Error Resilient Image Communication with Chaotic Pixel Interleaving for Wireless Sensor Networks

Cristian Duran-Faundez and Vincent Lecuire
{Cristian.Duran,Vincent.Lecuire}@cran.uhp-nancy.fr

Research Centre for Automatic Control (CRAN),
Nancy-University, CNRS, France

april 1st, 2008

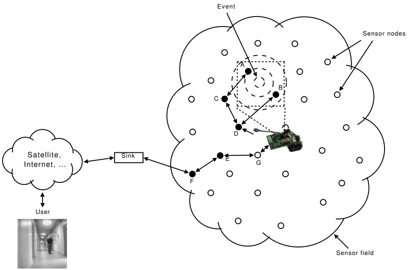
Planning

Introduction to image transmission over WSNs

Camera sensor networks

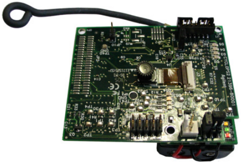
- A wireless sensor network where one or several nodes have image sensors (cameras).

- Applications
  - Surveillance and object recognition
  - Localisation and object tracking
  - Counting

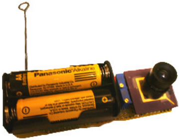


Introduction to image transmission over WSNs

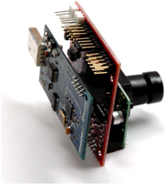
Current camera devices



(a) Cyclops camera (UCLA & Agilent) on Mica2 mote



(b) Aloha imager (Johns Hopkins University) on Mica2 mote



(c) Cmucam3 (Carnegie Mellon University) on Tmote

Figure: Different current camera devices for sensor networks

CRAN

Introduction to image transmission over WSNs
Reference platform

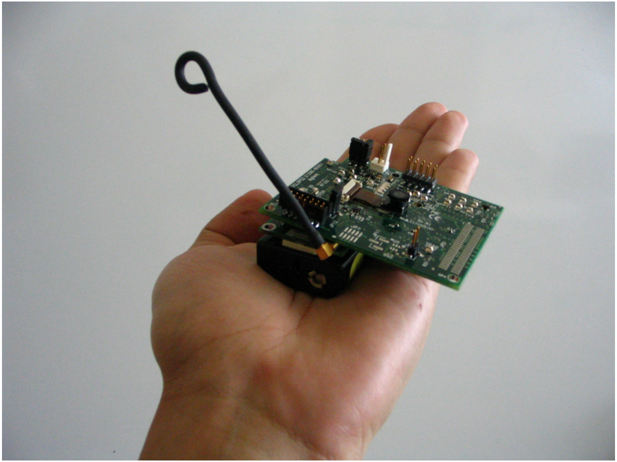


Figure: The Cyclops camera

Introduction to image transmission over WSNs

Reference platform

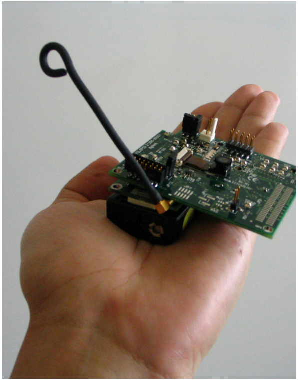


Figure: The Cyclops camera

Technical features

- Capture of images in selectable formats and resolutions
- ADCM-1700 CMOS imager
- ATMEL ATmega128L micro-controller (128KB memory program and 4KB SRAM)
- CPLD
- SRAM (64KB)
- Flash memory (512KB)
- 51-pin connector to interface with Mica2/MicaZ motes

Introduction to image transmission over WSNs
Constraints

- Low available resources (for processing, storage, etc.)

Introduction to image transmission over WSNs

Constraints

- Low available resources (for processing, storage, etc.)
- Big reported data losses

Introduction to image transmission over WSNs
Constraints

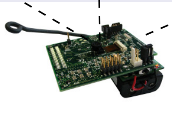
- Low available resources (for processing, storage, etc.)
- Big reported data losses
- Large amount of data (to process/transmit) ⇒ Big energy consumptions & time!!

Introduction to image transmission over WSNs
Constraints

- Low available resources (for processing, storage, etc.)
- Big reported data losses
- Large amount of data (to process/transmit) $\Rightarrow$ Big energy consumptions & time!!

Example.

Image 128×128
8-bit monochrome



Source's power out: -20dBm $\Longrightarrow$ 2307mJ & 29.55seconds

Pixel interleaving for robust image transport

Typical effects of packet losses

	2	3	4	5	6
68	164	162	164	155	146
69	148	162	162	162	139
70	164	162	X	146	146
71	148	162	148	157	139
72	164	162	164	157	146

Pixel interleaving for robust image transport

Typical effects of packet losses

	2	3	4	5	6
68	164	162	164	155	146
69	148	162	162	162	139
70	164	162	X	146	146
71	148	162	148	157	139
72	164	162	164	157	146

$\Rightarrow$ Error concealment method: Mean of well received pixels $\Rightarrow$

	2	3	4	5	6
68	164	162	164	155	146
69	148	162	162	162	139
70	164	162	**158**	146	146
71	148	162	148	157	139
72	164	162	164	157	146

PSNR = 78.23 dBm

(The original value was 162)

{Cristian.Duran,Vincent.Lecuire}@cran.uhp-nancy.fr

Pixel interleaving for robust image transport

Typical effects of packet losses

Normally, several pixels are lost per each lost packet

⇒ Error concealment method ⇒

(PSNR = 58.03 dB)

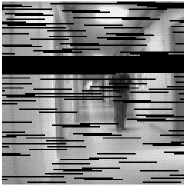
Pixel interleaving for robust image transport

Typical effects of packet losses

In a real scenario, packet losses can reach a 40% or even more

Original image

⇒



Received raw image with 29% of data losses

⇒

Reconstructed image after pixels averaging (PSNR = 25.63 dB)

What to do?

Pixel interleaving for robust image transport

Traditional error control methods

Traditional techniques for correction of errors like FEC or ARQ can be very expensive in terms of resource consumptions.

Method	Energy	Time
No ARQ	2307 mJ	29.55 sec
ARQ	3690 mJ	48.95 sec

* With no losses

In the presence of losses, these results can be greatly increased.

Pixel interleaving for robust image transport

Pixel interleaving principles

Pixel interleaving

- For each position pixel (x, y) calculate a new position (x', y')
- If we loss one packet, losses pixels wont be neighbors

$I_{0,0}$	$I_{0,1}$	$I_{0,2}$	$I_{0,3}$	...
$I_{1,0}$	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	...
$I_{2,0}$	$I_{2,1}$	$I_{2,2}$	$I_{2,3}$	...
$I_{3,0}$	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	...
...				

$\Rightarrow$

$I_{0,0}$	$I_{2,1}$	$I_{0,2}$	$I_{2,3}$	...
$I_{1,2}$	$I_{3,3}$	$I_{1,0}$	$I_{3,1}$	...
$I_{2,0}$	$I_{0,1}$	$I_{2,2}$	$I_{0,3}$	...
$I_{3,2}$	$I_{1,3}$	$I_{3,0}$	$I_{3,3}$	...
...				

$\Rightarrow$

X	$I_{0,1}$	X	$I_{0,3}$	...
$I_{1,0}$	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	...
$I_{2,0}$	X	$I_{2,2}$	$I_{2,3}$	...
$I_{3,0}$	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	...
...				

Pixel interleaving for robust image transport
Pixel interleaving principles

Pixel interleaving

- For each position pixel (x, y) calculate a new position (x', y')
- If we loss one packet, losses pixels wont be neighbors

$I_{0,0}$	$I_{0,1}$	$I_{0,2}$	$I_{0,3}$	...
$I_{1,0}$	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	...
$I_{2,0}$	$I_{2,1}$	$I_{2,2}$	$I_{2,3}$	...
$I_{3,0}$	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	...
...				

$\Rightarrow$

$I_{0,0}$	$I_{2,1}$	$I_{0,2}$	$I_{2,3}$	...
$I_{1,2}$	$I_{3,3}$	$I_{1,0}$	$I_{3,1}$	...
$I_{2,0}$	$I_{0,1}$	$I_{2,2}$	$I_{0,3}$	...
$I_{3,2}$	$I_{1,3}$	$I_{3,0}$	$I_{3,3}$	...
...				

$\Rightarrow$

X	$I_{0,1}$	X	$I_{0,3}$	...
$I_{1,0}$	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	...
$I_{2,0}$	X	$I_{2,2}$	$I_{2,3}$	...
$I_{3,0}$	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$	...
...				

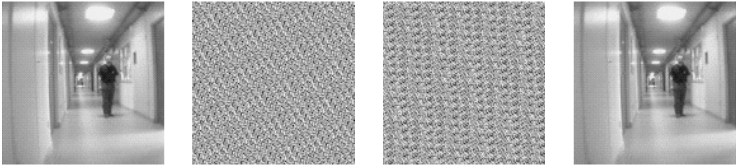
Good idea!! Problem: Which method to apply?

Pixel interleaving for robust image transport

Torus Automorphisms

Torus Automorphism

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



(a) Original image

(b) Mixed image with TA and $n = 8$

(c) Mixed image with TA and $n = 32$

(d) Mixed image with TA and $n = T = 96$

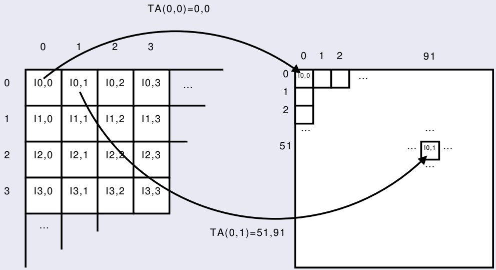
Figure: TA applied over a 128×128 'Corridor' image ($k = 1$).

{Cristian.Duran,Vincent.Lecuire}@cran.uhp-nancy.fr

Pixel interleaving for robust image transport

Torus Automorphisms

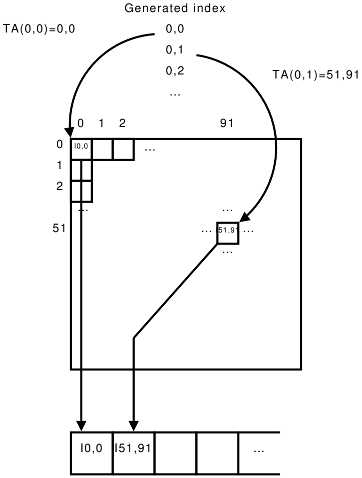
Problem with classical implementation



- We need additional memory to store the mixed image
- TA transform must be completed before starting the packetization process.

Experimentation and analysis results

Adapting TA to camera sensor nodes



Adapted TA-based pixel interleaving

$i \Leftarrow 0$ {position of data in packet}
$H \Leftarrow ImageHeight$, $W \Leftarrow ImageWidth$
for $y = 0$ to $H - 1$ **do**
  for $x = 0$ to $W - 1$ **do**
    Calculate (x', y') of position (x, y) using TA
    $Packet.data[i] \Leftarrow I[x', y']$
    if (packet is full) or $((x, y) = (W - 1, H - 1))$ **then**
      Send *packet*
      $i \Leftarrow 0$
    else
      $i \Leftarrow i + 1$
    end if
  end for
end for

Experimentation and analysis results
Experimental platform

Figure: Experimental topology

Experimentation and analysis results
Experimental platform

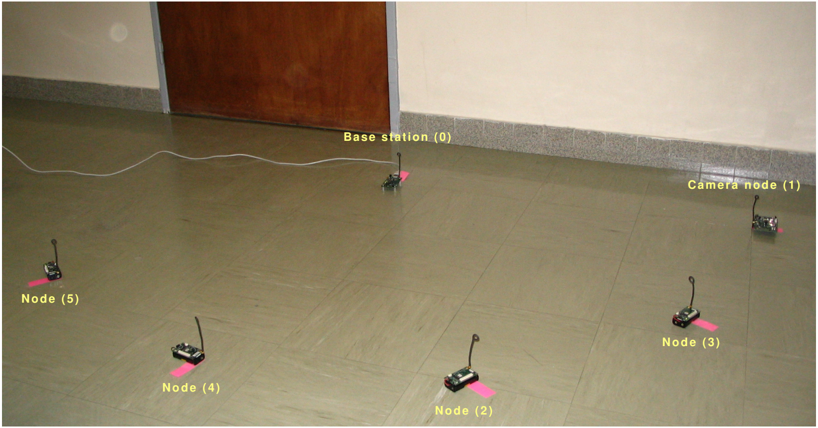


Figure: Experimental topology

Experimentation and analysis results

Experimental platform

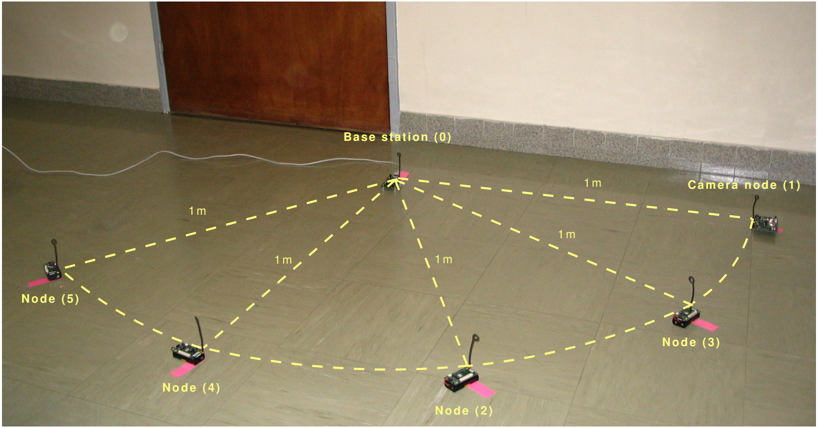


Figure: Experimental topology

Experimentation and analysis results

Experimental platform

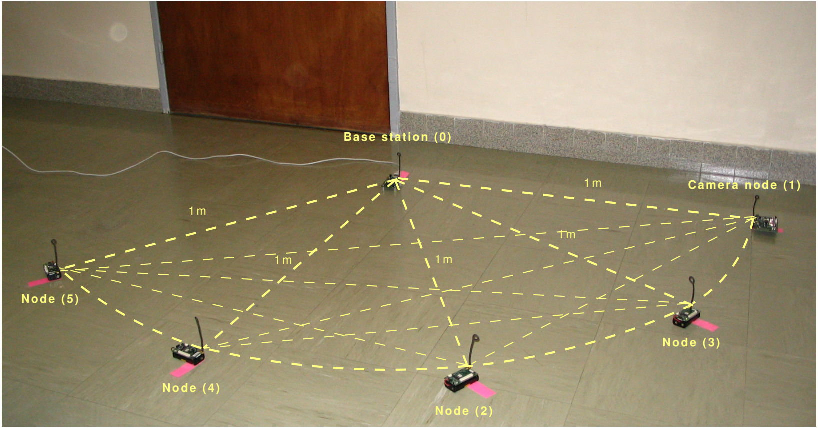


Figure: Experimental topology

Experimentation and analysis results

Implementation details

- Implementation in *TinyOS 1.x*
- Data payload of 29 bytes composed of a 2-byte header and 27 bytes to send image data
- For TA:
    - Small values for n and k (we chosed $k = 1$ and $n = 8$)
    - We inserted the pre-calculated matrix $A = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix}^n$

Experimentation and analysis results

Results

Energy / Time consumptions

- Transmission of a 8-bit monochrome 128 $\times$ 128 image
- Source's power out: -20dBm

Method	Energy consumption	Execution time
No processing	2307 mJ	29.55 sec
TA	2374 mJ	30.2 sec
ARQ-based	3690 mJ	48.95 sec

$\Longrightarrow$ *TA computation consumes only 4μJ and 40μs per pixel*

{Cristian.Duran,Vincent.Lecuire}@cran.uhp-nancy.fr

Experimentation and analysis results
Results

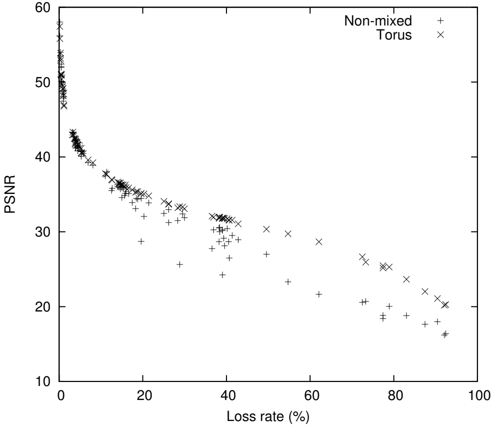


Figure: Non-mixed vs. Mixed comparison

Experimentation and analysis results
Results

Image quality visualization for a loss rate of 20.27%

No-mixed image

vs.

Mixed image

Experimentation and analysis results

Results

Image quality visualization for a loss rate of 40.18%

No-mixed image

vs.

Mixed image

Experimentation and analysis results

Results

Image quality visualization for a loss rate of 62.1%

No-mixed image

vs.

Mixed image

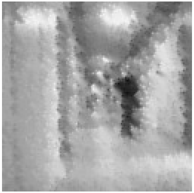
Experimentation and analysis results

Results

Image quality visualization for a loss rate of 83.03%

No-mixed image

vs.



Mixed image

Conclusion and Future work

Conclusions

- High probability to receive enough information to recover lost data.
- Good execution time and energy consumption on a wireless camera node,
- Increasing on the quality of transmitted images even with high loss rates
- No need of additional memory allocations, complex calculations, redundancy or retransmissions.

Future work

- Analysis of TA and other interleaving techniques.
- Low-complexity compression (block-based method).
- Evaluations in real multi-hop platforms.

Thank you!!

Questions??

Error Resilient Image Communication with Chaotic Pixel Interleaving for Wireless Sensor Networks

Cristian Duran-Faundez and Vincent Lecuire
{Cristian.Duran,Vincent.Lecuire}@cran.uhp-nancy.fr

Research Centre for Automatic Control (CRAN),
Nancy-University, CNRS, France