



# Error resilient image communication with chaotic pixel interleaving for wireless camera sensors

Cristian Duran-Faundez, Vincent Lecuire

## ► To cite this version:

Cristian Duran-Faundez, Vincent Lecuire. Error resilient image communication with chaotic pixel interleaving for wireless camera sensors. Workshop on Real-World Wireless Sensor Networks, REALWSN'08, Apr 2008, Glasgow, United Kingdom. pp.21-25. <hal-00271146>

**HAL Id: hal-00271146**

**<https://hal.science/hal-00271146v1>**

Submitted on 8 Apr 2008

**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.



HAL Authorization

# Error Resilient Image Communication with Chaotic Pixel Interleaving for Wireless Camera Sensors

Cristian Duran-Faundez  
Cristian.Duran@cran.uhp-nancy.fr

Vincent Lecuire  
Vincent.Lecuire@cran.uhp-nancy.fr

Centre de Recherche en Automatique de Nancy (CRAN),  
Nancy-Université, CNRS  
Boulevard des Aiguillettes, B.P. 239  
F-54506, Vandœuvre-les-Nancy, FRANCE

## Abstract

New applications of wireless sensor networks require vision capabilities. Considering the high loss rates found in sensor networks, and the limited hardware resources of current sensor nodes, low-complexity robust image transmission must be implemented, avoiding as much as possible the need for retransmission or redundancy. In this paper we propose a pixel interleaving scheme based in Torus Automorphisms, thus, neighboring pixels are transmitted in different packets. Hence, if packets are lost, we have a high probability of retrieving enough information to obtain an approximation of the original value. Results show an increase of the image quality in comparison with a sequential raw image transmission approach, while preserving similar energy consumptions, time and low-complexity.

## 1 Introduction

Many potential applications of wireless sensor networks (WSNs), such as object detection, counting, and tracking, require vision capability [1]. With the availability of low-power small-scale camera sensors, like Cyclops [6] and CmuCam [7], such applications are possible nowadays. However, digital images contain a large amount of data, thus, communication of a single image generates several packets. As a result, source image sensors, as well as the nodes forwarding packets towards the sink, consume their energy much faster than sensors collecting and forwarding scalar data such as, e.g., a 2-byte long temperature. In addition,

WSNs are subject to high packet losses due to wireless channel errors, collisions, congestion or unexpected node failures, that can cause degradation in the reconstructed image quality. Therefore, special challenges for energy-efficient and robust image communication must be addressed. It seems obvious that energy savings could be achieved by image compression. However, image compression removes spatial redundancy between neighboring pixels so that packet loss tolerance strongly decreases. When an image is compressed, the use of an ARQ or FEC technique is needed to control packet losses and to maintain a certain image quality. In return, the energy cost of an ARQ or FEC technique will counterbalance the energy savings achieved by image compression. Moreover, works in [5] demonstrated that the complexity of popular image compression algorithms (e.g., JPEG, JPEG2000, SPIHT) could lead to greater energy consumptions than the transmission of the uncompressed image. In order to avoid using ARQ or FEC, a simple way to reduce the effect of packet losses on image quality is pixel interleaving. Pixel interleaving schemes spatially de-correlate neighboring pixels (or blocks of pixels) by putting them into packets that are far distant apart from each other in the transmission sequence. If the interleaving scheme is properly designed, a missing pixel is likely to be surrounded by correctly received pixels. By doing so, the performance of an error concealment method may be significantly improved.

In this paper, we study the effectiveness of a pixel interleaving scheme, in a real wireless camera sensor, based on the torus automorphisms (TA) technique. We show that the TA algorithm is initially not optimized for implementation in resource-constrained camera sensors because of excessive memory cost. We suggest some simple modifications in the way the TA runs to address this issue. For validation purposes, we implemented our algorithm into a wireless camera sensor composed of a Mica2 processor/radio board and a ‘Cyclops’ imager. Then we performed some experiments for performance evaluation and comparison with an ARQ-based image transmission. Results show the efficiency of the adapted TA algorithm with regard to energy cost and reconstructed image quality. The amount of energy consumed by the adapted TA is about  $4\mu\text{J}$  per pixel. The energy cost for 1-hop transmission of a  $128 \times 128$ , 8-bpp monochrome image is of about 2374 mJ (including image capture and interleaving). Moreover, image quality remains acceptable above 40% of packet losses. In comparison, the energy cost for 1-hop transmission of the non-interleaved image with an ARQ protocol is at least 3690 mJ, when no packet retransmission occurs.

The remainder of this paper is organized as follows: In section 2 we present the problems of traditional image transmission, then we introduce and motivate the principles of pixel interleaving application. Section 3 presents the toral automorphism technique we adopted to perform pixel interleaving, explaining the general principles and a proposed algorithm to aid the application in camera sensor nodes. Experimental results in terms of image quality, energy consumption and execution time are given in section 4. Finally, in section 5, we summarize and provide some future directions.

## 2 Pixel Interleaving for Robust Image Communication

Figure 1 illustrates the typical effects of packet losses on the reconstructed image quality. In this example, we consider a camera sensor node capturing a  $128 \times 128$  8-bit monochrome image and sending it into 607 packets (see section 4.1). The original image is shown in figure 2(a). Figure 1(b) shows the received image at the destination, considering that the communication protocol provides no error correction mechanisms and only 71% of the packets were correctly received.

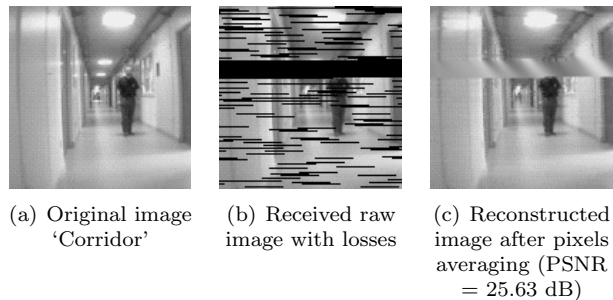


Figure 1: Visual losses by non-reliable image transmission without processing at the source level.

The black spaces represent missing pixels. Since pixels of the raw image were sequentially sent, missing pixels are perceived as lines on the pixels map. Losses can be concentrated in big areas, so, important information of the captured scene can be lost. As neighboring pixels are in most cases correlated to each other (except in edges or very detailed areas), we can apply a concealment method in order to compute an approximate intensity for each missing pixel. Figure 1(c) shows the reconstructed image after averaging of the received neighboring pixels. In some cases this error concealment can improve image quality. Indeed, on the image's bottom half, due to the losses distribution, missing pixels are almost perfectly reconstructed, and differences with the original image are practically imperceptible. On the other hand, since there is no information about missing pixels, a big area of the image on the top half can not be reconstructed by any method.

Using ARQ or FEC to deal with packet losses is questionable due to the energy limitation of sensor nodes. To avoid that, it is helpful to send neighboring pixels into different and distant packets. Hence, losses will be just scattered points in the image, being simplest to reconstruct by applying an error concealment method. This approach is known, roughly speaking, as pixel interleaving or 'mixing'.

Pixel interleaving has been used in a wide range of applications, in order to perform watermarking security [10, 3] or pixels recovery [8], among others. In the context of WSNs, the problem is to achieve an efficient method at the

lowest cost possible for the encoder (the source node).

### 3 Torus Automorphisms

Torus automorphisms (TA) [10] are strongly chaotic systems that can be used as a two-dimensional permutation transform. We adopted this method for mixing images because of its observed performance and simplicity, with almost no addition to the complexity of the calculus. In addition, as image is the most intrusive surveillance method, toral automorphisms provide security because of their chaotic behaviour, difficulting extraction from an unauthorized agent. In this section we explain the principles of this technique. Then, we will propose an algorithmic adaptation, to be efficiently used in resource-constrained wireless camera sensors.

#### 3.1 TA Principles

For simplicity purposes, we consider square images of  $N \times N$  pixels. We adopted the Torus Automorphism applied in [3]. The effect of TA on the pixel found at coordinates  $(x, y)$  is a new position  $(x', y')$  for the pixel calculated as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^n \begin{pmatrix} x \\ y \end{pmatrix} \mod N \quad (1)$$

where  $k$  is a selectable value and  $n$  is a scatter key ( $k, n \in \mathbb{N}$ ). Thus, by applying the transformation in formula 1 to all the pixels in an image  $I$ , we obtain a mixed image  $I'$ . An example of the TA application over the image ‘Corridor’ in  $128 \times 128$  pixels resolution is shown in figure 2. By applying TA

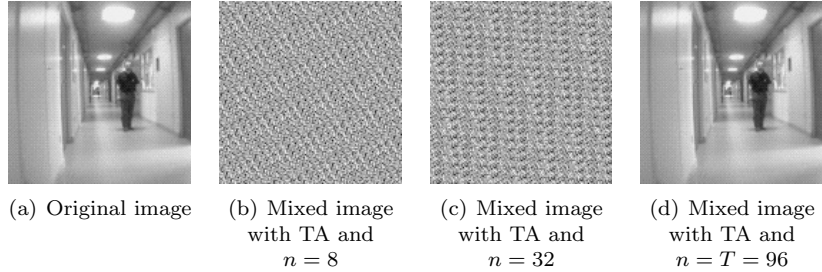


Figure 2: Torus Automorphisms applied over a  $128 \times 128$  ‘Corridor’ image ( $k = 1$  applied).

into the original image, at different steps, we obtain mixed images. Figures 2(b) and 2(c) show the transformed image at scattering keys of  $n = 8$  and  $n = 32$ , respectively, and  $k = 1$ . Note that with  $n = 96$  the TA results in the original coordinates, obtaining the original pixels map (see figure 2(d)). In fact, there is always a particular scattering key,  $T$ , which makes an image similar to that of

the original ( $\forall N \in \mathbb{N}, \exists n = T/I'_T = I$ , where  $I'_n$  is the transformed image after applying a scatter key of  $n$ ). This property can be used at the decoder side, by applying the same transform of equation 1 with scatter key of  $(T - n)$ .

### 3.2 Adapting TA to camera sensors

In standard implementation, Torus Automorphisms transform requires additional memory of the size of the input image to store the mixed image without overwriting the input pixels. Concerning memory access, each pixel is read and written twice to achieve both the TA transform and the packetization process. Note that the TA transform must be complete (i.e., all pixels of the input image have been projected into the mixed image) before starting the packetization process. Such an algorithm is not optimized for implementation in resource-constrained camera sensors because of excessive memory cost (and consequently excessive energy cost). We address this issue by applying an inverse approach: Instead of searching the projected position  $(x', y')$  of a given input  $(x, y)$  pixel, here  $(x', y')$  corresponds to the position of the input pixel to reach from a given projected position  $(x, y)$ . Once the position  $(x', y')$  is calculated, the corresponding pixel of the original image can be directly copied in the packet under construction. The pseudo-code of the proposed procedure is shown in algorithm 1. The algorithm starts by considering the projected position  $(0, 0)$ , and calculates the coordinate  $(x', y')$  for given  $x = 0$  and  $y = 0$ . Then the pointed pixel  $(x', y')$  is copied into the first place of the packet under construction. For each iteration, the algorithm repeats the same operations for the next projected positions,  $(0, 1)$ ,  $(0, 2)$ , and so on. When the packet under construction is completed, it is outputted to the radio transceiver to be sent, then a new packet is filled. This algorithm does not require much additional memory. In addition, each pixel is read and written once instead of twice for the standard algorithm. Consequently, our proposal is more efficient for implementation in resource-constrained wireless camera sensors. In the following section, performance of the TA algorithm is evaluated for the Cyclops camera mounted on a Crossbow Mica2 mote.

## 4 Experimentation and Analysis

Various experiments were conducted to evaluate our proposal. We implemented the TA in a real platform and we compared performance between mixed and non-mixed image transmissions. Performance criteria are defined in terms of image quality, energy consumption, and execution time.

### 4.1 Experimental platform

Wireless nodes used in our experiments consist of a Cyclops camera [6] attached to a Crossbow Mica2 mote [4]. Cyclops is a digital imager that allows capture

---

**Algorithm 1** Adapted TA-based pixel interleaving

---

```
1:  $i \leftarrow 0$  {position of data in packet}  
2:  $H \leftarrow ImageHeight$ ,  $W \leftarrow ImageWidth$   
3: for  $y = 0$  to  $H - 1$  do  
4:   for  $x = 0$  to  $W - 1$  do  
5:     Calculate  $(x', y')$  of position  $(x, y)$  using TA  
6:      $Packet.data[i] \leftarrow I[x', y']$   
7:     if (packet is full) or  $((x, y) = (W - 1, H - 1))$  then  
8:       Send packet  
9:        $i \leftarrow 0$   
10:    else  
11:       $i \leftarrow i + 1$   
12:    end if  
13:  end for  
14: end for
```

---

and processing of monochrome and color images with a relatively low-power consumption. It consists of a medium quality imager ADCM-1700 CMOS camera module, an ATMEL ATmega128L micro-controller unit (128KB of Flash program memory and 4KB of SRAM), a CPLD, 64KB of SRAM and 512KB of FLASH memory for data storage. It has a 51-pin connector to interface with Mica2 (or MicaZ) motes. The Mica2 motes consist of an Atmel ATmega128L micro-controller, a Chipcon CC1000 radio transceiver, 512KB of Flash Memory, and 4KB of EEPROM.

We implemented a one-hop transmission between a camera node, transmitting with a power out of -20dBm, and the sink (composed of a MIB520 and a Mica2 mote). A program developed in TinyOS/NesC [9] allows consecutive transmission of  $128 \times 128$  grayscale images towards the sink. In order to obtain packet losses, we added from 1 to 4 Mica2 motes running as packet generators for adding background traffic. The experimental topology is depicted in figure 3.

We get and align packets received from the camera source and we measure image quality, by using the obtained data raw to emulate data loss over some images on a computer application, developed in C language and GNU gcc compiler. For comparison purposes, we adopted the Peak Signal to Noise Ratio (PSNR) as image quality metric. Lost pixels were reconstructed by using the average between the values of adjacent pixels.

## 4.2 Implementation details

Nodes communication is achieved by using **MoteRelay** and **GenericBase** sample applications, available in the CVS repository of the Cyclops firmware [2]. **captureRadioTest** allows the capture of images in selectable formats and resolutions by a Cyclops imager and its dumping through the serial port. A connected Mica2 mote programmed with the **MoteRelay** application can receive and

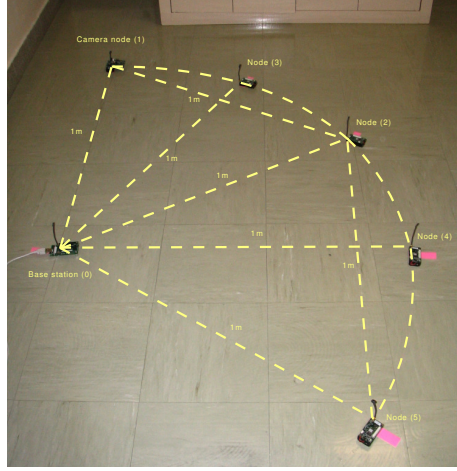


Figure 3: Experimental topology

relay data by the radio transceiver. At the base station, a mote programmed with the **GenericBase** receives data from the radio and retransmits to the serial port. At the same time, a computer application reads the serial port and stores the packets for subsequent analysis. At sensor level, communicating applications use the basic **TOS\_Msg** message struct. The payload field has a 2-byte header defined by the struct **serialDumpHeader\_s**, that indicates the remaining number of bytes to be transmitted in the sequence. As the size of the payload has been defined on 29 bytes (we keep this default parameter), we have 27 bytes available to send image data in each packet.

A second application replaces the **captureRadioTest** on the Cyclops camera, adapting the toral automorphism explained before. In order to avoid the handling of overly large values in the calculation of the TA, we suggest the use of small values for  $k$  and  $n$ . We arbitrarily chosed  $k = 1$  ('cat map') and  $n = 8$ , to obtain enough distance between the calculated coordinates, and to handle storable values of up to 2 bytes. In addition, we inserted the pre-calculated matrix  $A = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^n$ , so, we can directly calculate  $x'$  and  $y'$ .

### 4.3 Results

In section 2 we exposed the problems of 'non-mixed' image transmission when packet losses occur. Figure 4 shows the thrown results when we apply TA for the same image and same packet loss pattern as in figure 1. We see that the PSNR was significantly improved when the image is mixed before transmission. Visually, the image recovers important information, while observing a better spread out degradation through the entire pixels map.

Indeed, as neighboring pixels have been scattered in different packets, losses



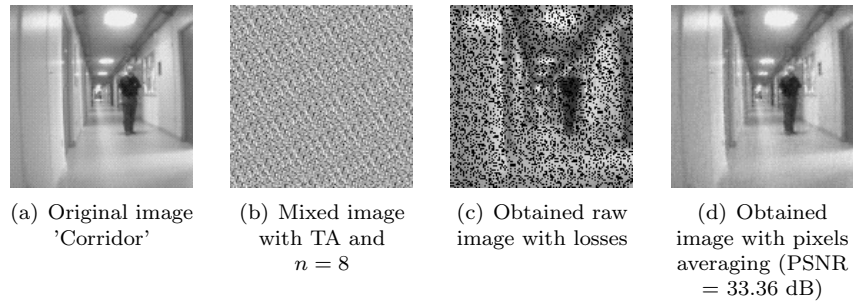


Figure 4: Torus Automorphisms applied over a  $128 \times 128$  ‘Corridor’ image.

are also distributed. The result is a smooth and gradual decreasing of the entire image quality, with the advantage of having an acceptable perception of the objects in the range of vision of the camera, even with high loss rates, as present WSNs applications. This gradual decreasing of the image quality is observed in the graphic of figure 5.

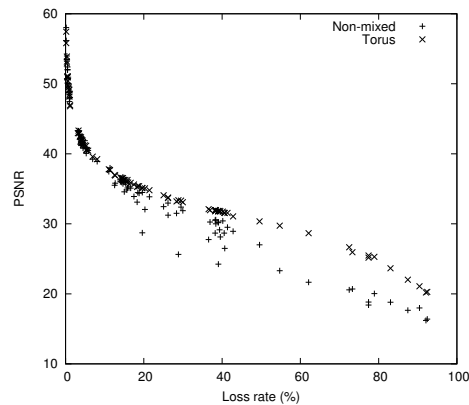







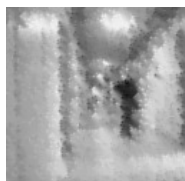


Figure 5: Image quality comparison for the image ‘Corridor’

TA present better results than a simple ‘non-mixed’ image transmission. Its application becomes more interesting with high loss rates, where we can see a marked difference on the obtained image qualities. The applied toral automorphism also presents a more controlled decreasing of the image quality. Indeed, we can observe that the distribution of points in the graphic, corresponding to TA application, decreases in a well discernible curve, while for the ‘non-mixed’ approach, since the image quality depends on the area affected by the loss of information, points are disorganized and in some cases present sharp contrasts for similar loss rates. We observe that when the loss rate does not surpass 40%, we can always retrieve an acceptable image quality. Beyond a loss rate of 60%, images start to be hardly recognizable. Sample images for both of the

considered strategies can be visually compared in table 1.

Table 1: Image quality visualization for various loss rates

Loss rate	Non-mixed	TA
20.27%	 PSNR = 32.05dB	 PSNR = 35.06dB
40.18%	 PSNR = 30.41dB	 PSNR = 31.74dB
62.1%	 PSNR = 21.65dB	 PSNR = 28.66dB
83.03%	 PSNR = 18.81dB	 PSNR = 23.66dB

We also evaluated the toral approach in terms of energy consumption. For this, a low ohmage, current sense resistor ( $1\Omega$ ) was connected in series with a camera node running and a DC power supply. The voltage drops over the DC power supply (3 Volts) and the resistor were gathered every 0.5 ms using an Agilent 54622A digitizing oscilloscope. Results show that our algorithm

does not increase considerably the energy consumption and execution time of the application. Obtained energy consumption and time are about 2307 mJ and 29.55 seconds respectively for the capture, processing and transmission of a  $128 \times 128$  non-mixed image, while for the TA application, these values are respectively about 2374 mJ and 30.20 seconds. This represents a cost of about  $4\mu\text{J}$  and  $40\mu\text{s}$  per pixel. We also measured the energy consumption for an ARQ-based image transmission. When no losses occur, energy consumption and time for capture, processing and transmission increase to about 3690 mJ and 48.95 seconds. In the presence of losses, these results can be greatly increased.

## 5 Conclusion and Future Work

This paper studied the application of a low-complexity pixel interleaving algorithm for robust image transmission over WSNs. We propose the application of a chaotic interleaving reached by a toral automorphism technique. Results show that TA application does not increase considerably the execution time and energy consumption on a wireless camera node, while increasing the quality of transmitted images over a perturbed channel even with high loss rates, and without need of additional memory allocations, complex calculus, redundancy or retransmissions.

Our work contributes to the development of simple and efficient image transmission on wireless camera sensor networks. Our future efforts will be focused in the analysis of TA and other interleaving techniques in order to improve our results. Furthermore, low-complexity compression algorithms will be combined, in order to achieve energy savings. Evaluations in real multi-hop platforms will be performed.

## 6 Acknowledgments

This work is supported by ANR (Project TCAP, No. 06-JCJC-0072) in France. We would like to thank Pr. Dominique Richier (Department of Networks and Telecommunications, University Institute of Technology, Nancy) for his aid in the energy consumption measurements.

## References

- [1] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921–960, March 2007.
- [2] Center of Embedded Network Sensing. CENS - CVS Repository. <http://cvs.cens.ucla.edu/>, 2004.
- [3] T.-S. Chen, J. Chen, and J.-G. Chen. A simple and efficient watermarking technique based on jpeg2000 codec. In *Proceedings of the Fifth International*

*Symposium on Multimedia Software Engineering*, pages 80–87, December 2003.

- [4] Crossbow Technology Inc. <http://www.xbow.com/>.
- [5] L. Ferrigno, S. Marano, V. Paciello, and A. Pietrosanto. Balancing computational and transmission power consumption in wireless image sensor networks. In *IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measures Systems (VECIMS 2005)*, Giardini Naxos, Italy, July 2005.
- [6] M. Rahimi, R. Baer, O. I. Iroezzi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the 3rd ACM Conference on Embedded Networked Sensor Systems (SenSys 2005)*, pages 192–204, San Diego, CA, November 2005.
- [7] A. Rowe, A. Goode, D. Goel, and I. Nourbakhsh. CMUcam3: An open programmable embedded vision sensor. Technical Report RI-TR-07-13, Carnegie Mellon Robotics Institute, Pittsburgh, Pennsylvania 15213, May 2007.
- [8] C. J. Turner and L. L. Peterson. Image transfer: an end-to-end design. In *Proceedings of the SIGCOMM '92 Symposium*, pages 258–268, Baltimore, Maryland, 1992. ACM Press.
- [9] UC Berkeley. TinyOS: An operating system for networked sensors. <http://www.tinyos.net/>.
- [10] G. Voyatzis and I. Pitas. Chaotic mixing of digital images and applications to watermarking. In *Proceedings of European Conference on Multimedia Applications, Services and Techniques (ECMAST'96)*, volume 2, pages 687–695, Louvain-la-Neuve, Belgium, May 1996.