with smooth spatial features (Soccer). In the former case, a lot of high frequency information is removed from the decoded hash, which lessens the accuracy of the side-information. Notice that, except for the case of Foreman, GOP2, $d = 4$, the proposed codec without refinement significantly surpasses the state-of-the-art DISCOVER codec for all other test conditions.

Interestingly, once side-information refinement is turned on, the results demonstrate that the proposed codec delivers similar RD performance for $d = 2$ and $d = 4$. Using
Fig. 5. RD performance evaluation of the proposed codec, with and without successive refinement, in regard to the hash downscaling factor value $d$: Foreman (a) GOP2, (b) GOP8, and Soccer (c) GOP2, (d) GOP8. The HIB.B prediction structure is employed.
Progressively Refined Wyner-Ziv Video Coding for Visual Sensors

Table VI. Bjøntegaard Deltas of the Performance of the Proposed Codec with Unidirectional Prediction Structures Compared to that Obtained With the Hierarchical Bidirectional Prediction Structure

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Prediction structure</th>
<th>GOP 2</th>
<th>GOP 4</th>
<th>GOP 8</th>
<th>GOP 16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ΔR (%)</td>
<td>ΔPSNR (dB)</td>
<td>ΔR (%)</td>
<td>ΔPSNR (dB)</td>
<td>ΔR (%)</td>
</tr>
<tr>
<td>Carphone</td>
<td>Unidirectional</td>
<td>4.86</td>
<td>−0.318</td>
<td>8.31</td>
<td>−0.540</td>
</tr>
<tr>
<td></td>
<td>Hierarchical unidirectional</td>
<td>4.86</td>
<td>−0.318</td>
<td>6.88</td>
<td>−0.422</td>
</tr>
<tr>
<td>Foreman</td>
<td>Unidirectional</td>
<td>4.71</td>
<td>−0.289</td>
<td>8.00</td>
<td>−0.489</td>
</tr>
<tr>
<td></td>
<td>Hierarchical unidirectional</td>
<td>4.71</td>
<td>−0.289</td>
<td>7.33</td>
<td>−0.430</td>
</tr>
<tr>
<td>Ice</td>
<td>Unidirectional</td>
<td>4.58</td>
<td>−0.340</td>
<td>6.62</td>
<td>−0.480</td>
</tr>
<tr>
<td></td>
<td>Hierarchical unidirectional</td>
<td>4.58</td>
<td>−0.340</td>
<td>7.78</td>
<td>−0.552</td>
</tr>
<tr>
<td>Soccer</td>
<td>Unidirectional</td>
<td>5.46</td>
<td>−0.275</td>
<td>7.68</td>
<td>−0.387</td>
</tr>
<tr>
<td></td>
<td>Hierarchical unidirectional</td>
<td>5.46</td>
<td>−0.275</td>
<td>7.68</td>
<td>−0.387</td>
</tr>
</tbody>
</table>

$d = 2$ brings small average BD rate savings of 0.48% and 1.18% over the configuration using $d = 4$, in Foreman, GOP2 and 8, respectively. On the contrary, in Soccer, $d = 4$ slightly improves the compression performance bringing average BD rate gains of 0.15% and 0.43% for GOP2 and 8, respectively. This is because successive refinement enables the hash information to be progressively improved at the decoder. Therefore, starting from a less accurate description of the original WZ frame, that is, the initial hash generated by $d = 4$, does not significantly influence the overall compression performance. On the other hand, using $d = 4$ reduces the resolution of the hash, which diminishes the hash rate overhead.

With regard to complexity, the higher the hash downscaling factor the lower the computational cost associated to hash coding at the encoder. Hence, in order to constrain the encoding complexity to promote lightweight WMSNs applications, the rest of the experimental results are conducted by using $d = 4$.

7.4. Prediction Structures RD Performance Evaluation

This section assesses the compression performance of the proposed HSRWZ system (with $d = 4$) using the different motion prediction structures in Section 5. The proposed codec with hierarchical bidirectional (HIB.B) prediction is taken as reference. The experimental average BDs depicted in Table VI, show that, compared to the HIB.B structure, unidirectional prediction typically incurs a performance loss. As explained in Section 5.1, this is due to predicting from two temporal directions, which is especially beneficial when irregular motion content is encountered.

When utilizing the unidirectional prediction (IP..P) structure the highest performance decrease compared to the HIB.B structure is experienced for a GOP of size 4, namely, up to 8.31% in the Carphone sequence. Nevertheless, the performance drop of the IP..P structure with respect to the HIB.B structure diminishes for long GOPs (i.e., GOP8 and 16). This is because in the IP..P structure the temporal distance between the frame to be motion-compensated and the references remains constant irrespective of the GOP size. On the contrary, in the HIB.B structure the temporal distance of references in coarse temporal levels increases with the GOP length. Intriguingly, this enables the proposed codec with the IP..P structure to advance over the codec’s configuration using the HIB.B structure by a slight average BD rate gain of 1.18% in Soccer, GOP16.
The hierarchical unidirectional prediction (HIP.P) structure is equivalent to the IP.P structure when the GOP size is 2. However, conversely to the IP.P structure, when employing the HIP.P structure the RD performance loss compared to the HIB.B normally mounts with the GOP size. This behavior is reasonable given that, in the HIP.P prediction structure, the distance of reference frames at the coarser temporal levels increases with the GOP size. This is the expense paid for supporting temporally scalable, low-delay and low-memory coding.

We highlight that the compression loss caused by the use of unidirectional prediction (i.e., either IP.P or HIP.P) in the proposed HSRWZ codec is notably limited compared to the losses experienced by employing MCE in alternative DVC schemes, for instance, Girod et al. [2005], Li et al. [2007], and Natario et al. [2005]. This substantiates the potential of the proposed techniques in our DVC scheme.

7.5. Comparison against State-of-the-Art Codecs
To appraise the proposed codec, the compression performance is compared against the state-of-the-art Wyner-Ziv DISCOVER codec [Artigas et al. 2007]. Contrary to the proposed codec, which is hash-based, DISCOVER creates side-information using an advanced hierarchical MCI framework [Ascenso et al. 2006], which does not include successive refinement. In effect, DISCOVER is a well-established reference in the related literature, its executables are available online and it is well documented. Moreover, the H.264/AVC [Wiegand et al. 2003] Intra (Main Profile) codec has been included in the comparison as a low-complexity reference for H.264/AVC. Note that H.264/AVC in Main Profile\(^3\) is among the most efficient intra-frame codecs, regularly outperforming JPEG2000 [Taubman and Marcellin 2002; Schelkens et al. 2009].

The compression results of the proposed codec against DISCOVER and H.264/AVC Intra, for the considered test sequences and GOPs, are illustrated in Figure 6. Our system was configured using HIB.B, which is the prediction structure employed in DISCOVER. The results show that the proposed codec indisputably outperforms DISCOVER, at all GOPs and sequences. While DISCOVER’s performance drops dramatically when the GOP size increases, the proposed system suffers only a fairly small compression loss. As a consequence, the longer the GOP size, the higher the compression gain of the proposed codec, up to 21.53%, 40.98%, 42.03% and 39.95% BD rate savings in Carphone, Foreman, Ice, and Soccer GOP16, respectively. The severe performance loss of DISCOVER as the GOP size increases is due to the failure of the linear motion assumption at the basis of MCI [Ascenso et al. 2006]. On the contrary, the proposed hash-based and successively refinement technique accurately captures motion even in difficult conditions.

The proposed codec partially outperforms H.264/AVC Intra in Foreman and Carphone. In Ice and Soccer, both sequences with hard-to-code motion content, the proposed codec vastly diminishes the performance gap of Wyner-Ziv systems with respect to H.264/AVC Intra. However, as shown in Section 7.6, H.264/AVC Intra inflicts much higher computational complexity at the encoder, that is, the sensor node, which is a tailback for WMSNs applications.

Additionally, our HSRWZ codec is evaluated against the hash-based DVC systems of Ascenso et al. [2010], Deligiannis et al. [2009b], and Deligiannis et al. [2012a], as well as

---

\(^3\)The Main profile of the H.264/AVC Intra codec is selected to draw the results for two reasons. First, this enables a comparison of the proposed system with the state-of-the-art standard in intra-frame coding. Second, this offers a common platform for comparison with competing schemes. Namely, the configuration of the H.264/AVC Intra encoder is the same with the one employed by the DISCOVER codec, and the codecs of Ascenso et al. [2010], Ye et al. [2009], and Martins et al. [2009].
Fig. 6. RD performance of the proposed codec, configured using bidirectional hierarchical prediction, the Wyner-Ziv DISCOVER codec and H.264/AVC Intra for (a) Carphone, (b) Foreman, (c) Ice and (d) Soccer.
Table VII. Bjøntegaard Deltas of the Performance of the Proposed Codec (With HIB..B) Compared to State-of-the-art DVC Systems

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Codec</th>
<th>GOP 2</th>
<th>GOP 4</th>
<th>GOP 8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ΔR (%)</td>
<td>ΔPSNR (dB)</td>
<td>ΔR (%)</td>
<td>ΔPSNR (dB)</td>
</tr>
<tr>
<td>Foreman</td>
<td>DISCOVER</td>
<td>−4.77</td>
<td>0.316</td>
<td>−20.84</td>
</tr>
<tr>
<td></td>
<td>[Ascenso et al. 2010]</td>
<td>−3.67</td>
<td>0.242</td>
<td>−16.63</td>
</tr>
<tr>
<td></td>
<td>[Deligiannis et al. 2009b]</td>
<td>−19.48</td>
<td>1.443</td>
<td>−26.01</td>
</tr>
<tr>
<td></td>
<td>[Deligiannis et al. 2012a]</td>
<td>−1.24</td>
<td>0.076</td>
<td>−2.86</td>
</tr>
<tr>
<td></td>
<td>[Ye et al. 2009]</td>
<td>−2.77</td>
<td>0.189</td>
<td>−13.26</td>
</tr>
<tr>
<td></td>
<td>[Martins et al. 2009]</td>
<td>2.94</td>
<td>−0.156</td>
<td>−8.71</td>
</tr>
<tr>
<td>Soccer</td>
<td>DISCOVER</td>
<td>−16.85</td>
<td>0.961</td>
<td>−30.29</td>
</tr>
<tr>
<td></td>
<td>[Ascenso et al. 2010]</td>
<td>−7.93</td>
<td>0.441</td>
<td>−15.78</td>
</tr>
<tr>
<td></td>
<td>[Deligiannis et al. 2012a]</td>
<td>−5.01</td>
<td>0.269</td>
<td>−7.26</td>
</tr>
<tr>
<td></td>
<td>[Ye et al. 2009]</td>
<td>−10.72</td>
<td>0.608</td>
<td>−23.36</td>
</tr>
</tbody>
</table>

the successively refined DVC schemes of Ye et al. [2009] and Martins et al. [2009]. The Bjøntegaard rate and PSNR deltas [Bjøntegaard 2001] of the proposed codec against the competing codecs, for Foreman and Soccer and for GOP of sizes 2, 4, and 8, are listed in Table VII. The results show that with respect to the hash-based system of [Ascenso et al. 2010], the proposed codec brings BD rate savings of up to 27.92%, and 20.96% in Foreman and Soccer, GOP8, respectively. Furthermore, the proposed codec extensively improves over our previous hash-based codecs. Specifically, compared to our hash-based system in Deligiannis et al. [2009b], lacking Wyner-Ziv coding in transform domain but using only OBME and PMC to reconstruct frames, the proposed scheme yields BD rate savings of up to 43.74% and 51.55% for Foreman and Soccer, GOP8, respectively. With respect to our system in Deligiannis et al. [2012a], which featured an efficient hash, a transform domain Wyner-Ziv layer and accurate CCE but lacked side-information refinement, the experimental results report BD rate savings of up to, respectively, 4.58% and 11.41% for Foreman and Soccer in a GOP of 8. These results highlight the compression improvements brought by the novel techniques proposed in this article over our prior work.

Regarding the comparison with alternative DVC systems featuring side-information refinement, the proposed codec outperforms the system of Ye et al. [2009] by up to 25.85% and 30.66% in BD rate reduction in Foreman and Soccer, GOP8, respectively. Furthermore, compared to the state-of-the-art scheme of Martins et al. [2009], the proposed codec brings average BD rate improvements of up to 20.24% and 25.21% in Foreman and Soccer, GOP8, respectively. In particular, the codec of Martins et al. [2009] proposes side-information refinement upon decoding the coefficients of each DCT band in the WZ frame. Thus, these results confirm that the proposed successive refinement technique delivers superior RD performance, while still confining the imposed decoding complexity and structural latency.

Next, we assess the performance of the proposed codec, now configured using the HIP..P structure, against the alternative Wyner-Ziv systems. Note that all competing DVC schemes employ bidirectional prediction, thereby incurring encoding delay and additional memory demands at the encoder. The experimental results tabulated in Table VIII demonstrate that the proposed codec typically outperforms the competition, while still offering both low-delay and temporally scalable coding. These assets advocate the capacity of the proposed system in emerging WMSNs applications.
Table VIII. Bjøntegaard Deltas of the Performance of the Proposed Codec (With HIP-P) Compared to State-of-the-art DVC Systems

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Codec</th>
<th>ΔR (%)</th>
<th>ΔPSNR (dB)</th>
<th>ΔR (%)</th>
<th>ΔPSNR (dB)</th>
<th>ΔR (%)</th>
<th>ΔPSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>DISCOVER</td>
<td>−0.31</td>
<td>0.036</td>
<td>−14.83</td>
<td>0.933</td>
<td>−25.34</td>
<td>1.903</td>
</tr>
<tr>
<td></td>
<td>[Ascenso et al. 2010]</td>
<td>0.85</td>
<td>−0.037</td>
<td>−10.30</td>
<td>0.654</td>
<td>−21.55</td>
<td>1.380</td>
</tr>
<tr>
<td></td>
<td>[Deligiannis et al. 2012a]</td>
<td>3.41</td>
<td>−0.211</td>
<td>4.29</td>
<td>−0.251</td>
<td>3.51</td>
<td>−0.195</td>
</tr>
<tr>
<td></td>
<td>[Ye et al. 2009]</td>
<td>1.73</td>
<td>−0.098</td>
<td>−6.66</td>
<td>0.428</td>
<td>−19.35</td>
<td>1.258</td>
</tr>
<tr>
<td></td>
<td>[Martins et al. 2009]</td>
<td>7.84</td>
<td>−0.434</td>
<td>−1.91</td>
<td>0.162</td>
<td>−13.42</td>
<td>0.889</td>
</tr>
<tr>
<td>Soccer</td>
<td>DISCOVER</td>
<td>−12.25</td>
<td>0.685</td>
<td>−24.83</td>
<td>1.534</td>
<td>−33.70</td>
<td>2.225</td>
</tr>
<tr>
<td></td>
<td>[Ascenso et al. 2010]</td>
<td>−2.81</td>
<td>0.168</td>
<td>−9.11</td>
<td>0.532</td>
<td>−14.16</td>
<td>0.832</td>
</tr>
<tr>
<td></td>
<td>[Deligiannis et al. 2009b]</td>
<td>−29.09</td>
<td>1.672</td>
<td>−34.69</td>
<td>2.056</td>
<td>−46.75</td>
<td>2.883</td>
</tr>
<tr>
<td></td>
<td>[Deligiannis et al. 2012a]</td>
<td>0.19</td>
<td>−0.005</td>
<td>−0.06</td>
<td>0.035</td>
<td>−3.91</td>
<td>0.212</td>
</tr>
<tr>
<td></td>
<td>[Ye et al. 2009]</td>
<td>−5.81</td>
<td>0.322</td>
<td>−17.42</td>
<td>0.628</td>
<td>−24.85</td>
<td>1.572</td>
</tr>
<tr>
<td></td>
<td>[Martins et al. 2009]</td>
<td>−1.25</td>
<td>0.101</td>
<td>−14.02</td>
<td>0.852</td>
<td>−19.03</td>
<td>1.218</td>
</tr>
</tbody>
</table>

Table IX. Encoding Execution Time (in seconds) of H.264/AVC Intra (CIntra) and the Proposed HSRWZ Codec (CHSRWZ) with \(d = 4\). The Ratio Between the Encoding Execution Time of the Proposed Codec and H.264/AVC Intra is Highlighted

<table>
<thead>
<tr>
<th>QP Intra</th>
<th>CIntra (sec)</th>
<th>QP Hash</th>
<th>QM</th>
<th>CKey (sec)</th>
<th>CWZ (sec)</th>
<th>CHSRWZ (sec)</th>
<th>CHash (sec)</th>
<th>CTime (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>40</td>
<td>14.22</td>
<td>1</td>
<td>7.02</td>
<td>0.91</td>
<td>0.55</td>
<td>8.48</td>
<td>(\frac{1}{1.79})</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>18.54</td>
<td>40</td>
<td>7.72</td>
<td>1.1</td>
<td>0.54</td>
<td>9.36</td>
<td>(\frac{1}{2.01})</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>18.97</td>
<td>39</td>
<td>8.86</td>
<td>1.33</td>
<td>0.55</td>
<td>10.74</td>
<td>(\frac{1}{2.32})</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>21.92</td>
<td>38</td>
<td>10.3</td>
<td>1.42</td>
<td>0.61</td>
<td>12.33</td>
<td>(\frac{1}{2.67})</td>
</tr>
<tr>
<td>Soccer</td>
<td>44</td>
<td>13.05</td>
<td>41</td>
<td>6.82</td>
<td>0.97</td>
<td>0.48</td>
<td>8.27</td>
<td>(\frac{1}{1.64})</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>14.49</td>
<td>42</td>
<td>7.32</td>
<td>1.11</td>
<td>0.5</td>
<td>8.93</td>
<td>(\frac{1}{1.92})</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>16.12</td>
<td>40</td>
<td>7.87</td>
<td>1.39</td>
<td>0.5</td>
<td>9.76</td>
<td>(\frac{1}{2.05})</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>20.72</td>
<td>38</td>
<td>9.83</td>
<td>1.64</td>
<td>0.52</td>
<td>11.99</td>
<td>(\frac{1}{2.53})</td>
</tr>
</tbody>
</table>

7.6. Encoding Complexity Analysis

Motivated by the low power processing capacity of sensor nodes, low-cost encoding is a key issue when designing a video compression system for WMSNs. To assess the encoding complexity of the proposed HSRWZ system, we conduct execution time measurements under regulated conditions\(^4\), similar to Artigas et al. [2007] and Pereira et al. [2007]. Table IX reports the encoding complexity in terms of execution time (in seconds) of H.264/AVC Intra and the proposed codec (with \(d = 4\)) per RD point determined by the QPIntra and QPHash, QM parameters, respectively. Experiments are reported for all frames of the Foreman and Soccer sequences at a GOP2 and 8.

The encoding execution time of the proposed codec, denoted by \(C_{HSRWZ}\), is defined by the sum of the H.264/AVC Intra encoding time of the key frames, that is, \(C_{Key}\), and the Wyner-Ziv encoding time, \(C_{WZ}\), together with the hash formation and compression

\(^4\)Encoding execution time tests using the executables of the JM implementation of H.264/AVC and the proposed system with \((d = 4)\) were conducted under the same hardware and software conditions. The employed hardware was a personal computer with Intel\textsuperscript{®} Core\textsuperscript{TM} i7 CPU at 2.2GHz, and with 16GB of RAM. As regards the software conditions, a Windows 7 operating system was used, while the executables where obtained using the Visual Studio C++ v8.0 compiler in release mode. We note that both the JM implementation of H.264/AVC and the implementation of our codec have not been optimized for speed.
time, $C_{Hash}$, for the WZ frames. The results in Table IX show that the encoding complexity of the proposed codec is primarily determined by the H.264/AVC Intra frame coding of the key frames. This is because Wyner-Ziv encoding poses very low computational demands, simply consisting of the integer approximation of the DCT, quantization, bit-planes extraction and LDPCA encoding. The latter is performed by the multiplication of a binary array with a sparse parity-check matrix. Conversely, apart from the integer approximation of the DCT and quantization, components that are similar with the ones employed by Wyner-Ziv encoding, the H.264/AVC Intra encoder deploys several $4 \times 4$ and $16 \times 16$ intra prediction modes, rate-distortion optimal mode selection and entropy coding. These features impose considerably higher computational demands at the encoder compared to Wyner-Ziv encoding [Pereira et al. 2007].

At large, the encoding time measurements in Table IX show that the proposed HSRWZ codec yields significantly lower encoding complexity compared to H.264/AVC Intra, irrespective of the RD point. This complexity reduction grows with the length of the employed GOP, as the number of coded key frames diminishes. In particular, the encoding time of the proposed codec is equal to 56% and 29% of the H.264/AVC Intra encoding time for Foreman, RD point 4, GOP 2 and 8, respectively.

Compared to pure TDWZ codecs, for instance, DISCOVER [Artigas et al. 2007], the proposed encoder exhibits higher complexity due to hash encoding. This hash complexity overhead is given by $C_{Hash}/(C_{Key} + C_{WZ})$ and can be derived from Table IX. This complexity overhead is small (about 5%) for short GOPs but increases with the GOP size (e.g., up to approximately 20% for GOP8), as more hash frames are encoded. However, this overhead is well-justified given the gain in RD performance (up to 42.03% over DISCOVER). This rate reduction causes an important diminution of the processing power consumed by the transmission part of a visual sensor node.

Concentrating on the relation between the rate and the encoding complexity, one notices that the encoding complexity of both the proposed system and H.264/AVC Intra increases with the rate. However, at the same time the relative encoding complexity between the two systems, that is, $C_{HSRWZ}/C_{Intra}$, decreases. In other words, the higher the rate the more the complexity reduction offered by the proposed codec. To explain this effect, we note that the H.264/AVC Intra codec's complexity increase is mainly accredited to the higher amount of entropy coded information (i.e., lower QP). For the proposed codec, a similarly increase with the rate is observed for the key frames' encoding complexity. The complexity increment of the hash, though, is moderate as the employed QPs are decreasing much less than the QPs used for the key frames. In addition, the Wyner-Ziv encoding complexity increase, due to a higher LDPCA rate, is quantitatively a lot less than the complexity of context-adaptive arithmetic entropy encoding in H.264/AVC Intra [Pereira et al. 2007]. It is also worth mentioning that the hash complexity overhead, $C_{Hash}/(C_{Key} + C_{WZ})$, with respect to TDWZ systems, diminishes with the rate due to the moderate hash complexity increment.

### 7.7. Encoder Power Consumption Analysis

We now endeavor an estimation of the power consumption of an implementation of the proposed encoder for visual sensors. Our analysis builds upon prior studies [Ahmad et al. 2009; Ukhanova et al. 2010] on the power consumption of DVC systems.

In the work of Ahmad et al. [2009] software implementations of the H.264/AVC Intra and the standard TDWZ encoder were compiled via arm-linux-gcc/g++ 3.3.2 and run on the Stargate platform; while the resulting bit-streams were transmitted via the TelosB mote. The energy depletion per module was obtained by computing the CPU cycles and then multiplying with the energy consumption per cycle count. The experiments from Ahmad et al. [2009] showed that H.264/AVC Intra consumes 57.74mJ/frame on average to encode Foreman QCIF at 15Hz. For the TDWZ encoder...
the corresponding consumption was 34.26, 22.49 and 17.48 mJ/frame for GOP2, 4 and 8, respectively. Given the proportion of key and WZ frames per GOP, one can compute that the Wyner-Ziv encoding part consumed roughly 10.8 mJ per WZ frame.

Compared to the TDWZ encoder assessed in Ahmad et al. [2009], the proposed HSRWZ allocates further resources to encode a hash per WZ frame. Assuming that the H.264/AVC Intra energy consumption scales roughly linearly with the frame resolution\(^5\), the extra consumption per WZ frame due to hash encoding can be approximated by dividing the average key-frame consumption by 1/d\(^2\). Under this assumption and using the energy consumption measurements for the key and WZ frames from Ahmad et al. [2009], one can infer that the consumption of the proposed encoder would be approximately 36.07, 25.2 and 20.64 mJ/frame for Foreman QCIF, 15Hz using GOP2, 4 and 8, respectively. This corresponds to a reduction by 37.5, 56.4, and 64.3% over the energy consumed by the H.264/AVC Intra encoder.

Instead of using measurements from software implementations on a reprogrammable platform, Ukhanova et al. [2010] considered a model based analysis to compare the encoder power consumptions of TDWZ and H.264/AVC No Motion. Using power consumption measurements obtained from implementations of the CAVLC [Huang and Lei 2008] and LDPC [Swamy et al. 2005] encoders on an ASIC 0.18 m chip, [Ukhanova et al. 2010] showed that a TDWZ encoder depletes approximately 15–60% less energy than the H.264/AVC No Motion baseline encoder.

Similar to the analysis of Ukhanova et al. [2010], we develop a model of the power consumption of the proposed HSRWZ encoder versus that of H.264/AVC Intra. The power consumption (expressed in mW) of an implementation of the H.264/AVC Intra encoder, configured in the Main Profile, can be formulated as

\[ P_{AVC\text{Intra}} = P_{PTQ}(H, V) + P_{EC}(R_{AVC\text{Intra}}) + P_{0}^{AVC\text{Intra}}, \]  

(14)

In (14), \( P_{PTQ}(H, V) \) is the power consumption associated to the intra prediction, mode decision, reconstruction, DCT, quantization and zig-zag scanning components of the H.264/AVC Intra encoder; in our notation PTQ stands for prediction, transform and quantization. \( P_{PTQ} \) depends on the frame resolution expressed by the frame dimensions \((H, V)\). Additionally, \( P_{EC}(R_{AVC\text{Intra}}) \) represents the power consumption of the entropy encoder, that is, the context adaptive binary arithmetic coding (CABAC) engine, which depends on the rate \( R_{AVC\text{Intra}} \). Finally, \( P_{0}^{AVC\text{Intra}} \) is a constant component related to the power consumption of the control unit and the interface for data transfer from/to the memory.

In our HSRWZ codec, the encoded video sequence is split into key and WZ frames with a proportion of \( F_{\text{Key}} \) and \( F_{\text{WZ}} \), respectively, for instance, \( F_{\text{Key}} = 1/4 \), \( F_{\text{WZ}} = 3/4 \) for GOP4. The power consumption of the HSRWZ encoder can then be expressed as

\[ P_{HSRWZ} = F_{\text{Key}} \times [P_{PTQ}(H, V) + P_{EC}(R_{AVC\text{Intra}})] + F_{\text{WZ}} \times [P_{WZ} + P_{\text{Hash}}] + P_{0}^{HSRWZ}, \]  

(15)

where the first term is the power consumption of the H.264/AVC Intra coding of the key frames. The second term expresses the consumption for encoding the WZ frames, which consists of a standard Wyner-Ziv coding part \( P_{WZ} \) and a hash coding part \( P_{\text{Hash}} \). The last term \( P_{0}^{HSRWZ} \) is a constant component similar to \( P_{0}^{AVC\text{Intra}} \) in (14). Since the hash is a low resolution H.264/AVC Intra coded version of the WZ frame, \( P_{\text{Hash}} \) can be given by

\[ P_{\text{Hash}} = P_{PTQ}(H/d, V/d) + P_{EC}(R_{\text{Hash}}), \]  

(16)

\(^5\)The encoding complexity results depicted in Table IX corroborate the assumption that \( C_{\text{Hash},\text{Key}} \approx 1/d^2 \). For instance, in a GOP of size 2, in which the number of WZ and key frames in the sequence are almost equal, the execution time ratio \( C_{\text{Hash},\text{Key}} \) in Table IX is approximately equal to \( 1/d^2 = 1/16 \).
where the power consumption of the down-sampling operation is considered negligible. In turn, the power consumption of the Wyner-Ziv encoder can be written as

\[
P_{WZ} = P_{TQ}(H, V) + P_{LDPC}(R_{LDPC}).
\] (17)

In (17), the first term corresponds to the consumption of DCT and quantization, components that are equivalent to those in H.264/AVC Intra. The second term represents the consumption of the LDPC encoder, which depends on the Slepian-Wolf rate.

A detailed report of the power consumption of the different components of the H.264/AVC Intra codec, implemented on an ASIC 0.12 m chip at 114MHz, can be found in Kuo et al. [2011]. Based on these measurements, the power consumption \( P_{Hash} \) can be estimated as well. Specifically, since the number of PTQ operations in the H.264/AVC codec scales linearly with the resolution, the \( P_{PTQ} \) for the hash is \( 1/16 \) of the \( P_{PTQ} \) for an H.264/AVC Intra coded frame at full resolution. Moreover, as in Ukhanova et al. [2010], we consider a linear relation between the power consumption of the entropy coding part and the rate, that is, \( P_{EC} \propto R \). Experimental evidence has shown that the hash rate per frame is on average about 10% of the H.264/AVC Intra rate (per frame), meaning that \( P_{EC}(R_{Hash}) \) is roughly 10% of \( P_{EC}(R_{AVC, Intra}) \).

Regarding the power consumption of the Wyner-Ziv encoder, the term \( P_{TQ} \) can be assembled with the numbers for the DCT and quantization consumption given in Kuo et al. [2011]. For the power consumption \( P_{LDPC} \), one may use the measurements given in Swamy et al. [2005], where an LDPC encoder was implemented on a 0.18 m\(^6\) chip at 125MHz consuming 3.4mW at 100Mbps, a throughput sufficient to code the data rates considered in Kuo et al. [2011]. Assuming \( P_{H_{SRWZ}} \propto P_{AVC, Intra} \), and using the measurements in Kuo et al. [2011] and Swamy et al. [2005], the relative reduction in power consumption of the proposed HSRWZ encoder versus H.264/AVC Intra can be approximated by 29.3, 43.9 and 51.2% for GOP2, 4 and 8, respectively. These gains are still modest due to the overestimation of the LDPC consumption.

### 7.8. Decoding Complexity Analysis

We now focus on the decoding complexity of the proposed system compared to that of the reference DISCOVER [Artigas et al. 2007] codec. In order to quantify complexity, we perform execution time measurements under the conditions described in Section 7.6. Such measurements are commonly used to assess the decoding complexity of DVC algorithms [Martins et al. 2009; Wang et al. 2012; Deligiannis et al. 2012a]. Table X reports the decoding execution time (in seconds) per RD point of the proposed DVC system with (\(T_{H_{SRWZ}}\)) and without (\(T_{noRef.}\)) side-information refinement. The corresponding decoding execution times of DISCOVER (\(T_{DISC}\)), acquired with the codec’s executables, are also provided. Results are reported for all frames of the Foreman and Soccer sequences at GOP sizes of 2 and 8.

The results in Table X show that, for a GOP of size 2, the decoding complexity of the proposed hash-based system without side-information refinement is comparable to that of the DISCOVER codec. In particular, although in Foreman GOP2 the decoder of DISCOVER is less complex, in Soccer GOP2, the proposed system’s decoder runs faster. For a GOP of size 8, the proposed system without refinement has systematically lower decoding complexity than DISCOVER. To explain this effect, we first remark that the proposed non-refined system employs hash-driven side-information generation using OBMEC and bit-plane-per-bit-plane successively refined correlation estimation. Although these components are typically more computationally expensive.

---

\(^6\)Power consumption measurements for an ASIC 0.12 m implementation of the LDPC encoder could not be found. Instead, the measurements from Swamy et al. [2005] provide an upper bound on the LDPC consumption, as for the same implementation, a 0.18 m ASIC consumes more than a 0.12 m one.
Table X. Decoding Execution Time (in seconds) of the Proposed Codec With \(T_{\text{HSRWZ}}\) and Without \(T_{\text{noRef.}}\) Side-Information Refinement Compared to DISCOVER’s \(T_{\text{DISC}}\)

<table>
<thead>
<tr>
<th>GP2</th>
<th>GOP8</th>
<th>RD</th>
<th>Foreman</th>
<th>Soccer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>T_{DISC}</td>
<td>T_{noRef}</td>
<td>T_{HSRWZ}</td>
</tr>
<tr>
<td>1</td>
<td>360.28</td>
<td>341.01</td>
<td>0.95</td>
<td>402.75</td>
</tr>
<tr>
<td>2</td>
<td>712.51</td>
<td>713.08</td>
<td>1.00</td>
<td>815.31</td>
</tr>
<tr>
<td>3</td>
<td>1243.95</td>
<td>1395.88</td>
<td>1.12</td>
<td>1367.54</td>
</tr>
<tr>
<td>4</td>
<td>1888.56</td>
<td>2340.62</td>
<td>1.24</td>
<td>2080.05</td>
</tr>
</tbody>
</table>

than the corresponding techniques in DISCOVER, they yield higher compression efficiency especially in difficult conditions, that is, in fast motion content and long GOPs—see Figure 5(b)–(d). This means that fewer LDPCA decoding iterations are required to obtain a given quality, which in turn decreases the overall decoding complexity. In Foreman GOP2, though, the proposed non-refined system is less efficient RD-wise than DISCOVER—see Figure 5(a)—which, as mentioned in Section 7.3, is due to the high hash down-sampling factor \(d = 4\). This explains why the proposed system’s decoder is more complex in this case. It is also noteworthy that, for a given sequence and GOP, the lowest relative decoding complexity values \(T_{\text{noRef.}}/T_{\text{DISC}}\) are obtained for RD point 1 and 2. This is because the proposed non-refined system brings higher RD gains over DISCOVER at low and medium rates—see Figure 5(b)–(d).

When the proposed side-information refinement is switched on, the HSRWZ system’s decoder is usually more time-consuming than the baseline version of the codec for a GOP of size 2. This complexity increment is attributed to the repeated side-information generation and correlation estimation operations in the HSRWZ codec. On the other hand, when the GOP size increases, for instance, GOP8, the proposed successively refined decoder becomes less complex than that of the nonrefined system. This is because in difficult conditions (e.g., large GOPs), the quality increase in the side-information leads to much less soft-decoding runs that compensate for the computational cost associated to side-information refinement. Additionally, observe that the relative decoding complexity, \(T_{\text{HSRWZ}}/T_{\text{noRef.}}\), between the refined and the nonrefined version of the proposed system reduces with the rate. This is expected as the impact of a high-quality side-information, obtained by the proposed refinement technique, grows with the amount of information that is LDPCA coded.

Compared to DISCOVER, the proposed HSRWZ system’s decoder is shown to be more complex in Foreman GOP2. This complexity rise is due to the successively refined OBMEC and correlation estimation processes in the proposed HSRWZ decoder. Then again, under complex motion content and long GOPs, for instance, in Soccer GOP8, 8 and in Foreman GOP8, the proposed HSRWZ decoder is drastically less complex (i.e., over 20%) than DISCOVER. Essentially, under such difficult conditions, HSRWZ brings significant RD gains over DISCOVER, which result in much lower soft-decoding iterations and thus in a reduced overall decoding complexity.

8. CONCLUSIONS

Targeting efficient video compression for visual sensor technology, this article has presented a novel Wyner-Ziv video coding architecture, offering low complexity encoding at the recording sensor node. The proposed scheme realizes successively refined side-information generation by decoding the WZ frames in distinct stages. At every stage,
an advanced motion-compensated prediction framework, tailored to the proposed refinement strategy, enables increasingly accurate side-information creation at the decoder as more stages are completed. Additionally, the proposed architecture featured several temporal prediction structures, intended for low delay and scalability. The experimental assessment of the proposed codec has demonstrated the superior compression performance with respect to the benchmark DISCOVER codec with BD rate savings of up to 42.03%. Moreover, the experimental results report significant performance gains of up to 30.66% and 25.21% in BD rate reduction compared to the recent successively refined TDWZ systems of Ye et al. [2009] and Martins et al. [2009], respectively. Finally, the proposed codec advances over our previous systems in Deligiannis et al. [2009b] and Deligiannis et al. [2012a], with BD rate savings of up to 51.55% and 11.41%, respectively.

REFERENCES


Received March 2012; revised September 2012; accepted January 2013