Sim-LIT: A simulation tool for image quality assessment in wireless visual sensor networks under packet loss conditions (extended abstract)

Eric Orellana-Romero*, Javier SanMartin-Hernandez*, Cristian Duran-Faundez*, Vincent Lecuire[†], and Cristhian Aguilera*

*Departamento de Ingeniería Eléctrica y Electrónica, Facultad de Ingeniería,
Universidad del Bío-Bío, Concepción, Chile
Email: {eorellan,javsanma}@alumnos.ubiobio.cl, crduran@ubiobio.cl, cristhia@ubiobio.cl
[†]Centre de Recherche en Automatique de Nancy (CRAN),
Nancy-Université, CNRS, Vandœuvre-lès-Nancy, France
Email: vincent.lecuire@cran.uhp-nancy.fr

Abstract—**In this paper we present Sim-LIT, a simulator of packet loss effects on image quality over wireless sensor networks. It can be used to evaluate error resilience during the communication. In this first version it allows the evaluation of block interleaving methods. Sim-LIT is highly configurable providing several options being useful to rapidly evaluate interleaving algorithms or to perform extensive tests.**

Keywords-**Wireless visual sensor networks; image communication; packet loss; block interleaving; simulation**

I. Introduction

As wireless sensor networks (WSNs) inherently involve packet loss [1], new techniques for energy-efficient and robust data communications are required. Moreover, WSNs with vision capabilities [2] present particular problems because of the transmission of hundreds of packets per captured image. Here, traditional ACK and FEC techniques are poorly suited and classical compression techniques must be hardly adapted for resource constrained systems. In this matter, compression should be performed over independent blocks in order to avoid making useless many information packets due to the loss of a previous one. Fig. 1 shows effects of packets losses over coded/non-coded images and a sequential transmission.

Original image (128×128 pixels)

(1% packet loss)   Received pixels   Received pixels

(5% packet loss)   Reconstructed image   Reconstructed image

Compressed with JPEG at 3.4bpp   Non-compressed (25% packet loss)   Non-compressed (29% packet loss)
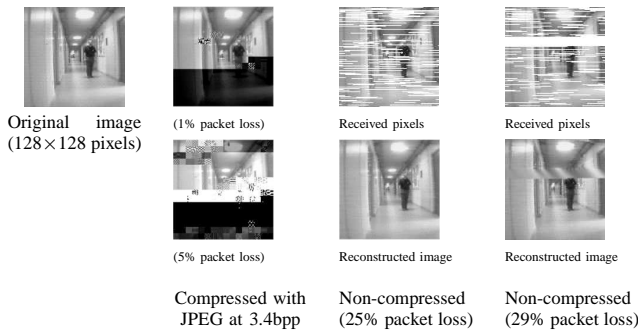
Fig. 1. Different effects over received images due to communication losses.

In the figure, the second column shows that few packet losses lead to unusable reconstructed images when we compress with techniques based on correlated coded blocks, such as traditional JPEG. Non-compressed image transmission can be a good approach for environments with high packet loss, but sequential communications can lead to the lost of big important portions of the image due to burst packet loss (an example is shown in fourth column).

A particular example for a low cost method for robust image transmission is block interleaving (or shuffling) [3], [4]. Here, neighboring (coded or raw) blocks of information are sent in different packets in the transmission sequence. Thus, if packets are lost, we have better probability of retrieving enough information to reconstruct a missing block by using the correctly received surrounding ones. In order to test the proposed strategies, many tools allow simulation of WSNs issues such as communication protocols [5] and power consumption [6], but there are not many available software oriented to image communications over sensor networks.

In this paper, we present Sim-LIT (Simulator for Lossy Image Transmission), a simulator of packet loss on image communications over wireless vision sensor networks (WVSNs). This first version includes the simulation of interleaving algorithms for image transmissions using coded and non-coded blocks. Section II presents the processing and transmission schemes possible to simulate with Sim-LIT. Section III explains implementation and use of the simulator. Finally, Section IV concludes and gives some future directions.

II. Block interleaving and communication models

The adopted models are simple. An input image I is a $(H \times W)$ matrix, $I = \{I_{r,c}\}$, with $r, c \in \mathbb{Z} \wedge 0 \leq r < H \wedge 0 \leq c < W$, each $I_{r,c}$ pixel containing its b bits per pixel. Now, we consider a scheme communication $\overleftrightarrow{\Gamma}$, were we transmit I in $q = \lceil \frac{H \times W \times b}{m} \rceil$ packets P, were m is the number of bits available for image data transmission in a packet. In a traditional block based communication, we firstly cut the image into $\frac{H \times W}{H_B \times W_B}$ blocks $B_{i,j}$, were H_B and W_B are the block's height and width, $B_{i,j} = \{I_{r_B,c_B}\}$, were $i.H_B \leq r_B < H_B.(i+1), j.W_B \leq c_B < W_B.(j+1)$.

TABLE I
LIST OF SOME OPTIONS FOR SIM-LIT

Option	Description
`-adapted-interleaving`	Option to use adapted interleaving (see doc)
`-b`	Bits per pixel (default: 8)
`-detailed-res-file`	Output file for detailed results (default: `./detailed_results.dat`)
`-gui`	Executes graphic interface
`-hb`	Block's height (default 1)
`-help`	Display help menu
`-ipath`	Defines images' path (default: `./images/`)
`-loss-file`	Defines loss file (default: `./loss-file.dat`)
`-no-console-messages`	Option to not display messages in console
`-no-detailed-res-file`	Option to not store detailed results file
`-no-resumed-res-file`	Option to not store resumed results file
`-no-res-img-store`	Option to not store resulting images
`-resumed-res-file`	Output file for resumed results (default: `./resumed_results.dat`)
`-s`	The number of bits available for image data per packet (default: 216)
`-wb`	Block's width (default: 1)

In a sequential communication, we assign and send the ith block to the tth packet, with $t = \lfloor \frac{i \cdot b}{m} \rfloor$. An interleaving disrupts this sequence. It can be considered as an bijective function $\vartheta : I \to I'$, were I' is a new bitmap with all original blocks $B_{i,j}$ are stored in a position (i', j') (likely different to (i, j)). An adapted version (requiring less memory and calculations) considers sequential operations $B'_{i,j} \leftarrow B_{i',j'}$ [3]. Each packet p_l has a probability to be lost during communication. Different loss models can be applied. At the receiver side, lost data are estimated by averaging the well received neighboring ones.

III. SIM-LIT

The simulator was constructed in C++, with g++ compiler over Linux. Main Sim-LIT's classes are shown in Fig. 2.
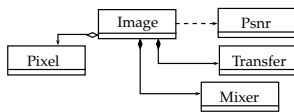

Fig. 2.   Sim-LIT's classes.

Each class is provided as both a `.c` and a `.h` files. The program is built with a `Makefile`. The execution is through the command `simlit [options]`. Some of the current available options and parameters are listed in Table I.

The interleaving algorithm must be implemented in the `Mixer` class. The program opens image files (from a given path) extracting the header parameters. `Image` class attributes are set, including size, codification and bitmap. The image can be then interleaved, using given user parameters or defaults. Then, the image is packetized. Packet loss simulation is performed using a loss-file with q registers using the following format: `[reg_id] rec_flag`, were `reg_id` is a sequential number and `rec_flag` is a boolean indicating `1` or `0`

depending on whether the packet was received at the decoder side or not. This file could be created by using user defined distribution, network simulators or real packet captures. A complementary tool for creating this file randomly is also provided for testing. After this, a new image is created reallocating correctly received data. Empty pixels are reconstructed by averaging well received neighbors (other error concealment methods are also implementable). PSNR between the original image and the reconstructed image is also provided as a quality metric. Various sample programs in C++ and `sh` are also provided, to facilitate calculations over extensive tests. Some execution options such as `-no-console-messages` or `-no-res-img-store` allow speed-up calculations. Some sample images obtained with Sim-LIT are shown in Fig. 3.
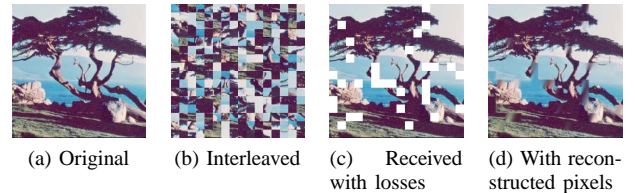

(a) Original   (b) Interleaved   (c) Received with losses   (d) With reconstructed pixels

Fig. 3.   Image samples obtained with Sim-LIT.

IV. CONCLUSION

Sim-LIT is a configurable simulator allowing image transmission with error robust methods such as block interleaving for WVSNs. It can be used to rapidly evaluate interleaving algorithms or to perform extensive tests. For now, this first version only supports .bmp format. The loss models depends on the constructed loss files, but it can be possible to connect with WNS's simulator such as TOSSIM [5]. We use Sim-LIT to evaluate existing and new interleaving schemes. Further versions will be developed improving fastness (may be with parallel programming), robustness, and other factors. Also, we will incorporate other methods and options.

REFERENCES

[1] C. Yeo and K. Ramchandran, "Robust distributed multi-view video compression for wireless camera networks," in *Visual Communications and Image Processing*, vol. 6508, San Jose, CA, USA, January 2007.

[2] S. Soro and W. Heinzelman, "A survey of visual sensor networks," *Advances in Multimedia*, vol. 2009, p. 21 pages, 2009, article ID 640386.

[3] C. Duran-Faundez and V. Lecuire, "Error resilient image communication with chaotic pixel interleaving for wireless camera sensors," in *REAL-WSN'08*, ACM, Ed., Glasgow, Scotland, UK, April 2008, pp. 21–25.

[4] S. Ye, Y. Lin, and Y. Yi, "Energy-aware interleaving for robust image transmission over visual sensor networks," in *IET-WSN 2010*, nov. 2010, pp. 317 –322.

[5] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *SenSys '03*. New York, NY, USA: ACM, 2003, pp. 126–137.

[6] V. Shnayder, M. Hempstead, B.-R. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *SenSys'04*, Baltimore, MD, November 2004.