Real-Time Distributed Video Coding for 1K-pixel Visual Sensor Networks

Jan Hanca\(^a\), Nikos Deligiannis\(^a\), Adrian Munteanu\(^a\)
\(^a\)Vrije Universiteit Brussel (VUB), Department of Electronics and Informatics/iMinds, Pleinlaan 2, 1050 Brussel, Belgium

Abstract. Many applications in visual sensor networks demand low-cost wireless transmission of video data. In this context, distributed video coding (DVC) has proven its potential to achieve state-of-the-art compression performance while maintaining low computational complexity of the encoder. Despite of their proven capabilities, current DVC solutions overlook hardware constraints and this renders them unsuitable for practical implementations. This paper introduces a DVC architecture that offers highly-efficient wireless communication in real-world visual sensor networks (VSNs). The design takes into account the severe computational and memory constraints imposed by practical implementations on low-resolution visual sensors. We study performance-complexity trade-offs for feedback-channel removal, propose learning-based techniques for rate allocation and investigate various simplifications of side information generation yielding real-time decoding. The proposed system is evaluated against H.264/AVC intra, Motion-JPEG and our previously designed DVC prototype for low-resolution visual sensors. Extensive experimental results on various data show significant improvements in multiple configurations. The proposed encoder achieves real-time performance on a 1k-pixel visual sensor mote. Real-time decoding is performed on Raspberry Pi single-board computer or a low-end notebook PC. To the best of our knowledge, the proposed codec is the first practical DVC deployment on low-resolution VSNs.

Keywords: visual sensor networks, distributed video coding, Wyner-Ziv video coding, low-resolution video, real-time video coding.

1 Introduction

Process management, healthcare monitoring or environmental sensing can easily be automated nowadays by using wireless Visual Sensor Networks (VSNs). Moreover, the availability of low-cost CMOS cameras has created the opportunity to increase the spatio-temporal resolution provided by visual sensors. This yields an accurate description of the physical phenomena around the sensor, which is useful in various applications such as area surveillance, tracking, environmental monitoring and medical examinations.\(^1,2\) Unfortunately, high-resolution images and videos require high bandwidth and powerful hardware for data processing. Thus, they are not always practical in real-world applications due to high computational demands and cost of the system (cameras,
networking and the computing infrastructure). In particular, the power consumption of high resolution visual sensing and processing devices is a serious bottleneck in case of battery-powered VSNs.

These limitations can be alleviated by employing low-resolution imaging devices (motes). Although the amount of collected information is much smaller in this case, VSNs composed of such motes proved to perform surprisingly well in several domains, such as security and surveillance. For instance, it has been shown that accurate occupancy maps can be generated using networks of sensors acquiring image data of $64 \times 48$ pixels\textsuperscript{3} or even $30 \times 30$ pixels\textsuperscript{4,5} Inexpensive low-resolution visual sensors employed in the latter are practically used for long-time behaviour analysis in elderly care.\textsuperscript{6} Furthermore, such sensors can also be used for posture recognition,\textsuperscript{3,7} depth estimation in case of a stereoscopic setup,\textsuperscript{8} or even fire detection.\textsuperscript{9}

Several applications in low resolution VSNs require the transmission of the video signal to the sink, which exploits the information from the multiple nodes at once. Solving ambiguities in computer vision problems (e.g. fall detection) benefits from centralized processing of visual information due to the very low spatial resolutions of the data acquired with such sensors. In case of emergency, users may request video transmission from the monitored environment in order to visually verify triggered alarms and discriminate between false and true positives before subsequent measures are taken.

Despite the low spatial resolutions, transmitting video from such motes is very challenging, as the data rates are extremely low (of the order of tens of kilobits per second). The limited computational power and energy supply of such wireless visual sensors poses severe constraints on the allowable complexity of the employed video coder. Additionally, high compression performance is required in order to minimize the amount of information transmitted by the sensor node. Sig-
nificant rate reductions decrease the power consumption of the transmitter, thereby prolonging the battery life. Hence, developing efficient video coding and transmission methods plays an important role in low-resolution VSNs.

Traditional video codecs, such as H.264/AVC, focus on achieving high compression performance in a downlink-oriented model, whereby the correlation is exploited at the encoder through the use of complex spatial and temporal prediction tools. Moreover, the modern video coding standards were designed to perform well on high-resolution input data. In contrast, Distributed Video Coding (DVC) follows an uplink-oriented model, whereby efficient compression is obtained by exploiting the source statistics at the decoder. This approach is based on the fundamental work of Slepian and Wolf, who proved that separate lossless encoding but joint decoding of two independent random sources can be as efficient as a joint encoding and joint decoding scheme. These results have been extended to lossy coding by Wyner and Ziv, who established the rate-distortion function when a single source is encoded independently but decoded in the presence of a correlated signal, called side information (SI). Recently, the exact Wyner-Ziv limit for binary source coding with SI correlated by a Z-channel was presented by Deligiannis et al. In case of mono-view video data, the Wyner-Ziv coding principle can be applied by splitting the input sequence into two sources of information. The first source is coded using a conventional image coding technique, such as JPEG or H.264/AVC intra, and is used to generate SI for the second source at the decoder. Hence, the computationally expensive inter-source redundancy removal can be skipped on low-power sensing devices. The encoding of the second source is reduced to quantization followed by Slepian-Wolf (SW) encoding, which is implemented in most cases using a low-complexity channel encoder.

However, current DVC systems still fail to match the coding performance of traditional com-
pression algorithms. Consequently, the most popular visual sensor network platforms employ predictive coding architectures,\textsuperscript{17} namely JPEG\textsuperscript{15} or H.264/AVC.\textsuperscript{10} Despite of the research efforts invested in DVC in the last years, there are still multiple open problems in such uplink-oriented compression paradigms. First of all, the efficiency of Wyner-Ziv (WZ) video codecs is limited by the temporal prediction tools (generating the SI), which assume a linear motion model. This assumption fails in case of irregular motion or large temporal distance between intra-coded frames.\textsuperscript{18} Secondly, a consequence of the asymmetric nature of a WZ coding paradigm is that the channel rate required for successful decoding is not known by the encoder. Conventionally, this is solved by making use of a feedback channel, which allows the decoder to progressively request information from the encoder until successful decoding is achieved. Although such architectures yield state-of-the-art performance in DVC, they suffer from delays and they require large memory footprint for buffering purposes.\textsuperscript{16} Finally, high decoding complexity may be a drawback if high node density sensor networks are deployed.\textsuperscript{17}

This work investigates the design and implementation of a transform-domain DVC architecture for a very low-resolution video sensor. Our contributions are as follows:

- We propose a novel DVC system, tailored to the characteristics of the considered sensor node,\textsuperscript{7} which captures the scene at $30 \times 30$ video resolution and poses severe design constraints in terms of memory and computational power. The low complexity and low memory requirements of the encoder are demonstrated by a practical implementation offering real-time execution on the mote.\textsuperscript{19}

- We carry out a thorough performance and complexity analysis of multiple state-of-the-art WZ decoding tools, including SI generation and refinement methods when applied to
extremely low-resolution input data. Moreover, we compare feedback-channel-free and feedback-channel-enabled codec architectures to estimate the performance loss of the more practical former configuration. Following this study, we propose multiple decoder configurations operating in real-time on a low-end notebook PC. Moreover, we evaluate the decoding speed on Raspberry Pi single-board computer. To better highlight the results, we use a single-thread CPU implementation.

- We evaluate the compression performance of the proposed system against a previously proposed DVC system designed for the same sensor, as well as Motion-JPEG (MJPEG), H.264/AVC intra and the predictive codec presented in our previous work. The experimental results prove that our proposed DVC architecture is an efficient low-complexity alternative to conventional coding systems.

We note that a first design of a DVC system for extremely low video resolution was described in our prior publication. Although the work presented in this paper shares a few conceptual ideas with its predecessor, there are many significant differences. First, our previous DVC design did not account for the actual memory and computational constraints imposed by practical deployments on such sensors. Additionally, a block-based codec design is followed in the proposed DVC, lying in contrast will all our previously proposed DVC systems. Moreover, a mode-decision algorithm was added, which was not present before. Furthermore, in this work we perform an exhaustive study of the influence of multiple decoding tools on performance and complexity, which was not in the scope of our previous work. Last, but not least, the experimental results with our previous DVC system were reported based on the offline simulations only, without any practical realization. In contrast, the proposed DVC architecture is implemented in the firmware of the
sensor node, while the decoder is executed in real-time on a low-end laptop PC and Raspberry Pi. To the best of our knowledge, the proposed DVC system is the first practical realization of a distributed video codec on low-resolution visual sensors.

The remaining parts of this paper are organized as follows. Section 2 gives an overview of related work and highlights the novel features of the proposed codec. The architecture of the proposed DVC is presented in Section 3. Section 4 reports the complexity of the encoder, decoding times and the rate-distortion performance obtained with the proposed system. Finally, Section 5 draws the conclusions of our work.

2 Related work

In terms of design, two major DVC architectures have been proposed in the literature so far. The first architecture, called PRISM, performs a rough, block-wise classification at the encoder, using the squared-error difference between the block and its co-located block in the previous frame. If the similarity is very high, the block is not transmitted, as it can easily be reconstructed at the decoder. Analogously, some of the bitplanes are omitted if the blocks are much alike. Otherwise, the block is SW encoded using Bose-Chaudhuri-Hocquenghem (BCH) codes or entropy coded if the blocks differ heavily.

In an alternative architecture proposed at Stanford, a conventional intra-frame codec is used to encode the key frames, based on which the decoder generates the SI. High performance channel codes are used to exploit the correlation between WZ data and the SI. Typically, accumulated low-density parity check codes (LDPCA) with a rate-compatible scheme are employed. The rate control is performed at the decoder and communicated to the encoder via a feedback channel. This architecture was adopted and optimized in the DISCOVER codec, which is the well-known DVC
benchmark.

2.1 SI generation and refinement techniques

First DVC schemes employed motion-compensated interpolation (MCI) to produce SI at the de-
coder. The precision of MCI was improved by performing bi-directional motion estimation
and compensation followed by a sub-pixel motion vector refinement and vector smoothing.
Bidirectional overlapped block motion compensation (OBMC) using more than a single symmet-
ric motion-vector per block further increased performance over MCI. A refinement step updat-
ing the interpolated frame by a weighted robust principal component analysis was recently intro-
duced. Currently, state-of-the-art DVC systems create the SI by making use of optical flow al-
gorithms, which produce high-quality temporal interpolations at the cost of substantially increased
computational complexity.

The performance of any SI generation algorithm significantly drops in videos including irreg-
ular motion. Moreover, if the temporal distance between key frames increases, the assumption
of a linear motion model is more likely to fail, causing a significant decrease in the overall cod-
ing efficiency. To overcome this problem, alternative DVC schemes employ hash-based motion
estimation, in which additional description of the WZ-coded frame is sent to the decoder to
support SI generation. Similar aid can be extracted from the successfully decoded WZ data. In
particular, many researchers designed methods to successively refine the SI. Recalculation of the
SI after decoding all DCT bands was suggested by Ye et al., while regenerating SI after recon-
struction of each DCT frequency band was evaluated by Martins et al. It was showed that motion
estimation and SI refinement can be intertwined with low-density-parity-check (LDPC) decod-
ing. Finally, it was also found that progressively refining both the correlation channel estimation
and SI generation leads to substantial improvements in DVC - see e.g. Deligiannis et al.\textsuperscript{18}

2.2 Feedback-channel-free architectures

In feedback-channel-free DVC architectures, it is the encoder which is responsible for determining the rate required for successful decoding. Thus, the encoder has to guess the quality of the SI, which is available only at the decoder. This problem is typically solved by generating a coarse approximation of the SI at the encoder. An example of such a computationally inexpensive method is averaging the intra-coded frames.\textsuperscript{42,43} Artigas and Torres\textsuperscript{42} derive the SW rate from the difference between the estimated SI and the input frame, based on empirical results obtained in offline experiments. Morbee et al.\textsuperscript{43} model the correlation noise distribution to be zero-mean Laplacian and use this model to derive the bitplane error probability, which was next mapped to a bitrate using offline training. Similarly, Brites and Pereira\textsuperscript{44} model the correlation noise per frequency band after performing simplified motion estimation. Low-complexity emulation of hash-based SI generation was showed in our prior work.\textsuperscript{16}

Apart of incorporating tools enabling unidirectional DVC, feedback-channel-free systems should also feature tools minimizing the effects of decoding failures. SI refinement is one of the means of reducing the total number of such failures. As suggested in our previous work,\textsuperscript{16} SW decoding should be attempted again for all SW coded bitplanes that failed to decode at previous SI refinement levels, when only a poorer version of the SI was available. Brites and Pereira\textsuperscript{45} proposed to flip the log-likelihood-ratios (LLRs) for the bits that are most likely to be erroneous. Soft channel decoding is attempted again afterwards. As suggested by Berrou et al.,\textsuperscript{46} soft reconstruction of unknown pixel values should exploit the probability of every bit in every bitplane; the weighted sum of the centroids of every individual bin, in which the weights are assigned according
to bitplane LLRs, achieves the best performance.\textsuperscript{46}

It is important to point out that feedback-channel-free architectures suffer a performance loss with respect to their feedback-based counterparts. The encoder does not have access to the SI nor feedback from the decoder and, therefore, it is not able to accurately determine the necessary rate that guarantees successful decoding. Hence, such systems benefit of lower latencies due to lack of feedback, but tend to overestimate or underestimate the rate needed to secure successful decoding.

2.3 \textit{Real-Time DVC for Visual Sensor Networks}

Although DVC has demonstrated high potential for many practical implementations, it has never became popular in \textit{real-world} applications. This is mainly due to (i) the feedback channel which incurs delays and (ii) the inherently complex decoder, which hinders the practical values of DVC for real-time utilization. The DVC decoder is inherently complex, as it has to create the SI, estimate the correlation model and attempt to decode each codeword multiple times. Moreover, SI and correlation channel refinement, which are required in order to provide competitive coding performance, add further costly computations to the sink node.

Recent hardware advances allowed for speeding up many decoding algorithms in DVC. In case of simple feedback-based architectures, if one ignores the refinement step, LDPCA decoding is reported to contribute to over 90\% of the decoder’s complexity.\textsuperscript{27} Sum-product decoding can be fully parallelized and implemented on a GPU,\textsuperscript{47} which resulted in real-time decoding of QCIF video ($176 \times 144$ pixels). The decoder’s complexity is significantly reduced in case of feedback-channel-free architectures, in which LDPCA decoding is executed only once per codeword.\textsuperscript{48} Such systems can achieve over 30 frames per second for CIF input ($352 \times 240$ pixels) if paired with lightweight SI generation.\textsuperscript{48} Finally, applying mode decision and a skip mode at the encoder
reduces the number of codewords which further speeds up the decoding process.  

3 Video Codec Architecture

The proposed system belongs to the popular Stanford class of distributed video codecs. The block diagram of the proposed codec is presented in Figure 1. The proposed coding framework operates on groups of pictures, called GOPs. The encoding process starts with the division of the input video sequence into key frames I, which are entropy coded, and WZ frames X, which are encoded using Wyner-Ziv principles - see Fig. 1. These coding modules are explained next.

3.1 Intra-frame codec

The key frames are encoded using the intra-frame codec described in our previous work. Its design is specially tailored to operate on extremely low resolution data and to work in real-time on very limited hardware. In our past work we analyzed the performance of H.264/AVC intra-prediction coding tools on extremely low resolution video. Based on this study, we have limited the processing options of the encoder to a minimum: we allow only one block size, i.e. 4 × 4, and only the DC intra-prediction mode. These two modes performed the best in rate-distortion sense for most of the blocks in extremely low resolution video; in this way, the computationally expensive
mode decision process present in H.264/AVC can be evaded. The block-based DC intra-prediction is followed by $4 \times 4$ integer DCT and quantization.\textsuperscript{21} We note that the input is mirror-padded prior to compression to accommodate an integer number of blocks per dimension, i.e. the video acquired with the sensor\textsuperscript{7} is padded from $30 \times 30$ to $32 \times 32$. The final bitstream is generated by performing entropy coding of the quantized coefficients based on context-adaptive variable length coding.

As shown in Figure 1, the intra-frame decoder follows the processing steps of the encoder in reverse order, i.e. it performs entropy decoding of the bitstream, inverse quantization of the coefficients, inverse DCT and intra-frame reconstruction. More details about the intra-frame codec can be found in our prior publications.\textsuperscript{21,50}

3.2 Wyner-Ziv codec

The proposed DVC follows a closed-loop scheme, whereby the encoder reconstructs the intra-frames to ensure that the same intra-frame predictions are employed at both sides of the transmission channel. As shown in Fig. 1, the reconstructed intra-frames $\hat{I}$ are stored in a frame buffer and subsequently used in the encoding of the WZ frames.

Similar to the intra-frame codec, the WZ codec operates on $4 \times 4$ blocks. This design allows us to reuse the functional blocks implemented for the intra-frame codec, which simplifies the implementation on the sensor node.\textsuperscript{19} The encoder finds the maximal amplitude of each frequency band in the reconstructed intra-frames. These values are used to calculate quantization bin sizes per band for all WZ frames in the GOP, as detailed next. Due to the availability of the same maximal amplitude values at the decoder side, the transmission of the quantization parameters can be bypassed, thereby reducing rate.
3.2.1 Mode selection process

The WZ encoder performs mode decision for each block. The first mode to be evaluated is the skip mode, which is realized by calculating the sum of squared differences (SSD) between the current block and the block located at the same spatial position in the previous frame. Although this evaluation is relatively simple, it estimates quite accurately the amount of motion inside the block. Thus, it allows the encoder to predict the quality of SI at the decoder. Moreover, such evaluation grants that the objective reconstruction quality does not fall below a user-defined level - the decoder can simply copy the collocated block from the previous frame, ensuring a reconstruction error at a certain SSD level.

We note that state-of-the-art feedback-channel free DVC architectures\cite{16,51} perform the skip mode in the transformed domain. However, the proposed spatial-domain design allows for further reducing the computational complexity of the encoder by eliminating the transform and quantization calculations for the skipped blocks.

Besides the skip mode, the mode decision algorithm allows the encoder to divide the WZ-coded blocks into several categories, based on the measured SSD. Each category corresponds to SSD values belonging to specific intervals. The blocks falling in each category are buffered and WZ encoded as explained in the next sub-section.

Increasing the number of separate buffers extends the decoding delay. This can be a significant issue in case of extremely low resolution sensors, as the available memory is limited. To alleviate this problem, the interval limits used to determine the block category are adjusted in real-time by the encoder. In particular, the values are determined online after calculating the SSD of each block using a sliding window procedure. The procedure ensures a quasi-uniform distribution of
the blocks in the different categories. In this way, the codec minimizes the delay introduced by using multiple buffers.

The threshold value which is used for skip mode decision is set as an input parameter together with the intra and WZ quantization parameters. The mode signaling information is multiplexed with the intra and WZ bitstreams and transmitted to the decoder.

3.2.2 WZ coding

Each $4 \times 4$ block that differs significantly from the collocated block in the past frame is signaled as non-skipped and encoded using the WZ encoding scheme. First, such blocks are transformed with the integer approximation of the DCT, as done for the intra-coded frames. Next, the derived DCT coefficients are grouped into 16 coefficient bands and subsequently quantized. The DC and AC coefficients are quantized using uniform and dead-zone scalar quantizers respectively. In particular, each frequency band $\beta$ is quantized into $2^{L_\beta}$ levels ($L_\beta$ bitplanes), in which the number of levels is given in the predefined quantization matrices (QMs). To this end, we modified the QMs from the DISCOVER codec to enable band grouping: specifically, all bands which are concatenated for SW coding should have the same number of quantization levels. The frequency band grouping and the employed quantization matrices are shown in Figure 2.

The conventional designs set the codeword length equal to the number of coefficients
belonging to the same frequency band in the input frame. However, there are only 64 blocks in 32 \times 32 input video (the resolution of the image after mirror padding). Moreover, some blocks are skipped, as explained earlier. Thus, grouping frequency bands together allows for filling in the buffers more quickly, and, therefore, it reduces delay. It also reduces memory usage at the mote.

Similar to state-of-the-art DVC systems, to perform SW encoding, accumulated linear error correcting codes\textsuperscript{24} are employed in our system. In this work, we evaluate the system employing a codeword length of 132 bits. Although this code-length is very small, it minimizes the encoding delay and it allows for matching the available memory on the sensor.

3.2.3 Encoder-driven Rate Allocation

The proposed encoder spends a specific bit budget for channel coding of each bitplane for each frequency band. The appropriate code rates are determined based on offline training, as proposed in.\textsuperscript{42} In the training stage, we ran the proposed DVC on extensive datasets with the feedback-channel enabled, which allows us to measure the actual WZ coding rate. It is important to note that the rates depend on the quantization parameters (on intra-frames and WZ frames), GOP size, frequency band and bitplane index. Furthermore, the amount of syndrome information transmitted to the decoder is selected based on the SSD between the current block and the collocated block in the previous frame. That is, the rates also depend on an SSD interval to which the block belongs. The number of SSD intervals is equal to the number of WZ block categories and is user-defined.

The result of this extensive training stage is a set of look-up tables that store the needed code-rates for specific codec settings. At run-time, the appropriate lookup tables that depend on the GOP size and quantization parameters are employed to allocate rate. More details about this training stage are given in Section 4.
3.3 Wyner-Ziv decoder

Prior to the WZ decoding, the key frames are intra decoded, reconstructed and stored in a buffer to serve as a reference for motion estimation and SI generation. Next, the mode signaling information per $4 \times 4$ block is recovered.

3.3.1 Side-Information Generation

SI generation for extremely low-resolution video is an unexplored topic. Consequently, the proposed architecture incorporates four different SI generation and refinements techniques, allowing us to thoroughly assess SI generation for this type of data, from both complexity and performance point of views.

The first method, referred to as nearest neighbour (NN) technique, simply copies the corresponding block from the nearest key frame (at the beginning of the GOP or the first frame of the next GOP). Although this mechanism is not expected to perform well, it is a lightweight SI generation tool that does not increase the decoder’s complexity. Hence, its application in a low decoding complexity profile is justified. Moreover, half of the WZ coded frames in this setup do not rely on the availability of future key frame, which can reduce the delay and buffering requirements of the decoding process.

The second tested method, namely the weighted average (WA) SI generator, calculates the weighted average of two pixels at the same coordinates in the neighboring intra-coded frames (one from the past and one from the future key frame). The weights are assigned to reflect the temporal distance between the WZ block and the corresponding intra-coded block. Thus, the closer key frame has a larger influence.
Next SI generation technique, called *overlapped block motion compensation* (OBMC)\textsuperscript{31} showed good performance on low-resolution input data in our prior work.\textsuperscript{20} Similar to the MCI algorithm, OBMC starts with performing forward block-based motion estimation with integer-pixel accuracy. The resulting unidirectional motion field serves as basis to estimate a bidirectional motion field between the interpolated frame (SI) and the reference frames (key frames). The resulting forward and backward vectors are refined at half-pel precision before performing motion compensation.\textsuperscript{31} In contrast to MCI, the final interpolation is achieved by calculating an average of a forward and a backward overlapped block motion-compensated frames.

The fourth SI calculation tool employed in our architecture is the *optical flow* (OF) algorithm proposed by Luong et al.\textsuperscript{33} OF calculates the motion between two image frames at every pixel position. Next, the motion estimated between two successive key frames is used for linear view interpolation of WZ frames.

### 3.3.2 Correlation Channel Estimation and Refinement

The virtual correlation channel between the frame to be encoded and the SI is expressed as a memoryless channel of the form:

\[ X = Y + Z, \]

in which \( X \) and \( Y \) are random variables representing the coefficients of a given DCT frequency band in the original WZ and SI frames respectively. \( Z \) is the so-called correlation noise which is modelled as having a zero-mean Laplacian distribution, similar to other works in the DVC literature.\textsuperscript{25} Its probability density function is given by:

\[ f_{X|Y}(x|y) = \frac{\lambda}{2} e^{\lambda|x-y|} \]
where \( \lambda \) is the scaling parameter. During the correlation channel estimation (CCE) process, we calculate a scaling parameter \( \lambda \) for each frequency band \( \beta \) of each WZ frame individually, using the correlation noise frame \( R \). In case of OF and OBMC SI generation algorithms, \( R \) is calculated as a difference between the forward and the backward motion compensated frames. The NN and WA tools do not generate two SI predictions for each WZ frame, and therefore the difference between the previous and the future intra-coded frames is used instead. Then, \( \lambda(\beta) \) is approximated based on the corresponding frequency bands \( \hat{R}(\beta) \) in the transformed correlation noise frame \( R \), namely,

\[
\lambda(\beta) = \sqrt{\frac{2}{\text{Var}[\hat{R}(\beta)]}},
\]

in which \( \text{Var}[:] \) is the variance operator.

This initial estimate is used to drive the model in eq. (2), to obtain the correlation model. This model is converted to soft-input estimates, i.e. LLRs, that provide \( a \, \text{priori} \) information for each bit to be 0 or 1 in each bitplane for every frequency band. The proposed codec follows the same LLRs calculation scheme as our previous work.\(^{20} \) Similarly, the correlation model is refined after each codeword is decoded.\(^{38} \)

3.3.3 Stopping criterion for Slepian-Wolf decoding

In case of a feedback-channel architecture, the rate control is performed through a loop whereby the decoder incrementally asks for more syndrome bits until successful decoding is achieved. The main problem in this approach is to detect when WZ decoding can be stopped. Similarly, if the feedback channel is not present in the system, it is crucial to determine whether or not the decoding attempt was successful for the given amount of syndrome data.
We employed two stopping criteria in the proposed WZ codec. First of all, we check if the soft decoder converges, i.e. if the corrected SI can be used to generate the same syndrome bits as received from the encoder. Unfortunately, the belief propagation decoding algorithm can also lead to such convergence by incorrectly modifying the bits in the SI codeword. The probability of such a "false-positive" decoding increases with decreasing codeword length. Using short codewords, as imposed by the limited available memory, leads to a non-negligible probability of false positives. To counter this effect, we have also employed a second stopping criterion, namely the Sign-Difference Ratio (SDR),\textsuperscript{52} in our system. This simple stopping criterion assumes that only a certain number of SI bits can be corrected with a certain number of syndrome bits. Particularly, considering $L_a^i$ and $L_1^i$ as the \textit{a priori} and \textit{a posteriori} LLRs respectively about bit $i$, decoding is considered successful if:

$$N \cdot T_{SDR} \leq \sum_{i=1}^{K} \delta_i, \quad \delta_i = \begin{cases} 1 & \text{sgn}(L_a^i) \neq \text{sgn}(L_1^i) \\ 0 & \text{otherwise} \end{cases}$$

in which the threshold $T_{SDR}$ is experimentally determined for a known number of $N$ syndrome bits and $K$ SI bits.

3.3.4 Side-Information Refinement

As explained earlier, the employed DVC architecture collects blocks that belong to the same WZ block category. Once the buffer collecting the blocks within a given block category is filled, SW encoding is performed and a code packet is generated. At the decoder side, it usually occurs that not all blocks in the WZ frame are reconstructed after one packet is decoded. Nevertheless, we
exploit the decoded information and perform side-information refinement\(^1\). During this step, SI is generated again by making use of those partially decoded frames (i.e. after decoding all the bitplanes of all frequency bands). The same interpolation refinement is applied to blocks which are signaled as skipped. The mode decision algorithm at the encoder compares each block with its co-located predecessor, hence it is natural to use the temporally neighboring frame for skipped block reconstruction (also in case of the predecessor being a WZ frame).

We note that the proposed architecture does not perform SI refinement after successful decoding of each frequency band, which was suggested in our base design for the same sensor node.\(^20\) This significant change is justified by exhaustive investigations of SI generation techniques (see Section 4) which show that the per-band SI refinement\(^20\) does not bring significant improvements to the SI quality and is too complex. Since the main focus is practical real-time DVC, such computationally expensive procedures had to be dropped in the proposed design.

### 3.3.5 Decoding Recovery Tools

If SW decoding fails, the proposed system tries to recover the data by applying two restoring procedures. First, it attempts to decode less significant bitplanes of the same frequency band group. Although the decoder described in this paper does not refine SI after decoding every single bitplane, each successful decoding imposes changes in LLRs, which can lead to correct SW reconstruction of a codeword which previously failed being decoded.

Another tool adopts the method suggested by Brites and Pereira\(^45\) to the proposed block-based coding scheme. In particular, the decoder recognizes the block which is most likely to be erroneous (which SI is the poorest) and sets its LLRs to 0 prior to soft decoding. This procedure is repeated

\(^1\)Due to its computational complexity, the refinement step can be disabled in some decoding profiles
multiple times, ignoring the SI generated for different blocks at every iteration.

The above-mentioned recovery technique is justified by the fact that motion occurring in $30 \times 30$ pixel video can be much smaller than one block, being nearly impossible to be predicted at the decoder side. If this happens, the LLRs calculated for all frequency bands in this block will introduce misleading information to the SW decoder. Thus, it is reasonable to try to avoid information coming from SI calculated for such blocks.

It is possible, that the recovery tools do not succeed and the SW decoding is unrealisable. If this happens, the proposed system reconstructs the missing bitplanes by copying information from the corresponding SI blocks.

4 Experimental results

The main focus of this work is to design a DVC system for real-world VSNs. Therefore, the proposed DVC encoder was implemented and evaluated on existing hardware. The wireless visual sensor employed in this work is the Silicam IGO\textsuperscript{7} stereo camera depicted in Fig. 3. The device is build around a 16-bit dsPIC microcontroller with 512 KB Flash memory and 52 KB SRAM. It contains a SX1211 radio transmitter and two synchronized ADNS-3080 optical mouse sensors,
<table>
<thead>
<tr>
<th>Name</th>
<th>Framerate</th>
<th>Textures</th>
<th>Motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>cards2</td>
<td>50 fps</td>
<td>simple</td>
<td>slow, limited to a small area</td>
</tr>
<tr>
<td>crowd1</td>
<td>50 fps</td>
<td>complex</td>
<td>slow, in the whole frame</td>
</tr>
<tr>
<td>loundry2</td>
<td>25 fps</td>
<td>complex</td>
<td>slow, limited to a small area</td>
</tr>
<tr>
<td>office2</td>
<td>25 fps</td>
<td>simple</td>
<td>slow, in the whole frame</td>
</tr>
<tr>
<td>piano1</td>
<td>50 fps</td>
<td>complex</td>
<td>fast, limited to a small area</td>
</tr>
<tr>
<td>ragtime1</td>
<td>25 fps</td>
<td>complex</td>
<td>fast, in the whole frame</td>
</tr>
<tr>
<td>squash1</td>
<td>50 fps</td>
<td>simple</td>
<td>fast, limited to a small area</td>
</tr>
<tr>
<td>swing2</td>
<td>25 fps</td>
<td>simple</td>
<td>fast, in the whole frame</td>
</tr>
</tbody>
</table>

which capture stereo video sequences at $30 \times 30$ pixels resolution with 6 bits depth. The video output provided by the sensors is modified prior to compression and transmission by several pre-processing operations, including fixed-pattern noise removal, vignetting correction, temporal low-pass filtering (exponential moving average), and 8 bit rescaling. All these processing operations are done real-time on the sensor. The main frame-rate of the sensor is set to 50 fps in order to eliminate the temporal flickering caused by interference of lights on the power grid, but not all the frames have to be necessarily encoded and transmitted (e.g. every second frame is transmitted for a 25 fps setting). Only one view (recorded with the left camera of the sensor) is used in our experiments.

In order to fully evaluate the complexity and applicability of the proposed codec, the decoder is executed on two devices: (i) a notebook PC running 64-bit Debian Linux equipped with the first generation Intel i5 CPU, 8 GB RAM and SSD drive; (ii) Raspberry Pi 1 B+ single-board computer running Raspbian Linux.

The compression performance of the system is evaluated offline, using two sets of 30 second video sequences recorded at the frame rates of 25 Hz and 50 Hz. Each test set contains four sequences with different motion characteristics, as described in detail in Table 1. The camera remained fixed during recording, which is typical for many VSN scenarios.\(^6\)
Table 2 Test settings

<table>
<thead>
<tr>
<th>setting</th>
<th>$QP_{intra}$</th>
<th>$Q_{WZ}$</th>
<th>$skip$</th>
<th>$T_{SDR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>18</td>
<td>QM3</td>
<td>50</td>
<td>0.17</td>
</tr>
<tr>
<td>1</td>
<td>18</td>
<td>QM3</td>
<td>250</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>QM2</td>
<td>250</td>
<td>0.17</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>QM1</td>
<td>250</td>
<td>0.17</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>QM0</td>
<td>250</td>
<td>0.17</td>
</tr>
<tr>
<td>5</td>
<td>26</td>
<td>QM0</td>
<td>250</td>
<td>0.17</td>
</tr>
<tr>
<td>6</td>
<td>28</td>
<td>QM0</td>
<td>250</td>
<td>0.17</td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td>QM0</td>
<td>250</td>
<td>0.17</td>
</tr>
<tr>
<td>8</td>
<td>32</td>
<td>QM0</td>
<td>250</td>
<td>0.17</td>
</tr>
</tbody>
</table>

4.1 Parameter training

As mentioned in Section 3, optimized codec parameter settings had been determined using an extensive training stage. To this end, we have employed 5 second video sequences containing similar content and having similar characteristics to the selected test data.

All trained parameters used in the experimental evaluations are listed in Table 2 and were selected based on all training sequences (i.e. they remain the same for any input and any framerate). The only exceptions are the rate-allocation settings, which were trained for further investigations both individually (using training data per sequence) as well as when using all training inputs. The necessary rate per bitplane per frequency band group is set to the value which ensures 95% decoding probability and was calculated using the DVC setup with feedback-channel enabled.

During the preliminary evaluation of the system, we observed that the 95% threshold provides the best results in rate-distortion sense - increasing the decoding probability significantly increases the bitrate while only slightly improving quality. Similarly, a threshold equal to 250 for skip-mode decision proved to perform best and was used in all but one configuration (designed to code the image at the highest possible quality). In order to maintain the transmission delay at minimum, we decided to encode all WZ-coded blocks at the same rate. Finally we note, that the
chosen quantization parameter pairs $QP_{\text{intra}}$ and $Q_{WZ}$ (see Table 2) guarantee Peak Signal-Noise Ratio (PSNR) differences between intra-coded and WZ-coded frames to be below 1 dB in case of $GOP = 2$, and up to 1.5 dB for $GOP = 4$. This corresponds to a quasi-constant visual quality of the compressed sequences.

During the training process, we have evaluated the four SI generation techniques described in the previous section. The performance of each method is calculated offline, i.e. the generated SI is compared to the input sequence. The average PSNR results calculated based on intra-coded frames with $QP_{\text{intra}} \in \{18, 20, ..., 30\}$ for different GOP sizes are presented in Table 3. The table reports also average execution times per frame, calculated based on $10^3$ simulations for all GOP and $QP_{\text{intra}}$ combinations mentioned in the table.

As expected, the performance of all SI tools drops with the increasing distance between intra-coded frames. It can be seen from the results that the OF method generates the best results at the highest complexity cost for larger GOP sizes. However, our single-thread implementation of OF is impractical for real-time decoding. The NN technique provides noticeably worse results than all other methods for all GOP settings. Moreover, it is only slightly faster than the WA algorithm. Based on these results, we selected the WA and OBMC techniques for further examination in low-complexity and high-complexity codec profiles respectively.
4.2 Complexity of the system

As explained earlier, the proposed video codec was designed to ensure real-time execution in VSNs. However, encoding and decoding complexities highly depend on video content. Thus, complexity measurements were conducted offline using the mentioned sets of video sequences. We selected coding parameters resulting in high and medium reconstruction qualities and compared the results against the intra-frame system designed for the same sensor in our previous study. In particular, the following pairs were evaluated (see Table 2): (i) $GOP = 2$ setting $= 2$, and (ii) $GOP = 4$ setting $= 0$ providing an average reconstruction quality of about 41.5 dB on the tested data, i.e. nearly lossless quality.

In case of medium quality reconstruction, we evaluated (i) $GOP = 2$ with setting $= 5$, (ii) $GOP = 4$ with setting $= 4$, and (iii) $GOP = 8$ using setting $= 3$, all three settings resulting in 36.5 dB on average. $QP_{\text{intra}} = 21$ and $QP_{\text{intra}} = 27$ were used for the intra-coded experiments.

4.2.1 Encoder complexity

The complexity of the encoder was measured as the number of specific calls executed from the source code. In particular, we counted the number of comparisons in logical statements, the amount of mathematical and bit operations, and, finally, the number of copy operations called from the code. These complexity measures highly depend on the implementation of the algorithm and they are not perfect choices as the compiler modifies the code during the optimization process. However, we believe that they still represent accurate means for comparing the complexity of different configurations.

The results for high and medium quality settings are presented in Fig. 4. Both graphs show the average number of calls per frame. We note that the numbers of comparisons and additions are the
Fig 4  Complexity analysis: average number of specific calls per frame executed at the encoder for different GOP settings. (a) high-quality transmission, (b) medium-quality transmission

highest, and this can be explained by the extensive usage of loops in any coding system (looping over blocks, pixels, bitplanes, etc.).

Compared to intra coding, it can be seen that the proposed DVC architecture reduces the overall complexity of the encoder in both cases, with notable differences in the medium quality range. It is important to observe that the increase of the GOP size results in worse SI (see Table 3). Thus, it is necessary to transmit more information to achieve the same quality (which means more processing at the encoder). Selecting the proper GOP size is thus of particular importance in DVC. In our example, the $GOP = 2$ configuration for high quality and $GOP = 4$ setting for medium quality give the best results, with 34% and 55% complexity reduction respectively when compared to intra-only coding.

The intra-frame video codec proposed in our prior work\textsuperscript{21} achieves 50 fps when encoding a stereo input. Similarly, the proposed DVC system is able to encode video in real-time when executed on the dsPIC microcontroller of the Silicam IGO sensor.\textsuperscript{7}
Table 4 Decoding speed [fps] on various hardware platforms

<table>
<thead>
<tr>
<th>SI generation</th>
<th>WA</th>
<th>OBMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI refinement</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

*high quality, GOP = 2*

<table>
<thead>
<tr>
<th>Hardware</th>
<th>SI gen</th>
<th>SI ref</th>
<th>Raspberry Pi</th>
<th>notebook PC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>41.66</td>
<td>no</td>
<td>41.22</td>
<td>19.78</td>
</tr>
<tr>
<td></td>
<td>770.57</td>
<td>yes</td>
<td>769.33</td>
<td>404.92</td>
</tr>
</tbody>
</table>

*medium quality, GOP = 4*

<table>
<thead>
<tr>
<th>Hardware</th>
<th>SI gen</th>
<th>SI ref</th>
<th>Raspberry Pi</th>
<th>notebook PC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>41.42</td>
<td>no</td>
<td>35.54</td>
<td>34.96</td>
</tr>
<tr>
<td></td>
<td>865.37</td>
<td>yes</td>
<td>758.70</td>
<td>428.21</td>
</tr>
</tbody>
</table>

4.2.2 Decoder complexity

Typically, decoding complexity is overlooked, playing a less significant role in uplink-oriented architectures, such as DVC. The commonly followed logic is that sufficient computing machinery is available at the decoder side to achieve real-time execution. However, achieving real-time decoding is important and not trivial, due to the complexity of the involved tools.

In case of feedback-channel free systems, SI generation and SI refinement are the most complex tools, and therefore different configurations of the decoder were evaluated. In particular, we compared the performance of the decoders employing WA and OBMC SI generation techniques, with SI refinement turned on and off. The average decoding speed results reported in Table 4 show that the proposed configurations fulfill the real-time decoding requirement by a large margin when running the decoder on a notebook PC. These results are due to extremely low video resolutions, which makes the decoding fast, but also due to the extensive use of the skip mode.

The low decoding complexity of the proposed architecture is also proved while running the decoder on Raspberry Pi single-board computer. In average, the hardware can process 41.66 and 41.42 frames per second for high and medium quality settings respectively (for decoders employing WA for SI generation). However, it is important to note that these results highly depend on video
content. In particular, the system can decode over 60 frames per second in case of highly static movies, such as *loundry2* or *piano1*. For scenes where changes are not limited to small regions, such as *office2* or *swing2*, Raspberry Pi provides average decoding speeds of around 20 frames per second. The decoding speed drops if more computationally demanding OBMC tool is used for SI generation, however this decoding profile was designed for more powerful devices.

Although the high quality setting utilizes smaller GOP size (i.e. there are less WZ-coded frames in this setting), the decoding speed is very similar to the medium quality setting. This is caused by the different value of skip threshold parameter used in the two GOP configurations, resulting in the smaller number of WZ encoded blocks in the second case. Moreover, different quantization settings result in the higher number of bitplanes which are transmitted in the first scenario.

### 4.3 Video codec performance

The performance of the proposed video codec was compared against our prototype DVC system designed for the same sensor\(^2\) and three intra-coding architectures, namely Motion JPEG (MJPEG),\(^{15}\) H.264/AVC High Profile\(^2\) and our intra-frame codec proposed in the past,\(^{21}\) which is also used for intra-frame coding in the proposed DVC system.

We evaluated four different settings. Firstly, we assessed the reconstruction quality of the best possible setting, i.e. the system exploiting the feedback channel and employing the OBMC SI generation method; this configuration is referred to as "Fbck". Next, we report the results of feedback-channel-free architectures, including a low-complexity decoding profile (denoted as "Low") and a high-complexity decoding profile (denoted as "High"). These two configurations yield the lowest and the highest decoding speeds in the complexity analysis reported in Table 4;

\(^2\)The High Profile of the H.264/AVC standard supports monochrome coding.
Table 5 Video codec performance: BD-rate results with respect to H.264/AVC intra [in %]; negative figures correspond to rate gains.

<table>
<thead>
<tr>
<th>GOP 1</th>
<th>GOP 2</th>
<th>GOP 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intra</td>
<td>MJPEG</td>
<td>DVC</td>
</tr>
<tr>
<td>cards2</td>
<td>9.10</td>
<td>157.96</td>
</tr>
<tr>
<td>office2</td>
<td>11.37</td>
<td>148.87</td>
</tr>
<tr>
<td>piano1</td>
<td>8.06</td>
<td>143.32</td>
</tr>
<tr>
<td>ragtime1</td>
<td>9.51</td>
<td>191.33</td>
</tr>
<tr>
<td>squash1</td>
<td>7.67</td>
<td>231.94</td>
</tr>
<tr>
<td>average 25fps</td>
<td>9.71</td>
<td>207.36</td>
</tr>
<tr>
<td>average 50fps</td>
<td>8.51</td>
<td>181.62</td>
</tr>
</tbody>
</table>

that is, this comparison measures the loses caused by disabling SI refinement and using WA for SI generation. Lastly, we present the performance of the "High" setting whereby the rate allocation module is individually trained (denoted "High,T"). Including these results allows us to assess if the system is vulnerable to inaccuracies on the rate control parameters.

Concerning the use of OBMC for SI generation, it is important to remark the following. As showed in Table 3, we note that OBMC yields slightly worse performance than WA during offline SI experiments. However, compared to WA, the OBMC algorithm can be used much more efficiently during CCE as it generates two predictions instead of one. This results in better overall performance of the DVC system when OBMC is used for SI generation in place of WA. This justifies the choice of OBMC in this set of online experiments.

Table 5 shows the average rate differences per sequence between the rate-distortion curves provided by the different systems. The average rate differences are calculated in Bjontegaard sense. H.264/AVC intra is used as reference.

Our intra-frame codec achieves uncompetitive compression performance in all bitrate ranges, losing around 10% rate in average relative to H.264/AVC intra. This is caused by the fact that

---

3We employed the JM18.6 reference software, available online at http://iphome.hhi.de/suehring/tml/download/
the retained prediction modes are not always optimal in rate-distortion sense. Moreover, larger block sizes can be very efficient for encoding large textureless regions (e.g. the background in the 
Squash1 sequence). The performance of this simple intra-coding system increases at lower rates, which is explained by the large overhead of signaling modes in H.264/AVC.

The MJPEG codec performs much worse than the two intra-frame competitors mainly due to the larger block size ($8 \times 8$), which is not well tailored for extremely low resolution videos. Furthermore, its Huffman entropy codec optimized for high resolution images fails to yield competitive performance against CAVLC.

It can be seen from Table 5, that the proposed DVC architecture employing feedback channel achieves very similar results to its prototype. The results also reveal the superiority of the proposed block-based DVC scheme over our previous design performing WZ coding at frame level. This highly competitive coding performance is offered despite of the fact that the proposed DVC (i) uses a less efficient intra-frame codec (when compared to the H.264/AVC intra used in the reference DVC system), (ii) it employs no refinement steps after successful decoding of each frequency band, and (iii) it uses a much shorter codeword. The largest difference relative to the reference DVC architecture appears on the Swing2 sequence. The results on this sequence are depicted in Fig. 5. In sequences with fast and complex motion, such as Swing2, the skip mode is selected less often, which increases the differences relative to the reference DVC architecture.

A comparison of the results obtained with global rate control parameters and rate control parameters trained individually is given in Fig. 5 for the Piano1 sequence. This sequence contains very irregular motion in a small part of the frame only (the motion occurs more than 1 meter away from the camera), which is expected for extremely low resolution sensors employed in surveillance applications. These results show that difference between the proposed feedback-free architec-
Fig 5 Video coding performance comparison. (a) Swing2 sequence, GOP = 4, (b) Piano1 sequence, GOP = 2.

The proposed feedback-free design outperforms H.264/AVC intra (which is too complex for real-time implementation on the sensor node) and our simplified intra-coding scheme providing up to 57% and 65% rate saving respectively (for GOP = 4 setting). The graphs averaging the video coding performance results for 25 fps and 50 fps video sequences are showed in Fig. 6.

The experimental results also reveal the trade off between the decoder complexity and compression performance. The architecture making use of OBMC as SI refinement ("High") improves the reconstruction by around 1 dB when compared to the low complexity system ("Low"). It is important to note that the quality difference is even larger if only WZ-coded frames are considered.
Fig 6  Video coding performance comparison. (a) average over all 25 fps sequences, (b) average over all 50 fps sequences

(the presented results account also for intra-coded frames). We note that the encoded bitstream size (i.e. the bit-rate) is the same in both cases.

5 Conclusions

In this paper we present a novel DVC solution for video transmission in wireless VSNs. The proposed system enables efficient compression of extremely low resolution visual data. The introduced solution targets real-world applications, and, to the best of our knowledge, it represents the first practical DVC system for low-resolution VSNs. The proposed system lies in strong contrast with alternative DVC systems in the literature (including our own past DVC designs), which focus on maximizing compression performance and overlook severe design constraints imposed by actual deployments in practical applications.

The proposed DVC system targets high-efficiency and low encoding complexity. Its low computational demands and low buffering requirements are proven by implementing and running the encoder on the dsPIC microcontroller of the considered visual sensor mote.\textsuperscript{7}
The complexity of the proposed WZ encoder is lower than that of the state-of-the-art intra-frame codec for extremely low resolution video. Secondly, the proposed system includes multiple decoding tools and configurations, which can satisfy different hardware profiles. The thorough evaluation of multiple SI generation tools working in feedback-enabled and feedback-free DVC architectures reveals complexity versus compression performance trade-offs. Moreover, decoding speeds of various DVC configurations running on a low-end notebook PC and a Raspberry Pi single-board computer were tested. This evaluation reveals the codec configurations that yield real-time decoding performance.

Finally, the proposed system is assessed against several reference video codecs typically used in wireless VSNs. Notable compression gains when compared to H.264/AVC intra and MJPEG prove that carefully designed DVC can be a solid alternative to conventional codecs in practical VSNs.

Acknowledgments

This work was supported by the Flemish Institute for the Promotion of Innovation by Science and Technology (IWT), PH.D. Grant No. 111014 of Jan Hanca, the VUB strategic research programme M3D2 and National Fund for Scientific Research (FWO) project G004712N.

References


2 N. Deligiannis, F. Verbist, J. Barbarien, J. Slowack, R. Van de Walle, P. Schelkens, and


9 J. Fernández-Berni, R. Carmona-Galán, J. A. Leñero-Bardallo, R. Kleihorst, and Á. Rodríguez-Vázquez, “Towards an ultra-low-power low-cost wireless visual sensor node


26 C. Brites and F. Pereira, “Correlation noise modeling for efficient pixel and transform domain


41 D. Varodayan, D. Chen, M. Flierl, and B. Girod, “Wyner-ziv coding of video with unsu-


**Jan Hanca** is a PhD student at the Department of Electronics and Informatics at Vrije Universiteit Brussel. He received the MSc and Engineer degrees in Electronics and Telecommunications from Poznan University of Technology, Poland, in 2010. In 2011 he obtained a grant from Flemish agency for Innovation by Science and Technology (IWT) for the preparation of his doctoral thesis.

His professional interests include video compression and depth estimation from multiview sequences.

**Nikos Deligiannis** is assistant professor with the Department of Electronics and Informatics at Vrije Universiteit Brussel. His current research interests include big data processing, compressed sensing, the internet of things, distributed computing, and visual search. He has authored over 60 journal and conference publications, book chapters, and holds two patent applications. He received the 2011 ACM/IEEE International Conference on Distributed Smart Cameras Best Paper Award and the 2013 Scientific Prize FWO-IBM Belgium.
Adrian Munteanu is professor at Vrije Universiteit Brussel, Belgium. His research interests include image, video and 3D graphics compression, error-resilient coding and multimedia transmission over networks. He is the author of more than 250 journal and conference publications, book chapters and contributions to standards, and received several awards for his work. Adrian Munteanu currently serves as Associate Editor for IEEE Transactions on Multimedia.

List of Figures

1 The block diagram of the proposed codec.
2 (a) Frequency band grouping; (b), (c), (d), (e) quantization matrices employed in the proposed architecture, referred to as QM0, QM1, QM2 and QM3 respectively.
3 The Silicam IGO7 sensor.
4 Complexity analysis: average number of specific calls per frame executed at the encoder for different GOP settings. (a) high-quality transmission, (b) medium-quality transmission
5 Video coding performance comparison. (a) Swing2 sequence, GOP = 4, (b) Pianol sequence, GOP = 2
6 Video coding performance comparison. (a) average over all 25 fps sequences, (b) average over all 50 fps sequences

List of Tables

1 Test video sequences
2 Test settings
Side-Information generation techniques evaluation

Decoding speed [fps] on various hardware platforms

Video codec performance: BD-rate results with respect to H.264/AVC intra [in %]; negative figures correspond to rate gains.