



HAL
open science

SenseVid: A traffic trace based tool for QoE Video transmission assessment dedicated to Wireless Video Sensor Networks

Moufida Maimour

► To cite this version:

Moufida Maimour. SenseVid: A traffic trace based tool for QoE Video transmission assessment dedicated to Wireless Video Sensor Networks. *Simulation Modelling Practice and Theory*, 2018, 87, pp.120-137. 10.1016/j.simpat.2018.06.006 . hal-01881197

HAL Id: hal-01881197

<https://hal.science/hal-01881197>

Submitted on 25 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SenseVid : A Traffic Trace based Tool for QoE Video Transmission Assessment Dedicated to Wireless Video Sensor Networks

Moufida Maimour^{a,b,*}

^aUniversité de Lorraine, CRAN, UMR 7039
Campus Sciences, BP 70239, Vandoeuvre-lès-Nancy Cedex, 54506, France
^bCNRS, CRAN, UMR 7039, France

Abstract

Video applications are being a key component to enhance traditional wireless sensor networks (WSN) applications. As a result, sensor network researchers need adequate and easy to use tools to assess the performances of their proposals. The existing transmission evaluation tools either consider video sequences along with codecs that are unsuitable for WSNs or make use of low cost compression methods for still images without inter-frame coding required for efficient video transmission. In this paper, we present SenseVid, an open source video transmission and evaluation tool that considers WSN specific characteristics. Besides low energy intra-frame compression based on fast pruned discrete cosine transform (DCT), a low complexity inter-frame encoding is provided to allow efficient support of video flows. A configurable fine-grain energy model is provided where both video capture and encoding cost are accounted for on a per frame basis. Video traffic differentiation based on priority levels is also provided. SenseVid adopts the video traffic traces approach, allowing its use in any simulation or real testbed environment. Using SenseVid, the user is able to reconstruct the received video considering lost packets during its transmission as well as estimating the achieved quality of service (QoS) and quality of experience (QoE).

Keywords: Wireless Video Sensor Networks (WVSN), Video Quality Evaluation, SenseVid, QoS, QoE, Energy.

1. Introduction

The emergence of low-cost and low-power visual modules [1, 2, 3] fostered the development of wireless video sensor networks (WVSN) and paved the way to a plethora of applications mainly in monitoring and surveillance fields. WVSNs generate unique challenging problems as the capture, the encoding and the transmission of video flows are resource-hungry while the underlying network is constrained in terms of energy, bandwidth, processing and storage means. As a result, a significant effort has to be made in the design and development of new algorithms and protocols where the network limited resources are saved while providing a good quality of service (QoS) to the end user.

Network researchers usually make use of discrete event network simulators such as ns2 [4] and OMNeT++ [5] to develop and evaluate their algorithms and protocols. Compared to a real WSN testbed, network simulators are good means to easily test protocols in a large scale with very limited cost. However, with respect to WVSN, pure simulation testing may not accurately reproduce real-life scenarios and may not provide appropriate metrics to assess proposed protocols and/or video encoding techniques. Metrics such as packet loss ratio, throughput, delay and jitter remain pure network related metrics that provide only limited insight into the video quality perceived by a human and do not reflect its associated subjective factors. One step forward to achieve more accurate assessment of network proposals is to be able to estimate the received video quality. Human perception can be evaluated using Quality of Experience (QoE) metrics such as structural similarity index (SSIM) [6]. This requires simulated experiments to be performed using actual encoders, decoders and video data. EvalVid [7] made it possible by providing a set of tools, that allows researchers to carry out simulations of real video sequences transmission. EvalVid takes the video traffic

*Corresponding author

Email address: moufida.maimour@univ-lorraine.fr (Moufida Maimour)

trace approach to characterise a video sequence where trace files with limited size are produced rather than using actual video bit stream [8].

In the context of WWSN, Authors of [9] proposed to integrate EvalVid along with WiSE-MNet [10] in the Castalia simulator [11]. However, standard video coding techniques based on motion estimation algorithm such as MPEG-4, H.263 or H.264 supported by EvalVid are not suitable for sensor nodes [12]. Even JPEG's low complexity still image compression algorithm is not very beneficial in terms of power consumption. This is mainly due to the discrete cosine transform (DCT) stage which consumes at least 60% of the whole power encoder [13]. Therefore, efficient compression algorithms, in terms of power dissipation, are mandatory to handle video flows. To fit WSN needs, WWSN model [14] integrates the low-cost compression method proposed in [15] into the Castalia simulator source code. However, the image bit stream adopted makes the WWSN model usable only with the Castalia simulation environment. Sim-LIT [16] is another tool targeted to image transmission designed to assess *block interleaving* algorithms using the TOSSIM simulation environment [17]. Both WWSN model and Sim-LIT are inefficient for video transport since they are only concerned with intra-coded still images. EvalVSN [18] based on Matlab, adopts the traffic trace method along with an inter-frame coding to consider video flows. However, it suffers from a heavy execution time as well as poor performances due mainly to the adopted inter-frame coding and the substantial length of the generated traffic traces.

The existing transmission evaluation tools either consider video sequences along with codecs that are unsuitable for WSNs or make use of low cost compression methods for still images without inter-frame coding required for efficient video transmission. In this paper, we propose *SenseVid* (publicly available at [19]), a video transmission and evaluation tool targeted to WWSN by implementing low complexity inter-frame coding as well as reduced cost DCT intra-frame coding. *SenseVid* adopts the video traffic trace approach, allowing its use in any simulation or real testbed environment provided that a trace-based transmission application is available. Currently, we provide an application module for the Contiki operating system [20] along with tools and scripts to automate the evaluation process. Other similar application modules can easily be developed for other environments. We implemented the PSNR (Peak Signal to Noise Ratio) and the SSIM (Structural SIMilarity) [6] as QoE metrics to assess the quality of both the encoded video before transmission and the received video with respect to the transmitted one. Finally, a configurable fine-grain energy model is provided where both the cost of the capture and the encoding of the video are accounted for on a per frame basis.

The remainder of this paper is organised as follows. Section 2 overviews the existing tools targeted to image or video transmission and evaluation along with some background on DCT-based compression algorithms that are suitable to WWSN. The *SenseVid* architecture and its main functionalities are presented in Section 3 while Section 4 summarises main numerical results. Section 5 gives use scenarios in a simulation environment (cooja [21]) as well as in a real sensor testbed (IoT-LAB [22]). Section 6 concludes the paper with some future directions.

2. Background and Related Work

EvalVid [7] is a popular framework used in the network research community to evaluate video transmission through QoE metrics such as PSNR in addition to other classic QoS metrics like delay, jitter and loss rate. Evalvid takes the traffic trace approach where, based on an input video and a given codec, a video trace file is generated. Each line of the trace file gives mainly, for each frame, its type and its size. A sender trace file is also generated with mainly the transmission time and the size of packets to be sent. The sender trace file can be used by any simulator or real application to emulate the transmission of a given video. Based on the receiver trace file that gives the list of the received packets, Evalvid rebuilds the received video and computes the achieved QoS and QoE metrics. Application modules that allow the use of EvalVid trace files with ns2 and NS3 have been proposed in [23] and [24] respectively.

Since EvalVid was developed to be used in less constrained networks, no energy model is provided. Moreover, the supported encoders, namely, MPEG-4, H.263 or H.264 are resource-hungry and as a result are not adapted to constrained networks such as WSN. These codecs concentrate the most computational complexity at the encoder while video sensors have neither sufficient processing power nor enough energy to perform complex compression algorithms. Nevertheless, compression is still needed to reduce the amount of transmitted data and thus save bandwidth, transmission time and energy dissipation. Hence, efficient compression algorithms, in terms of both processing and energy requirements, are mandatory to handle video transmission in WSN. Compared to lossless compression, lossy one is more suitable to WSN since it allows higher compression ratios which consume less communication resources.

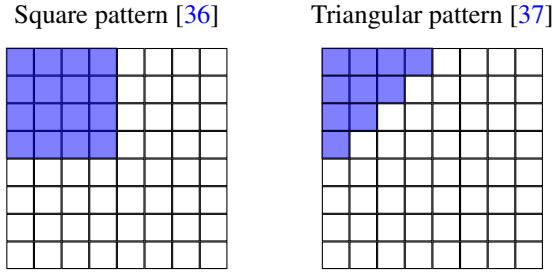


Figure 1: Pruned DCT.

Transform-based compression algorithms such as DCT-based ones are generally preferred over non-transform-based ones due to the fact that their encoder is less complex [25].

The most popular compression scheme based on DCT is JPEG [26]. The image is divided into small 8×8 blocks on which the DCT is applied to separate the high and low frequency information. To compress the image, a quantisation is applied on each DCT block to discard the coefficients with the least information (highest frequency components). A zigzag scan is then applied to reorder coefficients from lower to higher frequencies. Finally, an entropy coding is applied. It generally consists in a run-length encoding (RLE) followed by a Huffman or arithmetic coding. The DCT transform results in a low memory usage since it operates on small blocks and allows acceptable compression ratios. However, as reported in [13], DCT computation consumes at least 60% of the whole power encoder. Even more, authors of [27] have shown that compressing, using JPEG, and transmitting an image consumes more energy than simply transmitting the uncompressed image.

Research effort has gone into the development of methods to decrease the DCT computational complexity and thus reducing its energy footprint. *Fast DCT* computation algorithms mainly eliminate redundant operations through factorising the DCT matrix [28] and reducing the amount of multiplications, known to be greedier operations. Loeffler et al. [29] proposed a fast DCT called *LLM* that requires only 11 multiplications, the minimum achievable multiplicative complexity [30]. Based on the lifting scheme [31], authors of [32] proposed *binDCT*, a multiplierless version of DCT based only on shift and addition operations. The 2-D *binDCT* allows a 16-bit integer implementation that further reduces the DCT complexity. In the context of WSN, authors of [33] use JPEG with integer DCT kernel instead of the commonly used floating-point DCT. They show that encoding a grayscale QCIF image at 1 bpp using the processor StrongARM SA1110 requires only 2.87 mJ with the integer version while the floating-point one requires more than 22 mJ. The work in [34] proposes a video compression method for WSNs where the first frame is encoded in intra mode (I-frame). For subsequent ones, a low complexity change detection algorithm is applied instead of the JPEG motion estimation algorithm. Only a subset of blocks are considered and classified in the order of their importance. A reference frame that contains all blocks is periodically inserted. Also, if more than 75% of the blocks in a frame are marked then this frame is encoded in intra-mode.

Pruned DCT [35] is another method that allows reducing the complexity of DCT computation where only a subset of the DCT coefficients are computed (Figure 1). This is motivated by DCT energy compaction property where the useful signal information is concentrated in the lower DCT coefficients. Moreover, high frequency coefficients are likely to be zeroed by the quantisation process. Mammeri et al. in [36] produced an energy aware adapted version of JPEG called *squared JPEG* (S-JPEG) where only the upper-left squared portion of each DCT block is computed. In [37, 38], the quality of non-ROI (region of interest) part of the image is sacrificed by reducing the amount of DCT coefficients processed, while maintaining the quality of the ROI part. Authors of [15] combined the pruned DCT [35] with a fast DCT algorithm and proposed *fast zonal DCT* with both square [36] and triangular [37] forms of the pruned DCT along with LLM algorithm to fast compute the DCT.

WWSN model [14], an extension of the Castalia simulator, adopts the fast pruned DCT [15] to compress the captured images to fit WSN constrained resources. Instead of the disk model, the sensing range of camera nodes is defined as a triangular FoV (Field of View) depending on the direction of the camera, its angle and depth of view. WWSN model introduces the notion of cover-set and proposes an algorithm for coverset computation and node scheduling [39] where camera nodes with more covers can increase their frame capture rate. An energy model is also

provided. However, WWSN model does not consider efficient video transmission since the captured images are only intra-coded.

Sim-LIT (SIMulator for Lossy Image Transmission) [16] is an evaluation tool targeted to compressed and uncompressed still image transmission. It was mainly designed to assess *block interleaving* algorithms in the presence of different pattern losses. Losses are generated either randomly or based on a text loss-file provided by the user. In this way, as claimed by the authors, it is possible to connect with WSN simulators such as TOSSIM [17] as well as real experiment testbed. Nevertheless, the focus on still images and the lack of an inter-frame coding makes the transmission of video flows inefficient given the constrained nature of WSN. Moreover, aside from the PSNR metric, the authors do not mention any energy related metric to assess the performance of the proposed schemes.

Authors of [40], implemented Joined-JPEG using the Castalia simulator environment by embedding the JPEG and JPEG2000 image compression libraries in a new developed application. This is done in an attempt to mimic their previously designed hardware testbed [41] for WMSN by tuning the simulation parameters to match the real-life measurements obtained via experimental testing. The application provides statistics such as the length of path taken by each packet, the image transmission time, and the PSNR of the received images. Once again, by adopting the bit stream method, Joined JPEG can only be used with Castalia. Moreover compression using JPEG2000 is very time and energy consuming [42], and their efficient use in WWSN is still questionable.

WiSE-MNet++ [43] is an extended version of WiSE-MNet (Wireless Simulation Environment for Multimedia Networks) [10] that includes the implementation of several state-of-the-art algorithms for smart-camera networks. WiSE-MNet++ addresses the need for co-design of network protocols and distributed algorithms for WMSNs. WiSE-MNet++ extends the Castalia/Omnet++ sensing model to support complex vectorial data, such as video and to consider directionality of camera FOVs. At the physical process module, WiSE-MNet proposes the use of moving objects with different types of motion. These objects are represented as a box rather than points to enable object detection. The physical module is also extended to allow feeding the sensor manager with both real world and synthetic datasets. At the application level, several algorithms such as distributed tracking algorithm are developed to demonstrate the effectiveness of WiSE-MNet in modeling cooperative applications. WiSE-MNet++ represents a major step forward towards the valuation of distributed vision algorithm along with communication issues while providing an energy model for the camera systems. Nevertheless, it does not consider encoding techniques that are suitable to WSN and thus fails to achieve a fine-grain modeling of their power consumption. Moreover, It does not provide QoE metrics, an important feature in the process of video transmission evaluation.

M3WSN (Mobile MultiMedia Wireless Sensor Network) [9] is an OMNeT++ framework developed to support transmission, control and evaluation of real video sequences in mobile WMSNs. It builds on the fact that WiSE-MNet does not provide QoE support and combines WiSE-MNet and EvalVid as well as the WWSN model in order to benefit from their respective features as they are complementary. In addition to mobility support using BonnMotion [44], M3WSN allows including the energy consumption for retrieving each frame that composes a video according to values from CMUcam3 [45]. M3WSN makes use of traffic trace files generated by EvalVid and hence fails to consider WWSN constraints since EvalVid does not make use of adapted encoders. Table 1 summarises the tools reviewed in this section.

WWSN model, Sim-LIT and Joind-JPEG are based on the bit stream method that does not allow their easy use in other simulation or real experiment environment. M3WSN is based on a traffic trace approach however it is highly dependent on the Castalia simulator and can not be used in other simulation environments. EvalVSN [18] developed using Matlab, follows the same principle of EvalVid [7] by adopting the traffic trace method. It implements a modified version of MPEG-2 similar to the one proposed in [46] with priority assignment but additionally provides different DCT implementations. An application module that allows the use of EvalVSN with more realistic loss patterns is provided in the ns2 [4] simulation environment. Both PSNR and SSIM QoE metrics are implemented. EvalVSN supports video transmission since it implements inter-frame coding. However, it suffers from a heavy execution time as well as poor performances. On the one hand, the adopted inter-frame coding lacks efficiency in terms of compression ratio mainly for video sequences that result in higher GoP (Group of Picture) lengths. On the other hand, the big size of the traffic traces generated makes them useless in a real testbed environment. Finally, no energy-related metric is implemented in the EvalVSN framework.

SenseVid is a step forward in the development of an open source video transmission and evaluation tool targeted to WSN. It adopts the traffic trace model that allows its use in any simulation environment as well as real sensor network testbeds that lack camera nodes. Besides low energy intra-frame compression based on fast pruned DCT, a

Table 1: Summary Table of Existing Tools

Tool	Evalvid [7]	Sim-LIT [16]	WVSN Model [14]	WiSE-MNet++ [43]	M3WSN [9]	Joined-JPEG [40]	EvalVSN [18]	SenseVid
Language	C	C++	C++	C++	C++	C++	Matlab	C++
Open source	✓		✓	✓				✓
Video support	✓			✓	✓		✓	✓
Codec	MPEG-4 H.263/4	custom	custom	N/A	MPEG-4 H.263/4	JPEG JPEG2000	custom	custom
Bit stream		✓	✓	✓		✓		
Simulator	ns2	Tossim	Castalia	Castalia	Castalia	Castalia	ns2	cooja
Real testbed	✓	✓					✓	✓
QoE metrics	PSNR SSIM	PSNR	PSNR	PSNR SSIM	PSNR SSIM	PSNR	PSNR SSIM	PSNR SSIM
Other built-in metrics	loss delay jitter	loss	energy	energy	loss delay jitter	delay energy	loss	loss energy delay

low complexity inter-frame encoding is provided to allow efficient support of video flows. In addition to QoE metrics such as PSNR and SSIM, a consumption power model is provided to be able to estimate the consumed energy by video capture as well as its compression. Finally, SenseVid allows to differentiate priority levels of the data to transmit in order to adapt to network constraints and dynamics.

3. SenseVid Architecture and Main Functionalities

SenseVid (Figure 2) follows the main principle behind EvalVid where the interface with the real or simulation-based experiment is insured by the use of traffic trace files. Based on the user parameters, the encoder/packetiser module generates two trace files. The first one is the frame trace file (*st-frame*) that records required information on the encoded frames. The second one is the sender trace file (*st-packet*) that gives the sender application, the required information on the data packets to be sent. The receiver application records the received data packets in a receiver trace file (*rt-packet*). Based on the latter, SenseVid reconstructs the video clip allowing its subjective assessment by the user. Additionally, it generates statistics (both QoS and QoE metrics) on the conducted experiment. SenseVid implements a fine-grained energy model that allows the estimation of required energy to capture and encode each frame. A prioritisation procedure is also implemented in order to be able to adapt to network dynamics by allowing or not the transmission of less important data.

3.1. Video Preprocessing

In order to be able to use reference video sequences such as those available at [47] while being able to make them more realistic with respect to WSN characteristics, we introduce an optional preprocessing phase. The video preprocessing module allows to resize the video frames mainly to a lower resolution in order to reduce the required transmission rate. For the same purposes, the video original FFC (frame frequency capture) can be lowered. sending video sequences in a 30 or even 25 fps (frames per second) is simply infeasible due to the limited resources of a WSN. A gray scale QCIF (176×144) video compressed at 0.5 bpp requires a useful bit rate of 316.8 kbps when transmitted at 25 fps. With the same compression ratio, for instance, the micaZ [48] having a bit rate of (250 kbps), only allows a maximum theoretic frequency of 19 fps and even less since this maximum value is difficult to achieve due mainly to the concurrent nature of wireless media.

3.2. Video Encoder/Packetiser Module

Based on the user parameters, the video encoder/packetiser module compresses the captured video clip and generates the sender trace files namely *st-frame* and *st-packet*. It converts the captured frames to gray scale where each

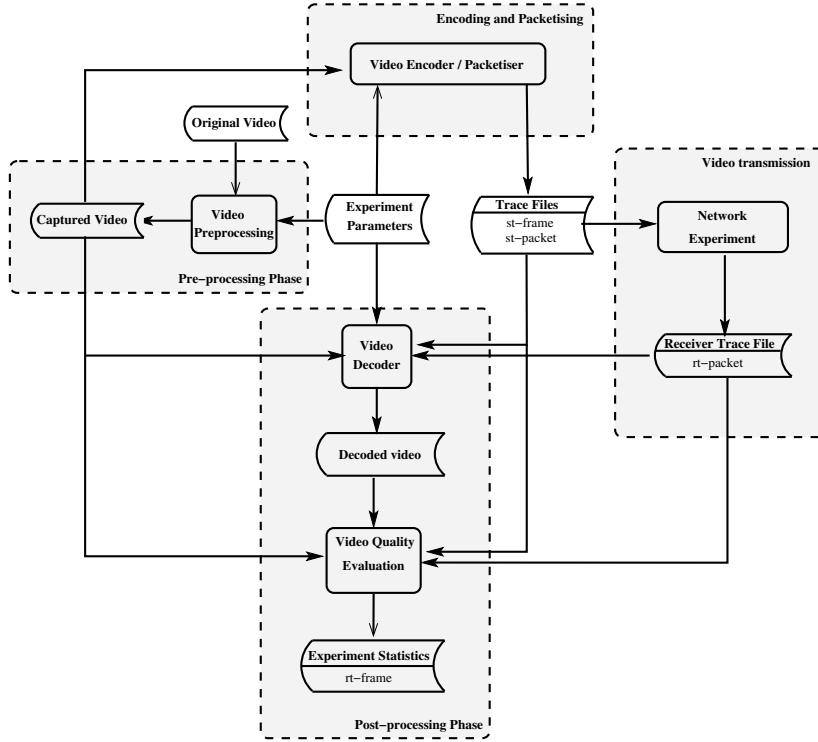


Figure 2: SenseVid Architecture

pixel is encoded using 8 bits with range $[0, 255]$. A low energy compression algorithm that considers both spatial and temporal redundancy in a video sequence is implemented. A frame is either intra-coded and qualified as a *main frame* (M-frame) or inter-coded with respect to the previous M-frame in which case it is referred to as a *secondary frame* (S-frame). The first frame is always encoded as an M-frame. A subsequent frame is M-encoded if it is sufficiently different from the previous M-frame ; otherwise it is S-encoded.

Whether a given frame is encode as a main or a secondary frame is based on a user parameter γ (the GOP coefficient) that allows to control the GOP of the resulting compressed video. γ takes positive values ranging from 0 to the maximum mean square error ($V_{peak} = 255$) two frames can have. For each frame, the mean square error (MSE) with respect to the previous M-frame is computed. If $MSE > \gamma^2$ then the frame is considered to be sufficiently different from the previous M-frame and thus is M-encoded ; otherwise, it is S-encoded. Note that if $\gamma = 0$ then all frames are M-encoded. To rise the number of S-frames, γ has to be increased. Setting γ to 255 results in all frames to be S-encoded except for the first frame.

3.2.1. Main Frame Encoding

As depicted in Figure 3, each M-frame is first decomposed into blocks of 8×8 each of which is shifted from range $[0, 255]$ to signed integers range $[-128, 127]$. Then, a DCT is applied on each block. In order to meet the requirements of constrained WSN, two fast pruned DCTs are implemented, the Loeffler-Ligtenberg-Moschytz DCT (LLM) [29] and the binDCT-C (BIN) proposed in [49]. Both the square and the triangular patterns of pruned DCT are provided. The square or triangle side length ρ is defined by the user. The default value is set to 8. For comparison purposes, we also implemented the traditional DCT variant, we refer to as CLA. The resulting DCT block coefficients are quantised using the JPEG standard quantisation matrix. Trade-off between quality level and compression rate can be obtained by selecting a proper quality factor (QF) that allows adjusting the quantisation matrix values. The user is able to decide the frame visual quality ranging from the poorest ($QF = 1$) to the best quality ($QF = 100$).

The obtained block (B) is then linearised using the traditional zigzag scan along with an optional prioritisation

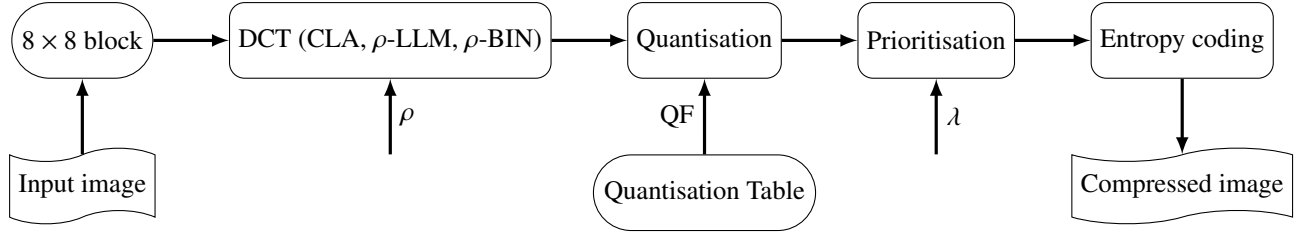


Figure 3: M-frame block encoding sequence

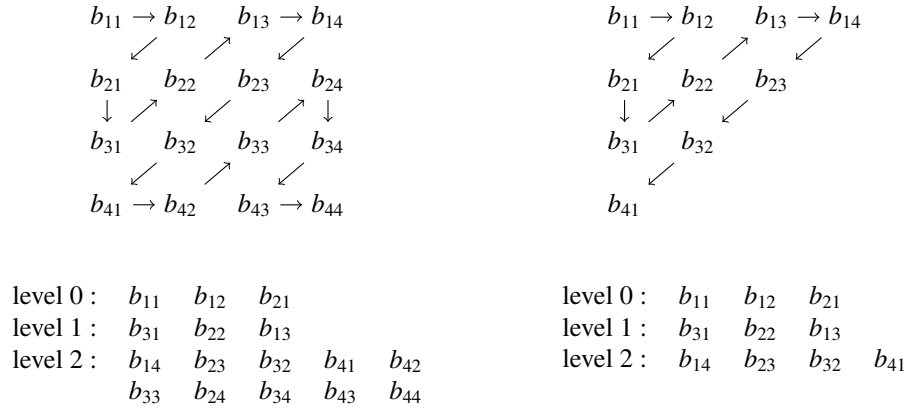


Figure 4: Linearisation and priority assignment (a) square (b) triangular pruned DCT. $\rho = 4$ and $\lambda = 3$.

procedure. For an M-frame, priorities range from 0 to 12 and follow the zigzag scan of the block. For the maximum number of priorities ($\lambda = 13$), the first priority level ($l = 0$) contains the DC component (b_{11}) and the following two AC components (b_{12} and b_{21}). The last priority level ($l = 12$) is composed of the last three coefficients of the linearised block : b_{78} , b_{87} and b_{88} . The remaining block coefficients are assigned to levels $l \in [1, 11]$ where coefficient b_{ij} gets priority level $l = i + j - 3$. Figure 4 shows the zigzag scan and priority assignment when the user selects three priority levels ($\lambda = 3$) and a fast pruned DCT with side length $\rho = 4$. Both the square and triangular shapes are illustrated.

Finally a lossless entropy encoding is applied. The user can choose among various encoders such as Huffmann and exponential-Golomb [50] that follow or not a Run Length Encoding (RLE). The default entropy coder consists in coding all coefficients using exponential-Golomb code.

3.2.2. Secondary Frame Encoding

A secondary frame is inter-coded with respect to the previous main frame. That is, what is encoded is the difference between the current frame and its main frame. The resulting difference blocks with only zero values are ignored since they correspond to blocks that are exactly the same as the ones in the previous M-frame. In order to reduce the amount of data to be transmitted, only a subset of the remaining blocks is encoded. Blocks deemed to be similar to the previous M-frame blocks get lower priority, lossy compressed or simply discarded depending on two parameters provided by the user required by the prioritisation and the thresholding steps (Figure 5).

The prioritisation step consists in attributing a priority level to each block. Priority levels range from 0, the highest, to 4, the lowest priority level. Highest priority blocks are those with the least degree of similarity with their related blocks in the previous M-frame. The five priority levels are related to the MOS (Mean Opinion Score) ratings and the corresponding mean square $MS = \sum d_{ij}^2/64$ of each block of the difference frame (Table 2). The highest priority is assigned to blocks with $MS > 650$ and the lowest priority to those with $MS < 13$. The rationale behind this is that if a block with a highest priority is missing, then the rebuilt block based on the previous M-encoded block will get a

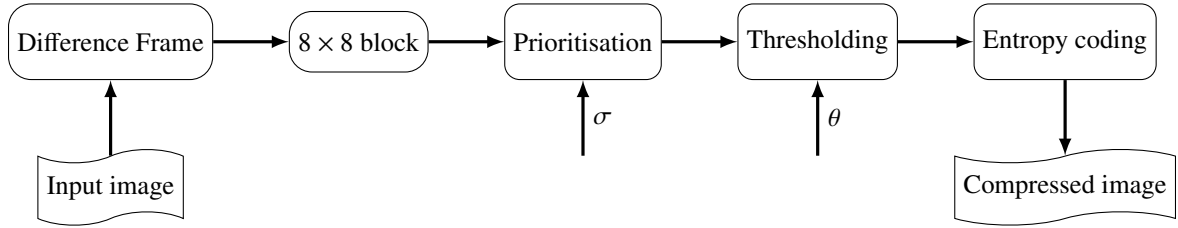


Figure 5: S-frame encoding sequence

Table 2: S-frame blocks priorities

MS	PSNR	Priority
< 13	> 37	4
]13, 51]]31, 37]	3
]51, 205]]25, 31]	2
]205, 650]]20, 25]	1
> 650	< 20	0

bad quality ($PSNR < 20$ dB). In contrast, loss of blocks with lowest priority does not affect the quality of the rebuilt block that still get an excellent quality ($PSNR > 37$ dB).

To save video sensors and network resources, the user can decide to only encode a subset of the blocks of a given frame by adjusting parameter σ that defines the lowest priority level to consider for the following step (thresholding). In order to be more suitable to WSN, σ default value is set to 0 where only blocks with the highest level of priority are retained. As a result the required energy will be drastically reduced. However, if σ is set to its maximum value 4, then all difference frame blocks are encoded. This may consume excessive energy and can result in a low compression ratio that does not justify the inter-coding choice for a given frame. The retained blocks by the prioritisation step are then thresholded using the *threshold similarity* parameter (θ). All pixels with value less than θ are zeroed. As a result, if $\theta = 0$ and $\sigma = 4$, then the S-frame is losslessly encoded. When θ is increased, the S-frame compression becomes more lossy. Default value of θ is 1.

As for the main frame, the last step consists in applying a lossless entropy encoding on the remaining blocks. Since not all blocks are transmitted, the sequence of the first block in a packet has to be included. For subsequent ones, we encode the difference with respect to the previous one. In both cases, the resulting number is encoded using VLI (variable-length integer) with $4 + n$ bits. The first 4 bits give the required number (n) of bits to encode the number using biased representation. The block's data are encoded using exponential-Golomb as default entropy coder.

3.3. Video Decoding and Quality Evaluation

The sender trace file is used by the sender application to get the list of data packets to transmit. At the end of the simulation or real experimentation, a receiver trace file (*rt-packet*) that records the list of the received packets has to be produced by the receiver application. Potential missing lines with respect to the sender trace correspond to the set of lost packets during transmission. Based on the different trace files in addition to the original captured video, the received frames are decoded considering the potential losses. Statistics on the conducted experiment are generated and recorded in the receiver frame trace file (*rt-frame*). Both network and video quality related metrics are computed in order to quantitatively assess the performances of the user's transmission strategies.

Network-related (QoS) metrics such as losses, delay and jitter can easily be computed using scripts based on the generated trace files. Video-related (QoE) metrics are implemented by the quality evaluation module. PSNR and SSIM metrics are implemented to assess the quality of both the encoded video before transmission (reference PSNR or SSIM) and the received video with respect to the initial lossless encoded video. The PSNR between the sent and the received, possibly distorted video frame is computed using :

$$PSNR = 20 \log \frac{V_{peak}}{MSE}$$

where MSE is the mean square error which is the average of the square of the pixel differences of the two images and V_{peak} is the maximum possible pixel value. The SSIM metric is computed as follows :

$$SSIM = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \times \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \times \frac{\sigma_{xy} + C_2/2}{\sigma_x\sigma_y + C_2/2}$$

where x and y are two non negative image signals, μ_x , σ_x and μ_y , σ_y are the mean and standard deviation of x and y respectively. σ_{xy} is the sample cross-covariance between x and y . We used the following values for the two constant parameters : $C_1 = 6.5025$ and $C_2 = 58.5225$.

3.3.1. Trace Files

The SenseVid encoder module generates a frame trace file (*st-frame*) that records information on how each captured frame is encoded :

1. frame number ;
2. frame type : M or S ;
3. frame size in bytes ;
4. the reference PSNR (dB) ;
5. the reference SSIM ;
6. the achieved compression ratio in bits per pixel (bpp) ;
7. the size (bits) of the different priority levels of the frame if M-encoded (up to 13 levels) ;
8. the estimated energy (mJ) required to capture this frame ;
9. the estimated energy (mJ) to encode this frame.

Table 3 gives a sample extract of the *st-frame* generated for a video clip where the first frame is M-encoded with size 3315 bytes resulting in 1.046 bpp compression and a reference PSNR of 33.889 dB. The seventh column gives the size of the four priority levels of this M-frame. The second frame is an S-frame compressed at 0.097 bpp with a reference PSNR of 31.937 dB. The two last columns give an estimation of the required energy to capture and encode the two frames assuming an 8-bit Atmel AVR micro-controller.

Table 3: *st-frame* sample with $\lambda = 4$, $\sigma = 3$, $\gamma = 21$ and $FFC = 1$ fps

Frame	Type	Size	refPSNR	refSSIM	bpp	Priority levels (M)	Capture energy	Encoding energy
1	M	3315	33.889	0.942	1.046	8138 4557 4295 9525	1.049	44.314
2	S	307	31.937	0.892	0.097	- - - -	1.049	0.266
...								

The packet trace file (*st-packet*) is produced by the packetiser module and gives information on the list of data packets to send by the sender application. The packetisation strategy is defined by the user by providing the maximum payload size that considers the wireless network characteristics as well as the application requirements. Each line of the sender trace file gives for each packet :

1. the transmission time ;
2. the sequence number ;
3. the payload size ;
4. the number of the frame to which belong the data of this packet ;
5. the frame type (M or S) concatenated to the number of the previous M-frame if S ;
6. the priority level of this packet ;

7. the list of the frame blocks carried by this packet.

Table 4 gives a sample *st-packet* for a payload size of 128 Bytes. Data of the same priority levels from different blocks are gathered in the same packet without exceeding the maximum payload size specified by the user. That is the second packet for instance has the highest priority level with data from blocks 0 to 39. The first packet (sequence 29) of the S-frame with priority 3 has a payload size of 118 bytes and contains data from blocks 0, 4, 5, 6, 26, 48, 66, 70 and 74.

Table 4: st-packet sample. Max packet payload size = 128 Bytes.

Time	Sequence	Size	Frame	Frame type	Priority	Blocks
0	1	127	1	M	3	0 28
0	2	125	1	M	0	0 39
0	3	119	1	M	3	29 74
0	4	128	1	M	2	0 77
0	5	128	1	M	1	0 79
...						
1	29	118	2	S1	3	0 4-6 26 48 66 70 74
1	30	108	2	S1	3	92 96 105-106 110
...						
1	36	127	2	S1	0	226 247-248 268 270
...						

The receiver frame trace file (*rt-frame*), records information on each received frame :

1. frame number ;
2. frame type : M or S ;
3. achieved PSNR (dB) ;
4. achieved SSIM.

Table 5 gives an *rt-frame* sample that corresponds to the transmission of the *st-packet* of Table 4 with 10% loss rate. The built first frame, for instance, has a PSNR of 30.225 dB to be compared with the reference PSNR (*st-frame*) of 33.889 dB.

Table 5: rt-frame sample

FrameNb	frameType	PSNR	SSIM
1	M	30.225	0.714
2	S	29.795	0.712
...			

3.4. Energy Model

Neglecting the computational (sensing and processing) cost with respect to communication cost may be suitable for WSNs. This does not hold in WWSNs where video acquisition and compression require significant computation resources that can not be neglected. As a result, an energy consumption model has to be provided in order to be able to assess the overall cost of video applications from video capture to its delivery to the final destination. Any power consumption model is dependent of the sensor node computation capabilities and instructions set. We give parameters for the Atmel Atmega128L, 8-bit micro-controller and allow the user to modify the parameters of the provided model to meet his target platform characteristics.

SenseVid gives separately an estimation of the required energy to capture and to encode each frame. Capturing energy depends mainly on the frame size. As reported by Cyclops's measures for instance, the frame size determines the size of sensor array and the subsequent number of necessary operations to convert each pixel to a digital value [1]. Currently, we adopted a simple model where the user can provide the required energy to capture an 8×8 block, then SenseVid computes overall capturing energy based on the frame size. The compression energy is estimated depending on how a frame is encoded. The compression cost of an M-encoded frame is estimated based on the required power to M-encode one block $P_{M-block}$ given by :

$$P_{M-block} = P_{DCT} + P_{Quantisation} + P_{entropy}$$

where P_{DCT} , $P_{Quantisation}$ and $P_{entropy}$ are amounts of power consumed by respectively DCT coefficients computation, quantisation and entropy coding of one block. P_{DCT} can be computed using :

$$P_{DCT} = \sum_{op} N_{op} * P_{op}$$

where P_{op} is power consumption of operation op (addition, multiplication or shift) that is mainly function of the number of required cycles to execute op , the underlying processor power and clock rate. N_{op} is the required number of operations of type op to compute the DCT. This number depends on the selected DCT as well as its zone size (ρ) and is given by :

$$N_{op}^{\rho} = \begin{cases} (\rho + 8) \times n_{op,\rho} & \text{if square DCT} \\ \rho \times n_{op,\rho} + \sum_{i=1}^{\rho} n_{op,i} & \text{if triangular DCT} \end{cases}$$

where $n_{op,i}$ is the number of operation of type op required by the 1-D DCT of side length i . Table 6 summarises the number of required operations for the implemented fast pruned LLM and BIN DCT. At this stage, it is worth noting that the cost of operation of type op is differentiated depending on integer (BIN DCT) or float (CLA and LLM) arithmetic. Per block quantisation is estimated as follows :

$$P_{Quantisation} = \begin{cases} 8^2 \times P_{div} & \text{classical DCT} \\ \rho^2 \times P_{div} & \text{square pruned DCT of side length } \rho \\ \rho \times (\rho - 1) \times P_{div} & \text{triangular pruned DCT of side length } \rho \end{cases}$$

where P_{div} is power consumption due to division operation. Finally, the entropy power is estimated depending on the chosen entropy coder. The default one is exponential-Golomb where the required power to encode a non null value can be given by :

$$P_{EG} = \begin{cases} 9 & \text{if CLZ instruction is available} \\ 3 \times \lfloor \log_2(\text{value} + 1) \rfloor + 7 & \text{otherwise} \end{cases}$$

where CLZ is the *Count Leading Zeros* instruction.

The required power to encode an S-frame is computed using :

$$P_{S-frame} = P_{diff} \times \text{blocksNb} + P_{priority} \times \text{nonNullblocksNb} + P_{thresh} \times \text{finalBlocksNb} + P_{entropy}$$

where blocksNb , nonNullblocksNb and finalBlocksNb are the number of the frame blocks, the non null blocks in the difference frame and the number of the retained blocks after thresholding respectively. $P_{diff} = 8^2 \times P_{sub}$ is the power consumed by the process of subtracting the main frame. $P_{priority} = 8^2 \times (P_{mul} + P_{add})$ is the consumed power by the mean square computation in the prioritisation step. $P_{thresh} = (P_{test} + P_{assignment}) \times 8^2$ is the required power by the thresholding step. Finally for the default entropy coding, $P_{entropy} = \text{numberOfNonNullValues} \times P_{EG}$.

Table 6: Fast pruned DCT cost for different values of ρ

ρ	LLM						binDCT-C					
	1-D DCT		sq. 2-D DCT		tr. 2-D DCT		1-D DCT		sq. 2-D DCT		tr. 2-D DCT	
	Add.	Mul.	Add.	Mul.	Add.	Mul.	Add.	Shift	Add.	Shift	Add.	Shift
8	29	11	464	176	413	152	30	13	480	208	420	172
7	28	11	420	165	348	130	28	12	420	180	346	139
6	26	10	364	140	280	102	28	12	392	168	290	115
5	25	9	325	117	223	77	28	12	364	156	234	91
4	24	9	288	108	169	59	27	11	324	132	174	63
3	23	8	253	88	118	38	19	6	209	66	96	26
2	20	6	200	60	66	18	13	2	130	20	46	6

4. Numerical Results

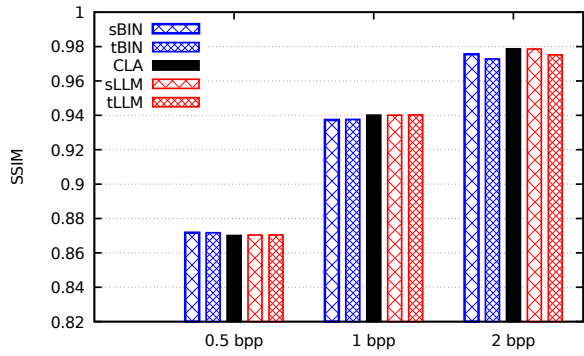
In this section, we are interested in assessing the effect of some parameters on the resulting compressed video sequence without considering its transmission. We consider two QCIF video clips namely the *hall* and the *flowers* video sequences [47] composed respectively of 300 and 250 frames captured at 25 fps. The *hall* video can be qualified as a low movement sequence with a static background while the *flowers* video is a medium motion sequence with many small details.

We begin by comparing the implemented DCTs. The two video sequences are considered without any preprocessing where the original FFC i.e. 25 fps and the original QCIF resolution are maintained. The quality coefficient (QF) is adjusted such that bit rates of 0.5, 1 and 2 bpp are obtained for the five implemented DCTs : the traditional one (CLA), the LLM and BIN using square and triangular patterns (sLLM, tLLM, sBIN and tBIN). The side length ρ is set to 8 and all frames are intra-coded. Figures 6a-6d depict the obtained video quality (reference SSIM/PSNR) for *hall* and *flowers* video clips. We observe that for both videos and for any chosen DCT, the lower the compression, the higher is the video quality. For *hall* video with CLA DCT, for instance, the fair quality (SSIM=0.8701 and PSNR=28.28 dB) obtained with 0.5 bpp is risen to excellent quality (SSIM=0.97871 and PSNR=39.07 dB). Compared to *hall*, *flowers* video exhibits low quality for the same compression ratio. This is due to the fact that the former has a static background with a small moving ROI while the latter has many small details and hence a low spatial redundancy.

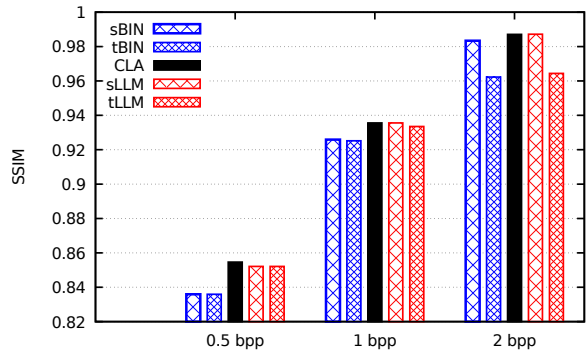
When comparing the three DCTs, we note that the BIN DCT produces slightly lower quality mainly for the lowest bit rate in the *flowers* video clip. This is widely offset by the corresponding energy footprint as shown by Figures 6e-6f. The BIN DCT exhibits the lowest energy expenditure since integer arithmetic requires less processor cycles. As expected, the CLA DCT consumes the largest amount of energy. The triangular pattern in both BIN and LLM results in lower energy cost (about 80% cost reduction) compared to the square pattern. The BIN DCT consumes around 1 mJ while CLA consumes more than 43 mJ for both *flowers* and *hall* video clips.

As it presents good image quality while consuming much lower energy compared to CLA and LLM, we only consider the BIN DCT in what follows. The DCT zone size ρ is varied to assess its effect on the video quality as well as the corresponding consumed energy. Both square and triangular shapes for the BIN DCT are considered. Figure 7 brings the obtained results when encoding the *hall* video sequence as the DCT zone size (ρ) varies from 3 to 8. The quality coefficient is adjusted to obtain bit rates of 0.5 and 1 bpp. Plot 7a shows that the required energy to encode a frame increases with the DCT zone size as the encoding complexity increases with the number of DCT coefficients to compute. For the same reason, the triangular pattern reduces the amount of consumed energy compared to the square one.

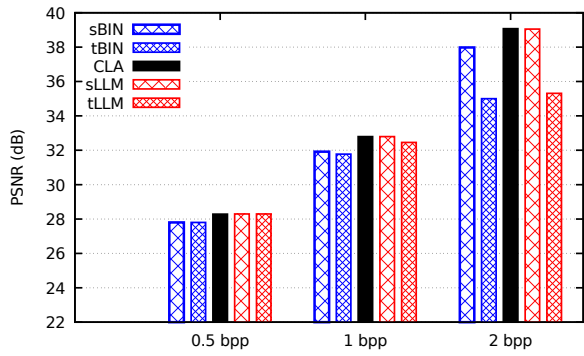
Figure 7b depicts the obtained SSIM and shows that beyond a given value of ρ the video quality does not improve mainly for the lowest bit rate. To obtain a bit rate of 0.5 bpp, considering the whole block ($\rho = 8$) in the square DCT which consumes about 43% additional energy does not allow any improvement with respect to $\rho = 5$. Similarly, triangular DCT achieves the same video quality as the square DCT beyond a given zone size. In fact, when increasing the triangle side size, all non null values are likely to be located at the left top sub-block. The square pattern does not add any significant DCT coefficient. Despite this, the square pattern consumes more energy. With 0.5 bpp



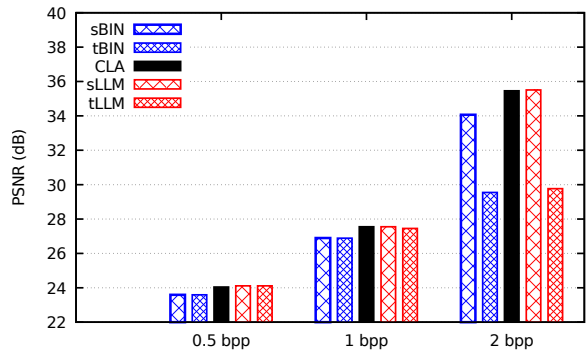
(a) Hall - SSIM



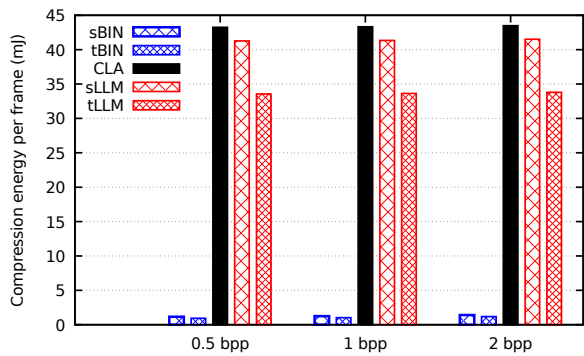
(b) Flowers - SSIM



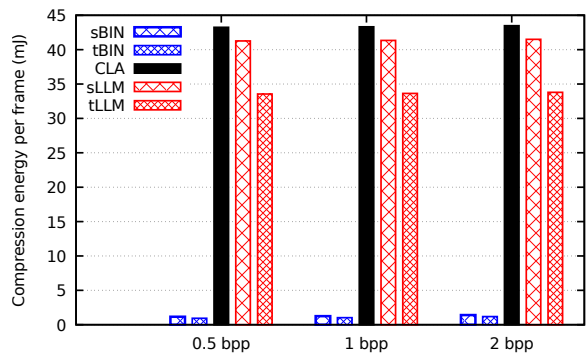
(c) Hall - PSNR



(d) Flowers - PSNR



(e) Hall - Encoding energy



(f) Flowers - Encoding energy

Figure 6: Comparing DCTs.

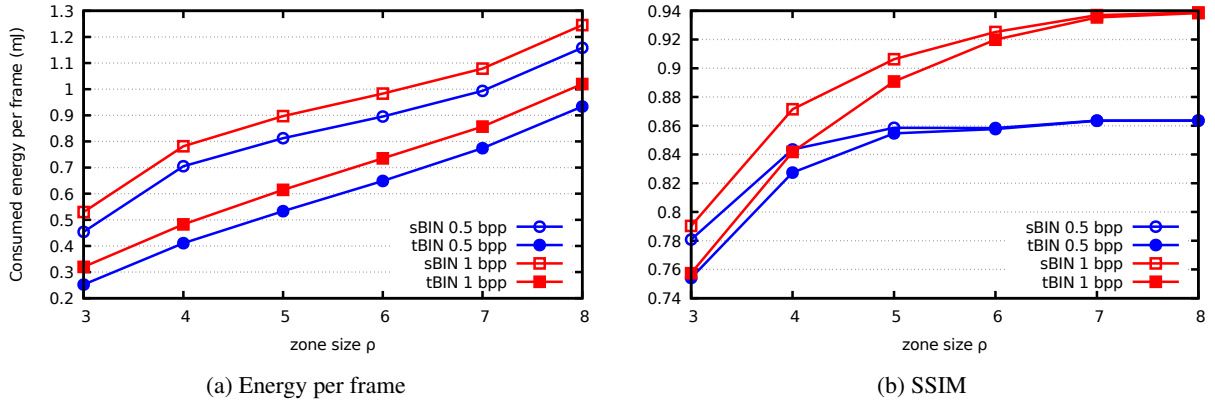


Figure 7: Varying the DCT zone size (ρ).

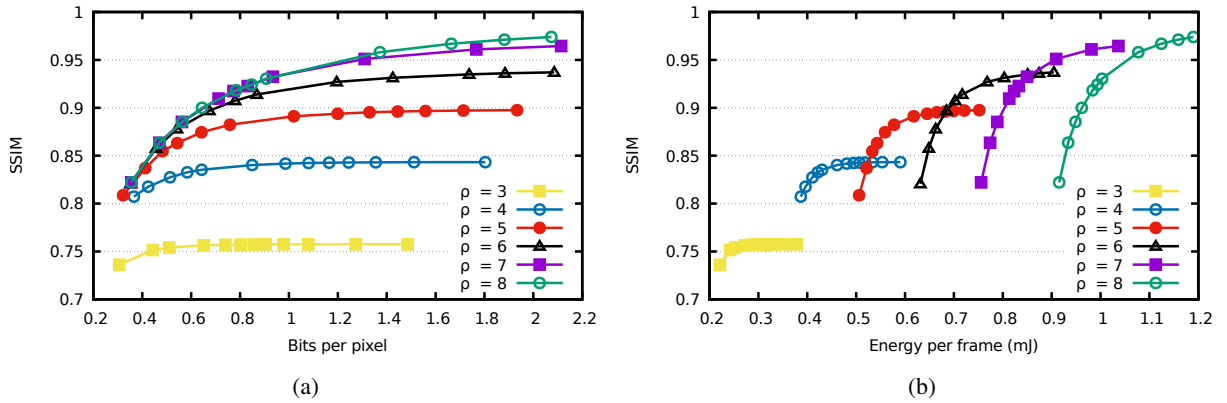


Figure 8: Rate distortion - Triangular BIN

compression, a square DCT with $\rho = 5$ consumes more than 50% of additional energy than the triangular DCT.

Figure 8a plots the video quality as a function of the obtained bit rate for the triangular BIN. For each zone size, the quality coefficient is varied to obtain different bit rates ranging from 0.3 to 2 bpp. The main observation is that, increasing the bit rate does not improve significantly the video quality mainly for small zone sizes. With a side size of 4, we can obtain a fair quality with 1 bpp that does not improve with higher bit rates. A good video quality (SSIM > 0.9) can be obtained by setting $\rho \geq 6$ and at least a bit rate of 1 bpp.

Figure 8b plots the SSIM as a function of the consumed energy. A higher energy corresponds to a lower compression ratio (higher bit rate). We can see that additional energy can be consumed to obtain less compressed video without improving the video quality. Thus, depending on the chosen side size, the quality coefficient has to be appropriately chosen to obtain the maximum video quality as well as a good compression ratio while consuming less energy. Our results show that the quality coefficient has to be set to lower values for lower ρ values and can be increased to 50 or more when $\rho = 8$.

Now, we consider inter-frame coding and assess the effect of parameters γ , θ and σ on the obtained S-frames as well as on the whole video. Figures 9a-11b depict using boxes, the distribution of obtained SSIM, bit rate and encoding energy by the 300 frames of the hall video sequence. The central rectangle of each box spans the first quartile to the third quartile (the interquartile range). A segment and a small black square inside each rectangle shows the median and the mean values respectively. The whiskers above and below the box show the minimum and maximum achieved values. We set γ to 6, 12 and 18 which results in respectively 156, 254 and 289 S-frames. θ is set to 1, 10 and 100 to

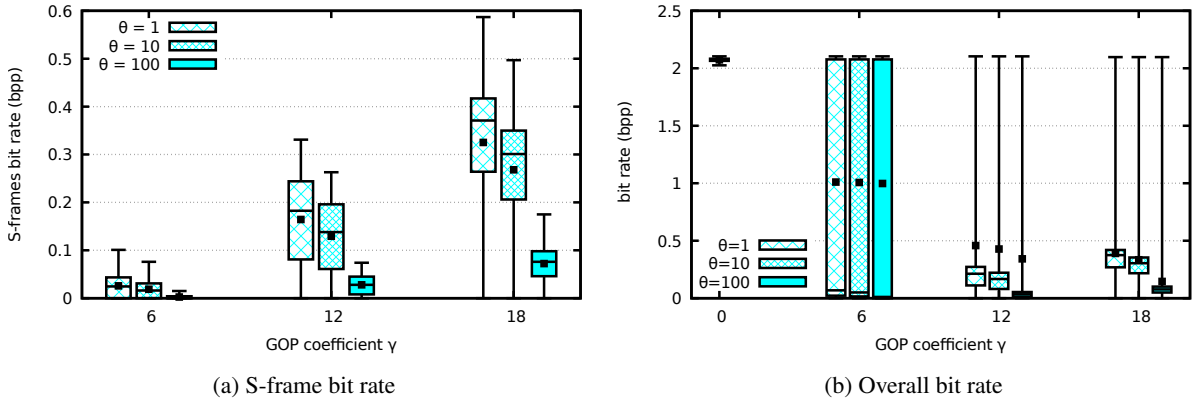


Figure 9: Inter-frame coding : impact of θ and γ on the bit rate

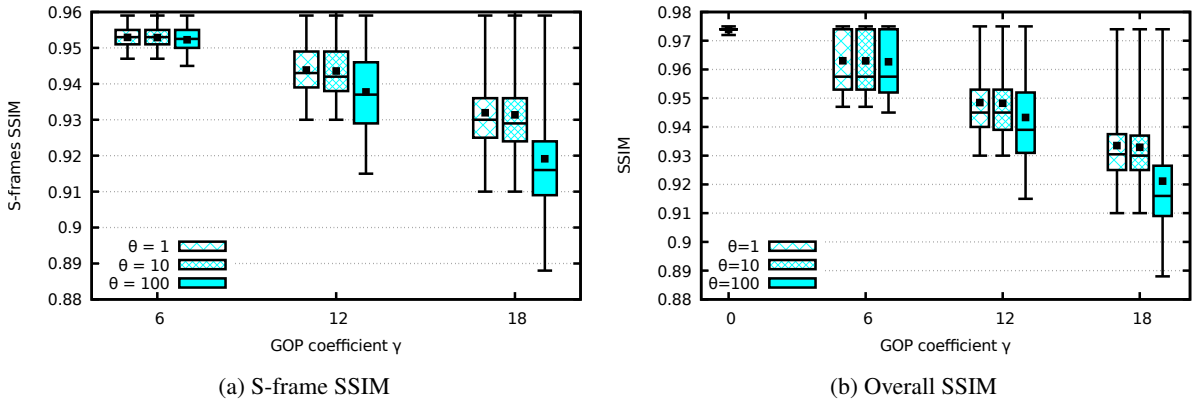


Figure 10: Inter-frame coding : impact of θ and γ on the video quality

assess the effect of thresholding on the S-frame encoding. σ is set to 0 its default value, which means that only the highest priority data are encoded. Figures 9a, 10a and 11a depict the results for only S-frames while Figures 9b, 10b and 11b depict the overall achieved values.

As shown by Figure 9b, when all frames are intra-coded ($\gamma = 0$), we obtain a bit rate of 2.07 bpp. When γ is set to a non zero value, the bit rate is reduced. For $\theta = 1$, we achieve a bit rate of 1, 0.46 and 0.39 bpp when γ is set to 6, 12 and 18 respectively. The corresponding quality (Figure 10b) does not significantly decrease since the SSIM is still bigger than 0.93 in the worst case ($\gamma = 18$). When considering only the S-frames, the bit rate increases with γ as shown by Figure 9a. This is due to the fact that a higher value of γ produces more S-frames which exacerbates their difference with the related M-frame. We note that some S-frames have null size and that their ratio increases when γ is set to lower values. S-frames are more likely to be very similar to the previous M-frame which increases the number of null values in the difference frame. The thresholding process further increases the number of null frames. Note that an M-frame has a better quality compared to the following S-frames mainly when γ and θ are set to high values.

When $\gamma = 0$, all frames are intra-coded and requires in average 1.19 mJ to encode each frame as depicted by Figure 11b. Energy saving is possible if a subset of frames are inter-coded. Overall, increasing γ allows to reduce the per-frame encoding energy. Compression energy is halved when $\gamma = 18$ and $\theta = 1$. When only S-frames are considered, the mean energy increases since more (non null) values have to be encoded due to the increasing number of S-frames. Thresholding allows to further decrease the consumed energy to encode the S-frames since it generates zero values in the difference frame allowing the lossless encoder to achieve higher compression ratios.

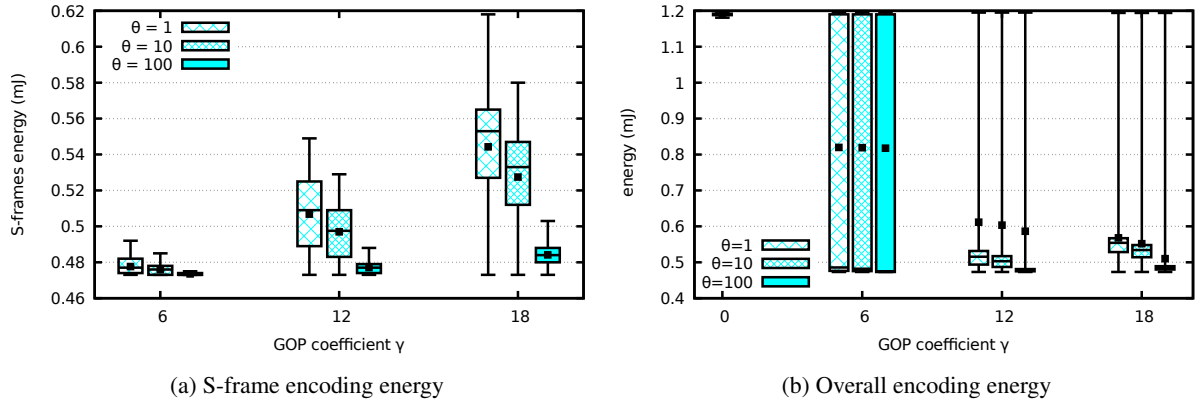


Figure 11: Inter-frame coding : impact of θ and γ on energy

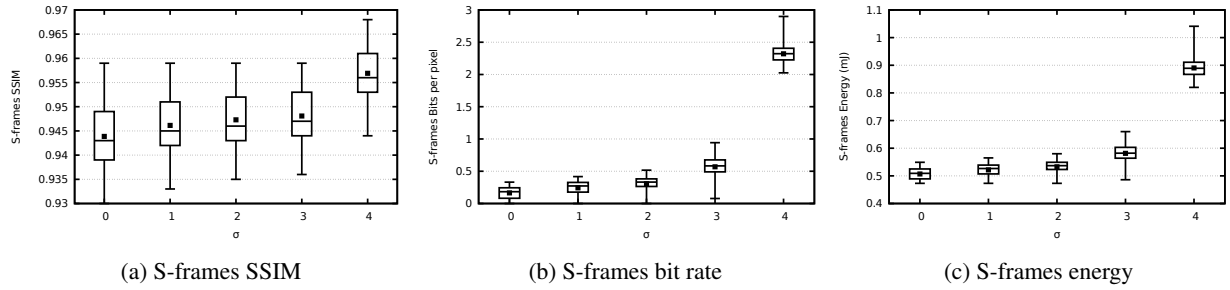


Figure 12: Inter-frame coding : impact of σ

Figure 12 plots the obtained results for S-frames when the σ parameter is varied. The highest the number of blocks we consider, the highest are the bit rate and the encoding energy. When all blocks ($\sigma = 4$) are encoded, the mean achieved SSIM is greater than 0.955 at the expense of a bit rate greater than 2 bpp and an encoding energy of approximately 0.9 mJ per frame. If only the highest priority blocks are considered ($\sigma = 0$), the mean achieved SSIM is greater than 0.94 resulting in a good quality video with less than 0.45 bpp average bit rate and about 0.5 mJ encoding energy.

5. Sample Experimentations

SenseVid traffic trace files can be used in any real testbed or simulated environment. The user can use or write application modules that make use of the trace files. The sender sends data packets as specified by the sender trace file (st-packet). The receiver records the list of received packets in a receiver trace file (rt-packet). As a first step, we provide Contiki [20] application modules that allow the use of SenseVid trace files in a real testbed, namely IoT-LAB [22] as well as in Cooja, a simulation environment. The purpose of this section is to give use examples of SenseVid rather than optimising the transmission process which is beyond the scope of this paper.

5.1. IoT-LAB Experiment

IoT-LAB provides large scale WSN testbed with over 2000 wireless sensor nodes with different processor architectures and different wireless chips. IoT-LAB can be accessed through a web portal or using provided command-line tools. Additionally, it offers a hosted environment on SSH front-ends with pre-installed command-line Tools and target architectures cross-compiler tool chains. It allows to retrieve experiments results and to access devices serial ports. The main benefit of SenseVid in a real testbed such as IoT-LAB is that it allows to perform real experiment without

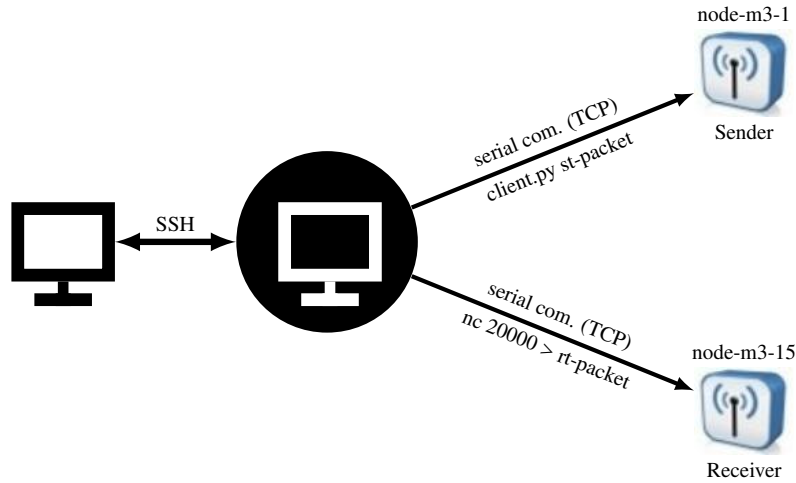


Figure 13: IoT-LAB sample configuration with SenseVid.

having a camera sensor. An application module that allows the transmission of packets based on a traffic trace file is developed. Based on the packet traffic trace, commands to send packets are injected into the sensor platform (sender node) using a python script (`client.py`) as shown by Figure 13. The receiver reports the sequence numbers of correctly received packets that can be retrieved using the netcat (`nc`) command to produce the receiver trace. This latter can be analysed using SenseVid on the user’s computer to rebuild the received video and compute metrics on the performed experiment.

We report in this section on our first experiments on top of IoT-LAB testbed. M3 open nodes (ARM Cortex M3, 32-bits MCU and 802.15.4 PHY standard) are used to emulate the transmission of the hall video clip. We used nodes m3-1, m3-10, m3-15, m3-16 and m3-24. Sender and receiver pairs are chosen so different loss rates are obtained. Average success rates obtained for pairs (m3-1 → m3-16), (m3-1 → m3-15) and (m3-24 → m3-10) are respectively of about 98%, 91% and 33%. The hall video is considered with an FFC of 1 fps giving 12 captured frames. The GOP coefficient γ is set to 22 which results in 4 M-frames and 8 S-frames with sequence “MSSSSSSMSMSM” where the i^{th} character gives the type of the i^{th} frame. The quality coefficient is set to obtain bit rates of 0.5 and 1 bpp. S-frames thresholding parameter is set to 10. Remaining parameters are left with their default values.

Figure 14 plots the achieved SSIM by the different frames of the hall video sequence as well as the reference SSIM that corresponds to the quality of the frames before transmission. Connection m3-24 → m3-10 exhibits more losses than the other two connections resulting in the lowest video quality. For a bit rate of 0.5 bpp, node m3-10 receives a video with an average SSIM = 0.2554 while the reference SSIM is greater than 0.9. Connection between node m3-1 and m3-16 provides the highest success ratio (98%) and thus achieves the highest video quality that is almost the same as the reference SSIM. Figure 15 shows frames 8, 9 and 10 before their transmission (first column) then when received by nodes m3-15, m3-16 and m3-10 respectively. We can easily observe the good quality of received frames by node m3-16. Because of the high experienced loss ratio, node m3-10 receives bad quality frames.

5.2. Cooja Sample Experiment

For the purpose of showing the use of SenseVid in a simulated environment, we chose Cooja [21], the Contiki network simulator. The RIME collect application is modified to allow the use of traffic trace file as well as retrieving the list of received packets. In this experiment, we aim to evaluate the effect of sending different priority levels of an M-frame. Our simulation parameters are summarised in Table 7. Figure 16 plots the achieved quality by the hall video frames when a given number of priority levels of an M-frame are transmitted. Figure 16a shows that when increasing the maximum priority level to transmit, the quality of each frame is improved. However when it is set to 5 or 6, the quality deteriorates as shown by Figure 16b. This is due to the fact that when increasing the amount of data to transmit, the transmission rate has to be risen to maintain real time transmission. This results in more lost packets.

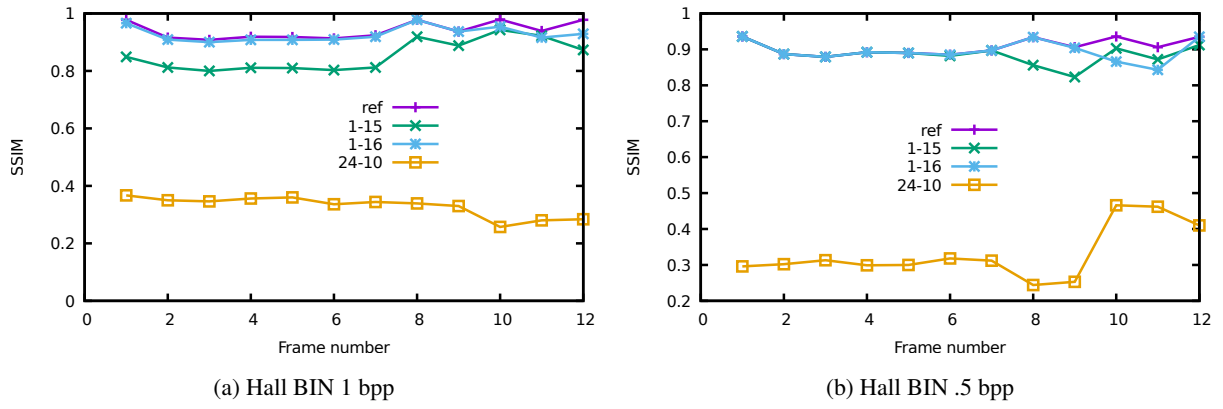


Figure 14: IoT-LAB experiment : Frames quality

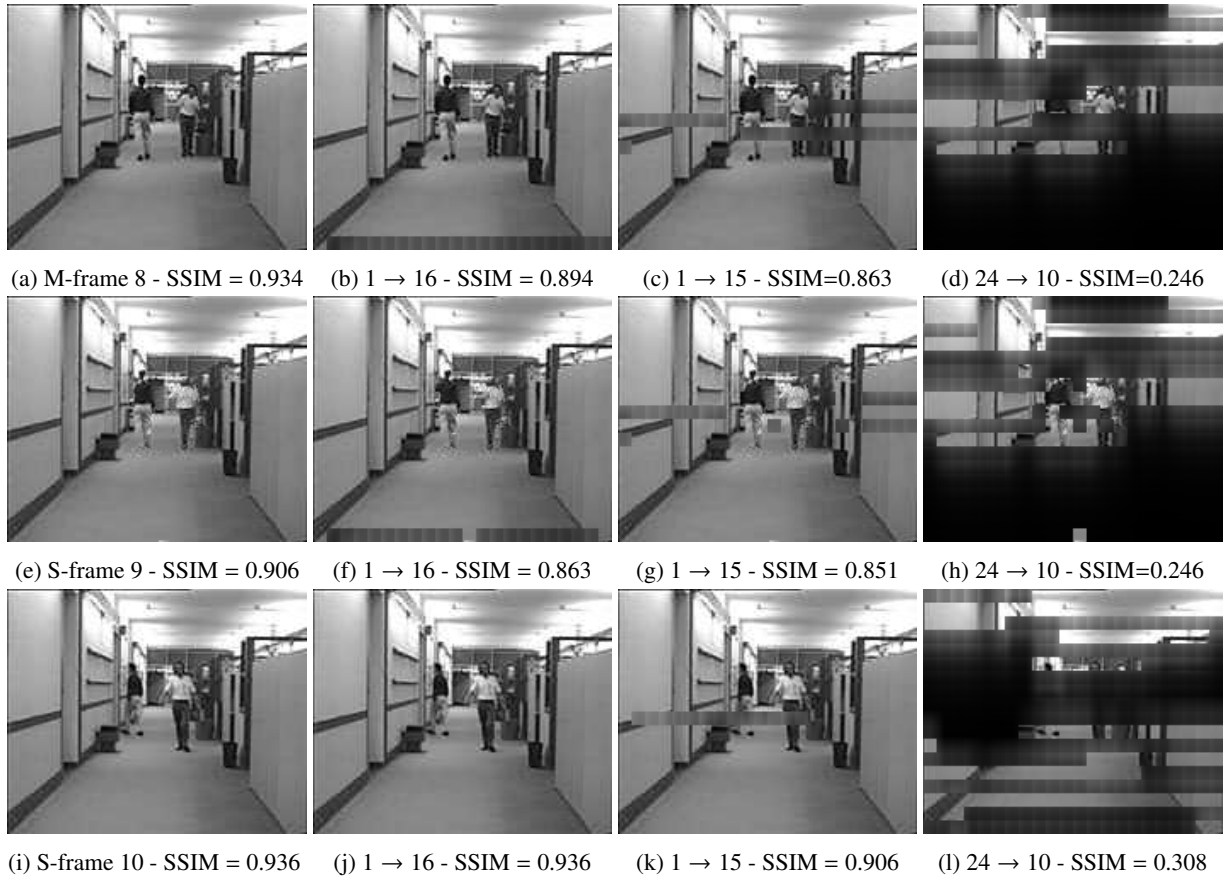


Figure 15: Sent (reference) and received frames in the three IoT-LAB experiments

Table 7: IoT-LAB Experiments Settings

Video	hall
Packet payload size	96
DCT	tBIN with side size $\rho = 8$
FFC	1 FPS : 12 frames are captured.
Resolution	half QCIF (88×72)
GOP coefficient	$\gamma = 15$
Resulting encoding sequence	MMMMMSMMSMSS
M-frames levels number	$\lambda = 0..6$
S-frame threshold	$\theta = 20$
S-frame maximum level	$\sigma = 4$

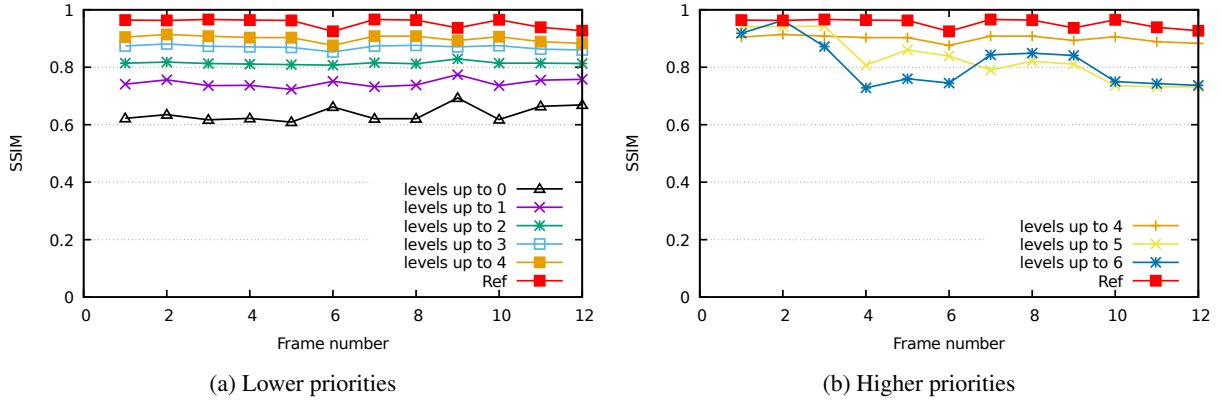


Figure 16: SSIM Hall - Cooja

When sending up to level 4 data, there is no loss. The sink experiences 16% and 23% of losses when sending data up to priority level 5 and 6 respectively. This means that the amount of data to transmit can be adapted to network dynamics through priority levels tuning. Figures 17 and 18 show frames 5 (M-encoded) and 6 (S-encoded) respectively as they would be rebuilt by the sink with their respective quality. The first image of each figure corresponds to the sent frame. We can see and confirm that the best quality is obtained when priority levels up to the fourth are transmitted by the sender node.

6. Conclusion

In this paper, we presented SenseVid, a traffic trace based transmission and evaluation tool targeted to WSN. In addition to intra-frame coding, SenseVid implements inter-frame compression to handle video transmission. To adapt to WSNs constrained resources, it adopts low-cost compression techniques where intra-frame coding is performed using fast pruned DCTs. Inter-frame compression allows to further reduce encoding energy by considering only the most important blocks. A configurable fine-grain energy model is provided where both video capture and encoding cost are accounted for on a per frame basis. Video traffic differentiation based on priority levels is also provided. SenseVid is based on the use of traffic traces derived from the video sequence to send instead of the corresponding bit stream. This allows its use in any real testbed or simulated environment. The received video is rebuilt considering potential losses and both QoS and QoE metrics are computed. Application modules that make use of SenseVid trace files are available in Contiki OS. Sample experimentations were provided using a real testbed (IoT-LAB) as well as a simulated environment (Cooja).

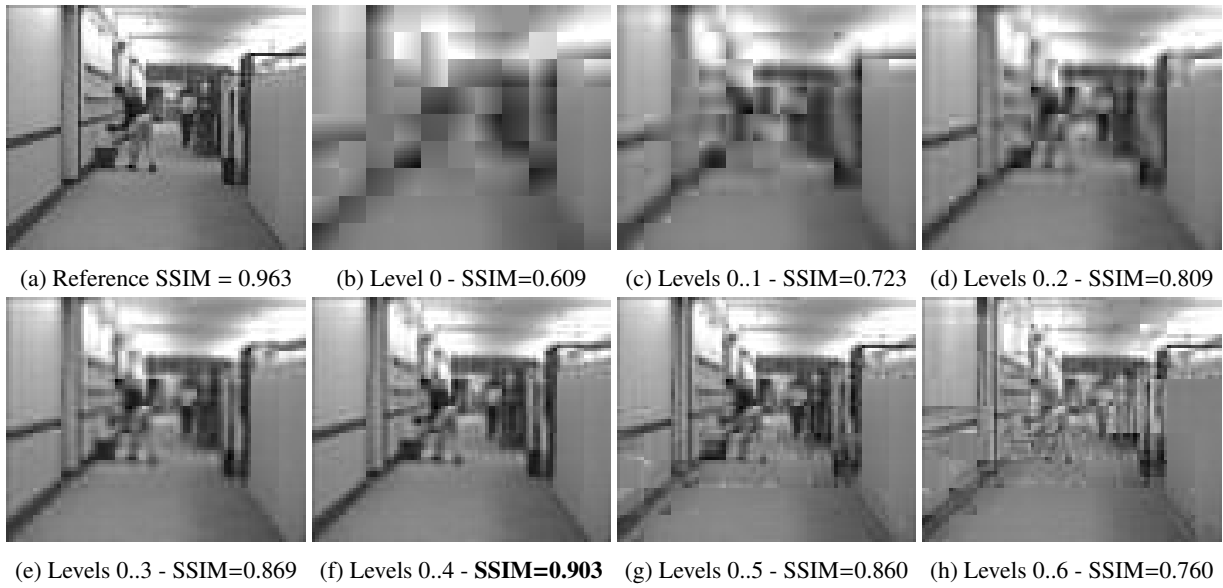


Figure 17: Cooja frame 5 : M encoded

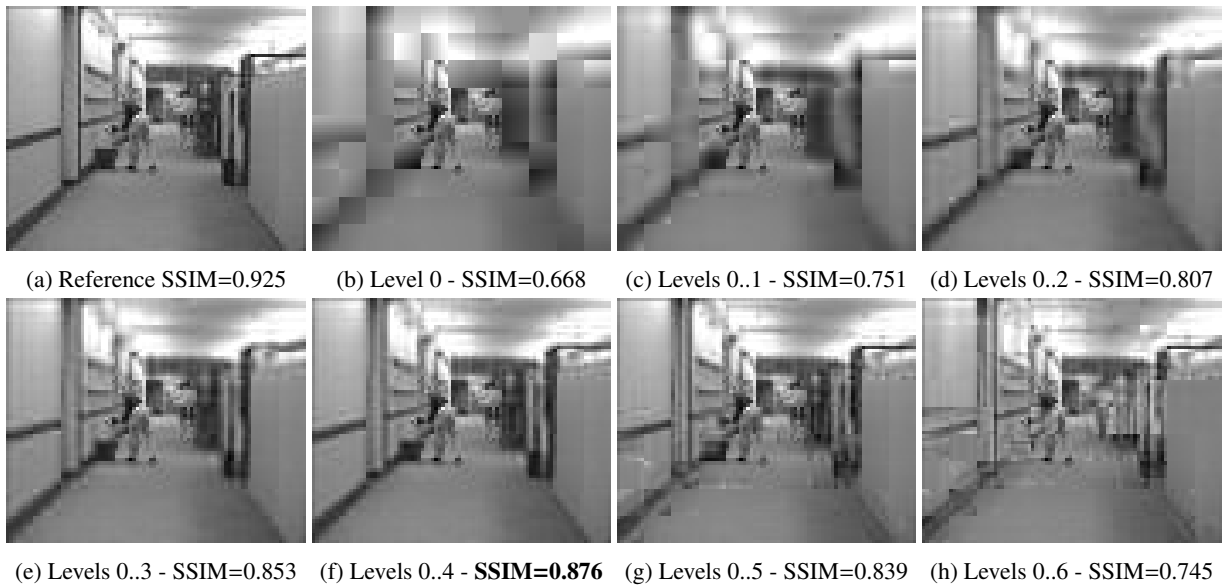


Figure 18: Cooja frame 6 : S encoded

In the future, we expect to implement other encoding techniques. As the tool is free open source, users can also implement their own compression techniques. Other application modules are also planned to be developed mainly for ns2, ns3 and Castalia simulators. In this latter, SenseVid can be combined to WiSE-MNet++ to benefit from its implemented video distributed algorithms. Finally, the energy model has to be refined and enriched to consider other sensor node types by considering their real energy consumption data.

Bibliography

- [1] M. Rahimi, R. Baer, O. Iroezji, J. Garcia, J. Warrior, D. Estrin, M. Srivastava, Cyclops: in situ image sensing and interpretation in wireless sensor networks, in: Proc. of the ACM Conf. on Embedded Networked Sensor Systems (SenSys), San Diego, CA, 2005.
- [2] A. Kerhet, M. Magno, F. Leonardi, A. Boni, L. Benini, A low-power wireless video sensor node for distributed object detection, *Journal of Real-Time Image Processing* 2 (4) (2007) 331–342.
- [3] K. R. Anupama, C. S. S. Reddy, M. V. Shenoy, FlexEye – a flexible camera mote for sensor networks, 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN) (2015) 1010–1015.
- [4] [Network simulator 2](http://www.isi.edu/nsnam/ns).
URL <http://www.isi.edu/nsnam/ns>
- [5] [Omnet++, discrete event simulator](https://omnetpp.org/).
URL <https://omnetpp.org/>
- [6] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, *Image Processing, IEEE Transactions on* 13 (4) (2004) 600–612. doi:10.1109/TIP.2003.819861.
- [7] J. Klaue, B. Rathke, A. Wolisz, Evalvid - a framework for video transmission and quality evaluation, in: In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation, 2003, pp. 255–272, <http://www.tkn.tu-berlin.de/research/evalvid/>.
- [8] P. Seeling, M. Reisslein, B. Kulapala, Network performance evaluation using frame size and quality traces of single-layer and two-layer video: A tutorial, *Communications Surveys Tutorials, IEEE* 6 (3) (2004) 58–78. doi:10.1109/COMST.2004.5342293.
- [9] D. Rosário, Z. Zhao, C. Silva, E. Cerqueira, T. Braun, An omnet++ framework to evaluate video transmission in mobile wireless multimedia sensor networks, in: Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 277–284.
- [10] C. Nastasi, A. Cavallaro, [Wise-mnet: an experimental environment for wireless multimedia sensor networks](http://www.eecs.qmul.ac.uk/~andrea/wise-mnet.html), in: Sensor Signal Processing for Defence (SSPD 2011), IET, 2011, pp. 1–5.
URL <http://www.eecs.qmul.ac.uk/~andrea/wise-mnet.html>
- [11] [Castalia network simulator](http://castalia.forge.nicta.com.au/), last accessed 2016-10-10 (2011).
URL castalia.forge.nicta.com.au/
- [12] I. F. Akyildiz, T. Melodia, K. R. Chowdhury, [A survey on wireless multimedia sensor networks](http://dx.doi.org/10.1016/j.comnet.2006.10.002), *Computer Networks* 51 (4) (2007) 921–960.
URL <http://dx.doi.org/10.1016/j.comnet.2006.10.002>
- [13] C. N. Taylor, D. Panigrahi, S. Dey, Design of an adaptive architecture for energy efficient wireless image communication, in: Embedded processor design challenges, Springer, 2002, pp. 260–273.
- [14] C. Pham, [videosense: a simulation model of image sensors under omnet++/castalia](http://cpham.perso.univ-pau.fr/WSN-MODEL/wvsn-castalia.html) (June 7th 2012).
URL <http://cpham.perso.univ-pau.fr/WSN-MODEL/wvsn-castalia.html>
- [15] L. Makkaoui, V. Lecuire, J. Moureaux, Fast zonal dct-based image compression for wireless camera sensor networks, in: Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on, 2010, pp. 126–129. doi:10.1109/IPTA.2010.5586798.
- [16] E. Orellana-Romero, J. SanMartin-Hernandez, C. Duran-Faundez, V. Lecuire, C. Aguilera, Sim-lit: A simulation framework for image quality assessment in wireless visual sensor networks under packet loss conditions, in: Computer Science Society (SCCC), 2011 30th International Conference of the Chilean, 2011, pp. 202–209. doi:10.1109/SCCC.2011.27.
- [17] P. Levis, N. Lee, M. Welsh, D. E. Culler, Tossim: accurate and scalable simulation of entire tinyos applications, in: SenSys, 2003, pp. 126–137.
- [18] Y. Baziz, M. Maimour, B. Kechar, Evalvsn : a new tool for video quality evaluation in wireless sensor networks, in: IEEE (Ed.), the 4th International Conference on Multimedia Computing and Systems, IEEE, Marrakesh, Morocco, 2014.
- [19] M. Maimour, SenseVid, <http://w3.cran.univ-lorraine.fr/perso/moufida.maimour/SenseVid/sensevid.html>.
- [20] A. Dunkels, B. Gronvall, T. Voigt, Contiki - a lightweight and flexible operating system for tiny networked sensors, in: Local Computer Networks, 2004. 29th Annual IEEE International Conference on, 2004, pp. 455–462.
- [21] I. Romdhani, M. Qasem, A. Y. Al-Dubai, B. Ghaleb, Cooja simulator manual, Tech. rep., Edinburgh Napier University (2016).
- [22] [Iot experimentation at a large scale](http://www.iot-lab.info).
URL www.iot-lab.info
- [23] C.-H. Ke, C.-K. Shieh, W.-S. Hwang, A. Ziviani, et al., An evaluation framework for more realistic simulations of mpeg video transmission., *J. Inf. Sci. Eng.* 24 (2) (2008) 425–440.
- [24] [Gercom, ns3+evalvid module](http://gercom.ufpa.br).
URL <http://gercom.ufpa.br>
- [25] A. Mammeri, B. Hadjou, A. Khoumsi, A survey of image compression algorithms for visual sensor networks, *ISRN Sensor Networks* 2012.
- [26] G. K. Wallace, The jpeg still picture compression standard, *Communications of the ACM* 34 (4) (1991) 30–44.
- [27] D.-U. Lee, H. Kim, S. Tu, M. Rahimi, D. Estrin, J. D. Villasenor, Energy-optimized image communication on resource-constrained sensor platforms, in: Proceedings of the 6th international conference on Information processing in sensor networks, IPSN '07, ACM, New York, NY, USA, 2007, pp. 216–225.

- [28] W.-H. Chen, C. Smith, S. Fralick, A fast computational algorithm for the discrete cosine transform, *IEEE Transactions on Communications* 25 (9) (1977) 1004–1009.
- [29] C. Loeffler, A. Ligtenberg, G. S. Moschytz, Practical fast 1-d dct algorithms with 11 multiplications, in: *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on, 1989*, pp. 988–991 vol.2.
- [30] On the multiplicative complexity of discrete cosine transforms, Vol. 38, 1992, pp. 1387–1391.
- [31] W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, *Applied and Computational Harmonic Analysis* 3 (2) (1996) 186 – 200.
- [32] J. Liang, T. D. Tran, A fast multiplierless approximations of the dct with the lifting scheme, *IEEE Transactions on Signal Processing* 49 (2) (2001) 3032–3044.
- [33] C. Chiasserini, E. Magli, Energy consumption and image quality in wireless video-surveillance networks, in: *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), 2002*, p. 23572361.
- [34] M. M. Magli, L. Merello, Low complexity video compression for wireless sensor networks, in: *Proceedings of 2003 International Conference on Multimedia and Expo, 2003*, p. 585588.
- [35] Z. Wang, Pruning the fast discrete cosine transform, *IEEE Transactions on Communications* 39 (5) (1991) 640–643. doi:10.1109/26.87153.
- [36] A. Mammeri, A. Khoumsi, D. Ziou, B. Hadjou, Energy-aware JPEG for visual sensor networks, in: *MCSEAI conference, Oran, Algeria, 2008*.
- [37] A. Mammeri, A. Khoumsi, D. Ziou, B. Hadjou, Modeling and adapting jpeg to the energy requirements of VSN, in: *Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on, 2008*, pp. 1–6. doi:10.1109/ICCCN.2008.ECP.151.
- [38] A. Mammeri, A. Khoumsi, D. Ziou, B. Hadjou, Energy-efficient transmission scheme of jpeg images over visual sensor networks, in: *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on, 2008*, pp. 639–647. doi:10.1109/LCN.2008.4664259.
- [39] C. Pham, A. Makhoul, Performance study of multiple cover-set strategies for mission-critical video surveillance with wireless video sensors, in: *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on, IEEE, 2010*, pp. 208–216.
- [40] F. Awad, E. Taqieddin, M. Mowafi, O. Banimelhem, A. AbuQdais, A simulation testbed to jointly exploit multiple image compression techniques for wireless multimedia sensor networks, in: *International Symposium on Wireless Communications Systems (ISWCS), Barcelona, 2014*, pp. 905–911.
- [41] M. Y. Mowafi, F. H. Awad, E. S. Taqieddin, O. Q. Banimelhem, A practical study of jointly exploiting multiple image compression techniques for wireless multimedia sensor networks., *JCM* 7 (4) (2012) 309–320.
- [42] A. Mascher-Kampfer, H. Stögner, A. Uhl, Comparison of compression algorithms' impact on fingerprint and face recognition accuracy, in: *Visual Communications and Image Processing, 2007*, pp. 650810–1.
- [43] J. C. SanMiguel, A. Cavallaro, Networked computer vision: the importance of a holistic simulator, *Computer* 50 (7) (2017) 35–43.
- [44] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, M. Schwamborn, Bonnmotion: a mobility scenario generation and analysis tool, in: *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010*, p. 51.
- [45] A. Rowe, C. Rosenberg, I. Nourbakhsh, A low cost embedded color vision system, in: *IEEE (Ed.), In International Conference on Intelligent Robots and Systems (IEEE/RSJ 2002), 2002*, pp. 208–213.
- [46] H. S. Aghdasi, M. Abbaspour, M. Moghadam, Samei, An energy-efficient and high-quality video transmission architecture in wireless video-based sensor networks, *Sensors* 8 (2008) 4529–4559.
- [47] *Yuv video sequences.*
URL <http://trace.eas.asu.edu/yuv/index.html>
- [48] Crossbow MICAz product datasheet, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf.
- [49] T. D. Tran, A fast multiplierless block transform for image and video compression, in: *Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, Vol. 3, 1999*, pp. 822–826 vol.3.
- [50] J. Teuhola, A compression method for clustered bit-vectors, *Information processing letters* 7 (6) (1978) 308–311.