EvalVSN: A new tool for video quality evaluation in Wireless Sensor Networks

Yacine Baziz, Moufida Maimour, Bouabdellah Kechar

EvalVSN : a New Tool for Video Quality Evaluation in Wireless Sensor Networks

Yacine Baziz
Oran Es-Sénia University
Facult des sciences - BP 1524
El M'Naouar, Oran, Algeria
bazizyacine@gmail.com

Moufida Maimour
Université de Lorraine, CRAN, UMR 7039,
Campus Sciences, BP 70239
Vandœuvre-lès-Nancy Cedex, 54506, France
moufida.maimour@univ-lorraine.fr

Bouabdellah Kechar
Oran Es-Sénia University
Facult des sciences - BP 1524
El M'Naouar, Oran, Algeria
bouabdellah.kechar@univ-oran.dz

Abstract—**Wireless video sensor networks (WVSN) are foreseen to be a key driver of the development of pervasive applications of every-day life. They trigger very challenging problems due to their constrained nature in terms of energy, bandwidth, processing and storage means. In order to be able to assess the performances of their proposals, WVSN's researchers community needs a transmission and evaluation tool that considers WSN specific characteristics. In this paper, we propose EvalVSN, a new tool for video transmission and evaluation targeted to WVSN. EvalVSN allows both real and simulation-based video transmission and evaluation in WSN. Currently application modules that make use of EvalVSN trace files are available in TinyOS, ns2 and Castalia environments.**

Keywords—Wireless Video Sensor Networks (WVSN); Video/Image Quality Evaluation ; EvalVSN ; PSNR ; SSIM.

I. INTRODUCTION

Recent technological advances have led to the emergence of small low-power devices that integrate sensors with on-board processing and wireless communication capabilities. Pervasive networks of such sensors open new vistas for a wide spectrum of applications. Nowadays, making use of Wireless Sensor Networks (WSN) is foreseen to be a crucial element to allow a major evolution in numerous applications. WSN are constrained networks since they have very limited resources in terms of energy, bandwidth, processing and storage means. This is why, WSN were traditionally targeted to relatively low rate event-driven applications where a limited amount of data is transferred from a sensor node to the *collect station* or the *sink*. This is the case for instance of applications concerned with temperature and humidity measures.

More recently, devices with low-cost and low-power visual modules [1], [2] have emerged allowing for fostering the development of Wireless Video Sensor Networks (WVSN). These latter provide significant benefit to many sensor networking applications such as surveillance, target tracking, environmental monitoring, and traffic management systems. However, WVSN generate unique challenging problems and should be designed to satisfy limited resources while providing a good quality of service (QoS). A significant effort has to be made by the research community to make WVSN applications a reality. To do so, a video transmission and evaluation tool targeted to WSN is required.

In WSN, a tool called Sim-LIT [3] is proposed but it is targeted to image instead of video transmission evaluation.

One popular tool used to evaluate video quality in computer networks is EvalVid [4]. However, the set of encoders such as MPEG-4, H.263 or H.264 supported by EvalVid are resources-hungry and as a result are not adapted to constrained networks such as WSN. In the last couple of years, a significant effort has been made to provide energy-aware encoding techniques that are more suitable to WSN. Work in [5] makes use of a modified version of MPEG-2 where video frames are assumed to be in gray-scale with the possibility to split each block data into priority levels. In [6], the effort is made in optimising the DCT (Discrete Cosine Transform) computation which is known to consume a significant amount of energy. The authors propose to use a fast zonal DCT which combine pruned DCT [7] with a fast DCT algorithm [8].

In this paper, we propose *EvalVSN* (publicly available at [9]), a new video transmission and evaluation tool that follows the same principle of EvalVid [4] while being specific to WSN. EvalVSN can help a network researcher to assess network protocols targeted to video applications in WSN. It implements a modified version of MPEG-2 similar to the one proposed in [5] with priority assignment but additionally provides different DCT implementations as the one proposed in [6]. Application modules that allow the use of EvalVSN with more realistic loss patterns are also provided in the TinyOS [10], ns2 [11] and Castalia [12] environments.

The remainder of this paper is organised as follows. Section II gives background preliminaries. The EvalVSN architecture and main functionalities are presented in Section III. An illustrative example of a possible use of EvalVSN is given in IV. Section V concludes the paper.

II. BACKGROUND PRELIMINARIES

A. M-MPEG Codec Overview

In this first version of EvalVSN, we implemented M-MPEG (Modified-MPEG) [5] based on MPEG-2 with some changes that will be reported in Section III-B. In M-MPEG, each video frame is supposed to be in gray-scale. Two types of video frames are considered Main-Frame (M-Frame) and Difference-Frame (D-Frame). M-Frames are compressed before transmission, using an extended JPEG method [13]. Compared with JPEG, the extended JPEG method has an additional step that prioritises image data. A DCT [14] is applied to each 8×8 block (after subtracting 128), then the less important DCT coefficients are wiped out using quantisation
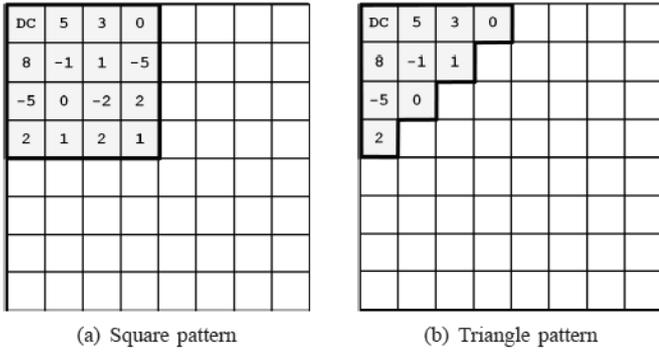
(a) Square pattern   (b) Triangle pattern

Figure 1. Zonal DCT patterns $K = 4$ (redrawn from [6])

matrix. A quality coefficient can be used to adjust compression ratio to get the expected video frame quality. Afterwards, the elements of each block are partitioned into up to 13 levels using a zigzag scanning. Finally, the 64 elements of each block are linearised and run-length encoding (RLE) is applied to the result.

A D-frame is the result of subtracting current frame from the previous M-Frame. Each 8×8 block is assigned a priority depending on the number of zeros it contains. Blocks with minimum number of dissimilar pixels (maximum of zeros) have the lowest priority and can be prevented from being transmitted. Finally, RLE and then Huffman coding are applied. In [5], it is suggested to use a variable length GOP (Group of Pictures) instead of transmitting an M-frame at regular intervals.

B. Optimising DCT Computation

Since radio is the main energy consumer in a sensor node, a WVSN with longer lifetime can result from reducing the amount of data to be transmitted/received. Video compression appears to be a good start ; however methods used nowadays in computer networks are not suitable to constrained networks such as WSN. In fact, a DCT consumes about 60% of the total consumed energy to encode an image [15]. This is why DCT complexity has to be reduced. Methods to reduce the DCT computation complexity can be either based on *zonal* or *fast* DCT algorithms. Only a subset of the DCT coefficients are computed in the zonal DCT while in the fast DCT, redundant operations are eliminated. In order to be more adapted to WSN, authors of [6] combined these two methods and proposed *fast zonal DCT*. They considered both square and triangular [16] forms of the zonal DCT along with LLM algorithm to fast compute the DCT. Another fast DCT that is based on shift instead of multiplication operations called binDCT-C is proposed in [17].

As shown in Figure 1, given an 8×8 block, only the upper-left square or triangular portion, with side length $K \in [2..8]$, is considered. In this way the DCT complexity is reduced as shown in Figures 2 and 3 where the flow graph of LLM and binDCT-C algorithms respectively for $K = 4$ are illustrated. Table I summarizes the cost in terms of the number of required operations (addition, multiplication, shift) of fast zonal DCT using LLM and binDCT-C algorithms.
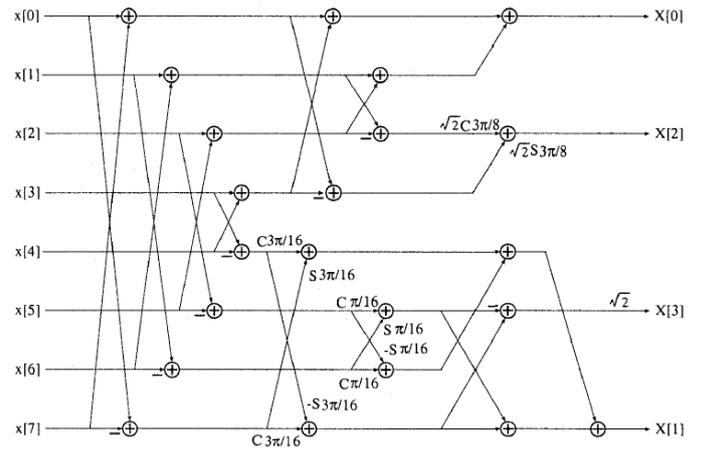

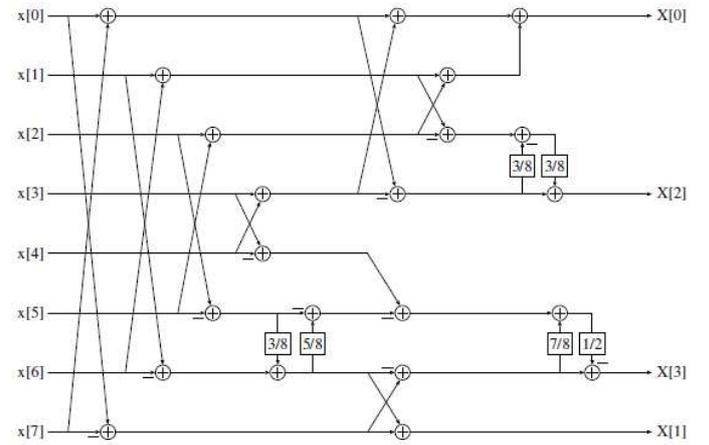
Figure 2. Flow Graph of LLM $K = 4$ (redrawn from [6])

Figure 3. Flow Graph of BinDCT $K = 4$

III. EvalVSN Architecture and Main Functionalities

EvalVSN (Figure 4) follows the main principle behind EvalVid where the interface with the real or simulation-based experiment is insured through the use of trace files : the sender and the receiver trace files. Based on the user parameters (the video clip to be encoded, the DCT computation method, the number of priority levels ...), EvalVSN generates a sender trace file (ST) that gives the sender application the required information about the data packets to be sent. In order to generate the ST file, two phases are required : a pre-processing

Table I. Fast zonal DCT cost for different values of K

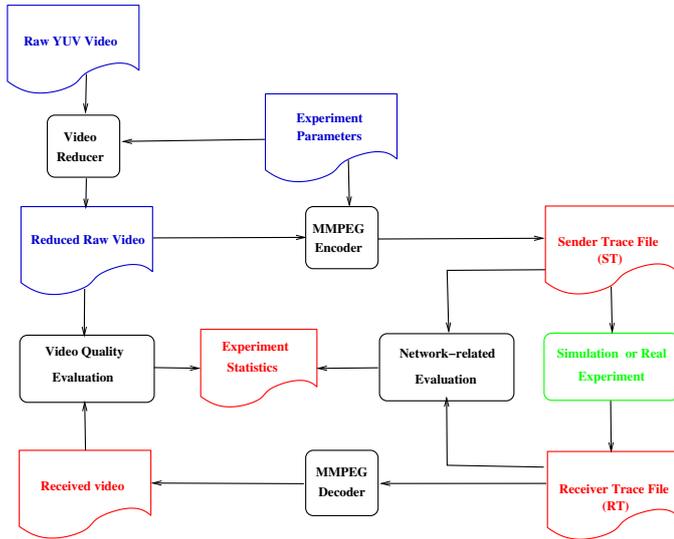| K	LLM				binDCT-C			
	1-D DCT		2-D DCT		1-D DCT		2-D DCT	
	Add.	Mul.	Add.	Mul.	Add.	Shift	Add.	Shift
8	29	11	464	176	30	13	480	208
7	28	11	348	130	28	12	420	180
6	26	10	364	140	28	12	392	168
5	25	9	325	117	28	12	364	156
4	24	9	288	108	27	11	324	132
3	23	8	253	88	19	6	209	66
2	20	6	200	60	13	2	130	20

Figure 4.   EvalVSN Architecture

and an encoding phase. A post-processing phase is performed in order to construct the received video and provide statistics based on the receiver trace file (RT) that results from the conducted simulated or real experiment. In our package, we provide application modules that make use of EvalVSN trace files in TinyOS [10], ns2 [11] and Castalia[12] environments.

A. Pre-processing Phase

This phase is required to make the video clip more realistic with respect to WSN characteristics. In fact, in the current state of the art, sending video sequences with an FPS (frames per seconds) of 30 fps is simply unfeasible due to the limited resources of a WSN. In order to be able to use reference video sequences such as those available at [18], we introduce this pre-processing phase that allows to consider only a subset of the frames resulting in a more reasonable FPS. For instance, consider the *Hall Monitor* video sequence that lasts 10 seconds and consists of 300 frames in QCIF resolution (176×144). If we want to consider this sequence as a video captured at $1fps$, then only 1 frame every 30 is kept in the reduced video clip with only 10 frames. It is worth mentioning that this phase is optional and the user can keep the original FPS.

B. Video Encoding Phase

In this phase, MMPEG codec (overviewed in Section II-A) is applied on the video clip with the following changes :

- The work in [19] showed that depending on the sensor node hardware, image compression (even based on fast zonal DCT) may introduce additional energy consumption compared to the case where the image is transmitted without compression. This is why in our tool, we give the user the possibility to not compress a given frame if a minimal variance with the previous frame is observed. In this case, the application can additionally decide to not transmit the frame at all. To do so, the user has to give a positive value to the *threshold similarity* parameter. If this latter is set to 0 then all the frames will be compressed.

- The first frame is always encoded as an M-frame. A subsequent frame is either encoded as an M-frame or a D-frame depending on its similarity with the previous M-frame. Let $X \times Y$ be the resolution of a frame and FZP be the number of zero pixels in the result of subtructing the previous M-frame from the current frame. If $FZP < qualityCoef \times (X \times Y)$ then the current frame is encoded as an M-frame otherwise it is encoded as a D-frame. This means that when $qualityCoef$ is increased then the number of M-frames is increased. This parameter has to be specified in the configuration file by the user.

- We allow the user to use the traditional DCT implementation and give him the ability to experiment less complex DCTs that are more suitable to WSN. In this version, LLM and binDCT-C fast zonal DCTs are implemented using a triangular pattern. We chose to implement a triangular rather than a square pattern since the former is more adequate to the decomposition in priority levels using the zigzag scan. Higher priority data are located in the upper-left triangle in the 8×8 blocks. If for some reason, only a subset of the data can be transmitted then the application layer can take the decision to only encode and transmit data with the highest priority. This allows saving the video sensor resources from encoding all the frame while only a subset can effectively be delivered to the sink.

- Based on our experiments, we found that applying Huffmann coding in addition to RLE on D-frames may generate very large D-frames. We chose to use only RLE encoding.

The output of the encoding phase is the sender trace file (ST) whose each line is structured as follows :

1) time at which this packet has to be sent ;
2) the number of the frame to which belong the data of this packet ;
3) the type (M or D) of the frame to which belong the data of this packet ;
4) this packet priority ;
5) this packet payload size ;
6) this packet payload (data) in hexadecimal.

C. Post-processing Phase

The experiment tool has to provide a receiver trace file (RT) that is potentially different from the ST file due to network losses. Missing lines in the RT tell about the set of packets that have been lost in the network before achieving the sink. Statistics on the conducted experiment are generated by EvalVSN based on the trace files (ST and RT) in addition to the sent and the received (reconstructed) video clips. The provided metrics can be either *network-related* or *video-related*.

We implemented the PSNR (Peak Signal to Noise Ratio) and the SSIM (Structural SIMilarity) [20] metrics 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. EvalVSN provides the mean PSNR computed on a per frame basis. The PSNR

between the sent (s) and the received (r), possibly distorted video frame is computed using :

$$PSNR(s,r) = 20log\frac{V_{peak}}{MSE(s,r)} \quad (1)$$

MSE is the mean square error which is the average of the square of the errors (pixel differences) of the two images and V_{peak} is the maximum possible pixel value.

SSIM exhibits much more consistency with subjective measure compared to PSNR and includes comparisons of luminance, contrast and structure as follows :

$$SSIM = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \times \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \times \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (2)$$

where x and y are two nonnegative 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 give the following values for the three constant parameters : $C_1 = (K_1 L)^2$, $C_2 = (K_2 L)^2$, $C_3 = C_2/2$ with $K_1 = 0.01$, $K_2 = 0.03$ and $L = 255$.

Other statistics such as the number of M-frames and D-frames as well as the number of generated packets per priority level in addition to experienced loss ratios related to each frame type and priority level are provided by the post-processing phase.

IV. SIMULATIONS

In this section, we limit ourselves to give a use example of EvalVSN rather than optimizing the transmission process which is beyond the scope of this paper. The reader can refer to [21] for another use example of EvalVSN where the performances of a multipath routing protocol along with a multiqueue multipriority scheme is performed. Here, we aim to illustrate the use of EvalVSN on the hall monitor video clip using ns2 simulator with the parameters summarized in table II. FPS is set to 1 so only 10 among 300 frames are considered. In order to get all these 10 frames encoded, the similarity threshold is set to 0. The quality coefficient is set to 0.1 which results in encoding only the first frame as an M-frame ; the 9 others are encoded as D-frames. We repeated the same simulation with the three DCT implementations and varied the parameter K of the fast zonal DCTs.

As stated before, EvalVSN provides the reference PSNR (SSIM) that allows assessing the video compression quality before transmission. This is done in order to separate the distorsion due to compression from the one due to experienced losses in the WSN. The obtained mean reference PSNR (SSIM) are reported in table III. We can see that the traditional DCT achieves the best PSNR (SSIM) since all the data of each frame are encoded. When using a triangular fast zonal DCT, the PSNR of reference decreases and especially when K decreases. This distortion can be acceptable if we consider the compression complexity reduction induced by the fast zonal DCT algorithm (see table I). One can consider the trade-off between distortion and compression cost in the design of WVSN applications.

Table II.   SIMULATION PARAMETERS

Network Parameters	Area dimensions	1000×1000
	Number of sensors	100
	Inter-packet delay	0.025 s
	Routing Protocol	AODV
	MAC	IEEE 802.11
Video Parameters	Initial number of frames	300
	Raw video source	hall.avi
	Raw Video Resolution	176×144
	Raw Video Duration	10 seconds
	FPS to apply to the video to send	1 fps
Codec Parameters	Encoder name	MMPEG
	Similarity threshold	0
	Quality Coefficient	0.1
	Max. number of priority levels	13
	DCT	DCT, LLMDCT, BINDCT
	K	2,3,4,5,6,7,8

Table III.   REFERENCE PSNR AND SSIM

K	Reference PSNR (dB)		Reference SSIM	
	LLMDCT	BINDCT	LLMDCT	BINDCT
8	37.33	37.01	0.99	0.99
7	34.32	34.13	0.97	0.97
6	30.98	30.87	0.95	0.95
5	28.26	28.10	0.92	0.91
4	25.98	25.87	0.87	0.87
3	23.65	23.55	0.79	0.79
2	21.98	21.93	0.71	0.71

Using the ST file, the sender application agent (at the video sensor) developed for ns2 performs a CBR (constant bit rate) transmission of the video data packets. The receiver application agent (that runs at the sink side) records the received data packets in the RT file. The post-processing phase of EvalVSN is applied to reconstruct the video clip based on the received data packets and generate statistics on the conducted experiment. Figure 5 shows the 5th frame of the reduced hall monitor video before transmission and after it is received by the sink for different DCT algorithms and K values.

The first observation we can make is that all the received images have experienced distortion due to transmission losses. It is worth saying that general conclusions can not be done from these simulations since each simulation is repeated once and the purpose of this section is only to illustrate the use of EvalVSN. However, we can see that better performances can be achieved using a zonal DCT. For instance, When $K = 8$ or even $K = 4$ (for LLM version), we can see that it is possible to obtain a higher PSNR for the received frame than the traditional DCT.

In order to be able to interpret the achieved video quality results depicted by Figure 5, information about loss rates are necessary. Figure 6 shows the loss ratio experienced by data packets that belong to M or D-frame in addition to the overall loss ratio. We can see that in this set of experiments, the LLM DCT gets the lowest loss ratio compared to the traditional and binDCT-C which explains its highest achieved PSNR.

V. CONCLUSION

In this paper, we presented a new transmission and evaluation tool targeted to WSN called EvalVSN and showed how this tool can be used to evaluate a video transmission in WSN. EvalVSN allows to make use of priority levels in addition to

DCT - $PSNR = 45.66dB$ DCT - $PSNR = 21.13dB$

LLM $K = 8$ - $PSNR = 37.33dB$ LLM $K = 8$ - $PSNR = 24.87dB$

binDCT $K = 8$ - $PSNR = 37.01dB$ binDCT $K = 8$ - $PSNR = 21.76dB$

LLM $K = 4$ - $PSNR = 25.98dB$ LLM $K = 4$ - $PSNR = 24.41dB$

binDCT $K = 4$ - $PSNR = 25.87dB$ binDCT $K = 4$ - $PSNR = 20.54dB$

(a) Sent frames   (b) Received frames

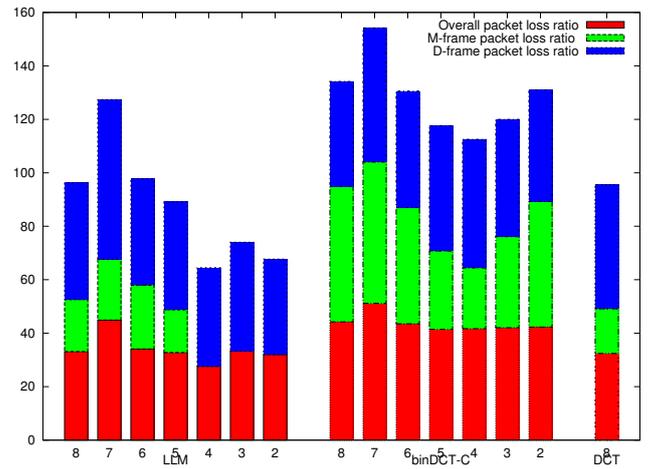Figure 5. The 5th frame with different DCT and K values

Figure 6. Packets Loss Ratio for different values of K

reduce the DCT complexity. It generates both network and video quality related metrics in order to quantitatively assess the performances of one's transmission strategies. Application modules that make use of EvalVSN trace files are available in TinyOS, ns2 and Castalia environments.

In the future, we expect to implement other encoding techniques as the image compression community is currently active in proposing new ones. Other application modules are also planned to be developped in other environment such as Contiki. Moreover, we aim to enhance our application modules by providing a more realistic energy model for data encoding. Finally, a C++ library is under developpment to allow better integration of compression algorithms to simulation environment so a decision on how to encode a given frame is performed on the fly and on a per-frame basis depending on the network status.

REFERENCES

[1] M. H. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. B. Srivastava, "Cyclops: in situ image sensing and interpretation in wireless sensor networks," in *SenSys*, 2005, pp. 192–204.

[2] W. chi Feng, B. Code, E. Kaiser, M. Shea, and W. chang Feng, "Panoptes: A scalable architecture for video sensor networking applications," *ACM Transactions on Multimedia Computing, Communications and Applications*, January 2005.

[3] E. Orellana-Romero, J. SanMartin-Hernandez, C. Duran-Faundez, V. Lecuire, and 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.

[4] J. Klaue, B. Rathke, and 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, ttp://www.tkn.tu-berlin.de/research/evalvid/.

[5] H. S. Aghdasi, M. Abbaspour, M. Moghadam, and Samei, "An energy-efficient and high-quality video transmission architecture in wireless video-based sensor networks," *Sensors*, vol. 8, pp. 4529–4559, August 2008.

[6] L. Makkaoui, V. Lecuire, and 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.

[7] Z. Wang, "Pruning the fast discrete cosine transform," *Communications, IEEE Transactions on*, vol. 39, no. 5, pp. 640–643, 1991.

[8] C. Loeffler, A. Ligtenberg, and 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.

[9] Y. Baziz, M. Maimour, and B. Kechar, "Evalvsn," http://page-perso.cran.uhp-nancy.fr/Perso_Maimour/evalvsn/.

[10] C. Lu, "Berkley motes and tinyos," 2001.

[11] "Network simulator 2." [Online]. Available: http://www.isi.edu/nsnam/ns

[12] "Castalia," http://castalia.npc.nicta.com.au/.

[13] *Digital Image Processing Using MATLAB (DIPUM)*. Prentice Hall, 2004, ch. 8, pp. 282–333.

[14] Y.-J. Chen, S. Oraintara, and T. Nguyen, "Video compression using integer dct," in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 2, 2000, pp. 844–845 vol.2.

[15] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *Communications Letters, IEEE*, vol. 12, no. 6, pp. 411–413, 2008.

[16] A. Mammeri, A. Khoumsi, D. Ziou, and 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.

[17] J. Liang and T. D. Tran, "A fast multiplierless block transform for image and video compression," *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 3032–3044, December 2001.

[18] "Yuv video sequences," http://trace.eas.asu.edu/yuv/.

[19] L. Makkaoui, "Compression d'images dans les réseaux de capteurs sans fil," Ph.D. dissertation, Université de Lorraine, 2012.

[20] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[21] H. Zeghilet, Y. Baziz, M. Maimour, B. Kechar, and N. Badache, "The effects of interference on video quality over wireless sensor networks," in *Proceeding of the 2013 International Workshop on Communications and Sensor Networks (ComSense-2013)*, Ontario Canada, October 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187705091300851X