Sim-LIT: A simulation framework for image quality assessment in wireless visual sensor networks under packet loss conditions

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

† Centre de Recherche en Automatique de Nancy (CRAN)
Université de Lorraine, CNRS, Vandœuvre-lès-Nancy, France

XXX International Conference of the Chilean Computer Science Society
(SCCC'2011), Curicó, Chile

Outline

1. Introduction
2. Sim-LIT
3. Conclusion and Future Works

Wireless Sensor Networks

Definition (Wireless Sensor Network (WSN))

A large-scale distributed system normally composed of a large number of very small devices called **sensor nodes**. These sensor nodes are able to measure certain physical phenomena in the environment where they are deployed and to report its findings to one (or various) central gateway(s) (*sink*).

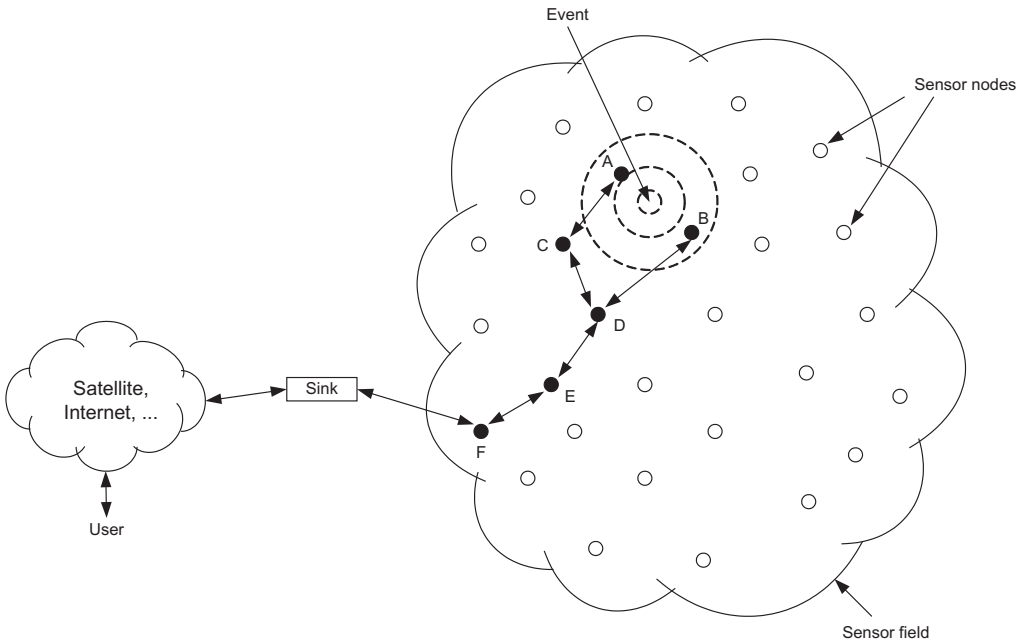


Figure: A typical wireless sensor network

Typical measurements:
Temperature, light, magnetism, pressure, vibrations, ...

Applications

- Military applications,
- Environmental applications,
- Industry, Robotics, Security, ...

Concerns

- **At the node level:** Limited energy, low processing/storage capacities, low bit rates, ...
- **At the network level:** Large-scale, high-density, dynamic topology, high loss rates, ...

Wireless Visual Sensor Networks

Definition (Wireless Visual Sensor Network (WVSN))

A WSN where one or several nodes have image sensors (cameras).

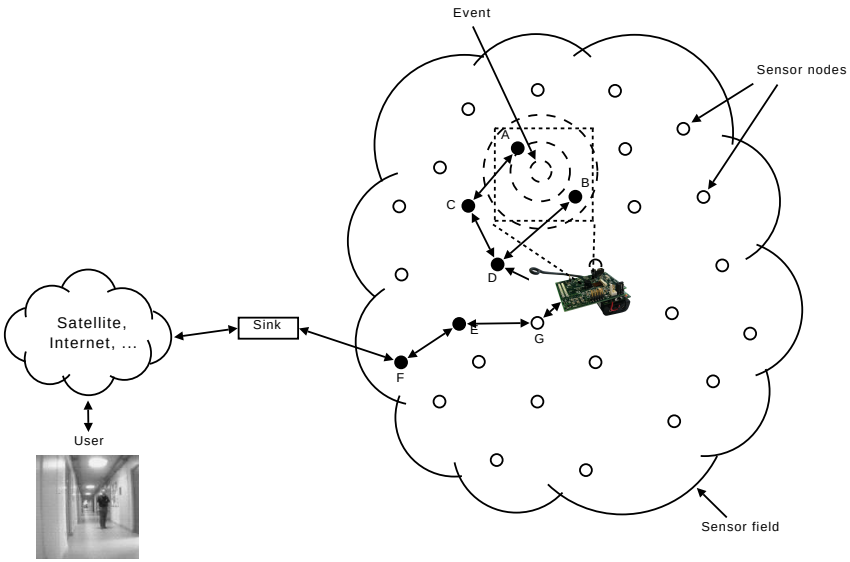


Figure: A wireless visual sensor network

Applications

- Object recognition ...
- numbering ...
- localization, tracking, ...

... of objects by vision.

Concerns

(All of the traditional WSNs)

+

A more complex (expensive) sensor

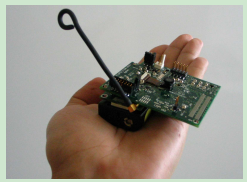
+

Large amount of data to process/transmit

Example of resource consumption [a]

[a] (using a Mica2 mote, data payload: 27 bytes, pout: -20dBm)

- Capture/Transmission of 1 scalar measure:
  - Number of packets: 1 (actually, 1 byte of the packet)
  - Energy consumption: <3 mJ
  - Time: <0.04 sec.
- Capture/Transmission of 1 (128×128) 8bpp image (+ Cyclops camera)
  - Number of packets: 607
  - Energy consumption: $\sim$2358 mJ
  - Time: $\sim$29.9 s



The effects of packet loss

- As one of the main issues in WSNs is **energy consumption**, image compression seems an obvious solution. . .

The effects of packet loss

- As one of the main issues in WSNs is **energy consumption**, image compression seems an obvious solution... but wait!! Don't forget about packet loss!!

Example (Considering JPEG image compression (3.4bpp) at the source node)


Original (128 × 128) image


1% packet loss


5% packet loss

The effects of packet loss

- As one of the main issues in WSNs is **energy consumption**, image compression seems an obvious solution... but wait!! Don't forget about packet loss!!
- In WSNs, packet loss can be important

The effects of packet loss

- As one of the main issues in WSNs is **energy consumption**, image compression seems an obvious solution... but wait!! Don't forget about packet loss!!
- In WSNs, packet loss can be important

(it can reach 40% and more)

The effects of packet loss

- As one of the main issues in WSNs is **energy consumption**, image compression seems an obvious solution... but wait!! Don't forget about packet loss!!
- In WSNs, packet loss can be important
- Non compressed images are more resistant to packet loss

Example (Considering non-compression applied and 25% packet loss)

Original (128 $\times$ 128) image

Received pixels

Reconstructed image
(Use of an error concealment method)

The effects of packet loss

- As one of the main issues in WSNs is **energy consumption**, image compression seems an obvious solution... but wait!! Don't forget about packet loss!!
- In WSNs, packet loss can be important
- Non compressed images are more resistant to packet loss
- Burst packet loss can be a problem

Example (Considering non-compression applied and 29% packet loss)

Original (128 $\times$ 128) image

Received pixels

Reconstructed image
(Use of an error concealment method)

The effects of packet loss

- As one of the main issues in WSNs is **energy consumption**, image compression seems an obvious solution... but wait!! Don't forget about packet loss!!
- In WSNs, packet loss can be important
- Non compressed images are more resistant to packet loss
- Burst packet loss can be a problem
- When we work with images over lossy environments, there is an evaluation factor to consider: loss data leads to quality losses
- Possible solutions include:
  - ACK, FEC based protocols
  - Block interleaving
  - ...

Simulation of WVSNs

- Simulation is, many times, required for validation
- Few developments on simulation environments consider particular problemas of WVSNs
- Many adapt standard network or WSNs simulators (eg., OMNet++, Castalia, . . .)
- Most of the times, adaptations over Matlab are used
- Particular example oriented to WVSNs: WISE-MNet [Nastasi and Cavallaro, 2011] (not focused on image quality issues)

C. Nastasi, A. Cavallaro (2011). "WiSE-MNet: an experimental environment for Wireless Multimedia Senso Networks". In : *SSDP'2011*.

Proposed framework

Sim-LIT (*Simulator for Lossy Image Transmission*)

- It is a simulation framework oriented to the analysis of images transmission schemes over lossy environments (such as WSNs)
- It allows image quality assessments considering:
  - image processing schemes at the source node (e.g. block interleaving), packet loss (during transmission), post-processing schemes at the decoder (e.g. error concealment)

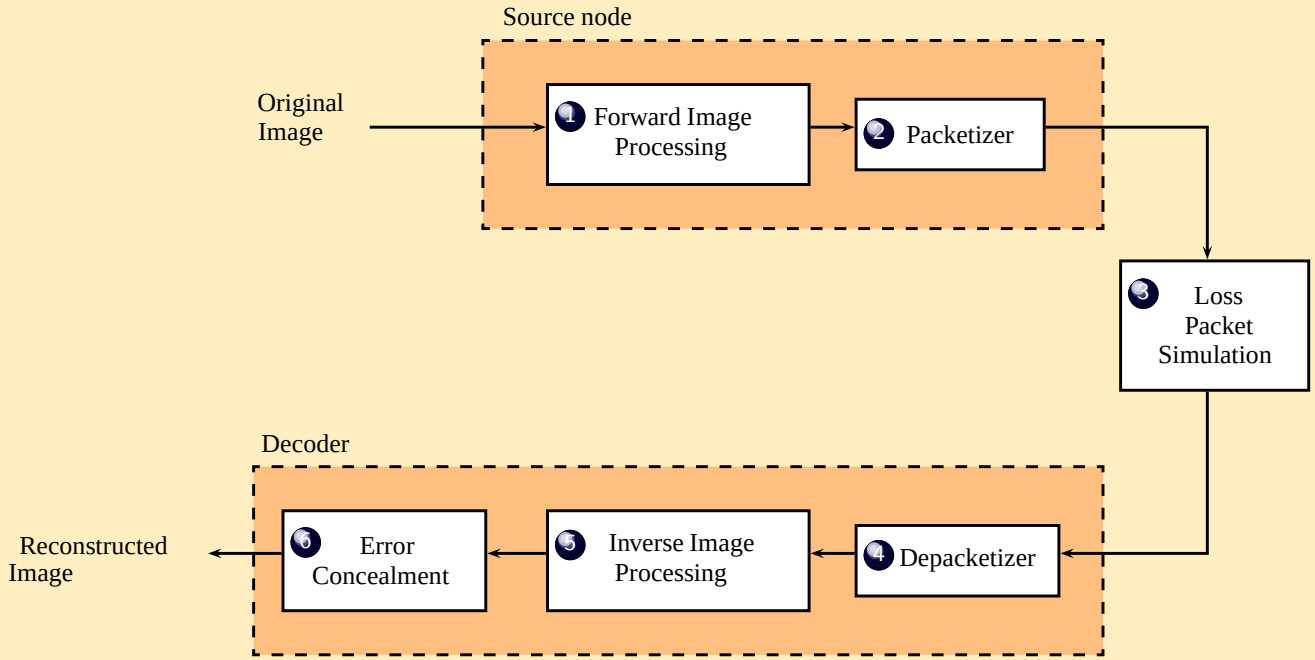
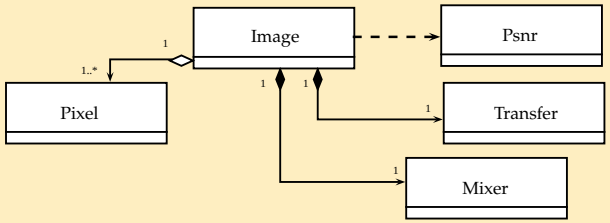


Figure: Simplified Sim-LIT's simulation scheme.

Sim-LIT features

Implementation details

- Object-oriented programming
- C++ language, g++ compiler
- GNU/Linux



Implemented Models

- Generic simulation:
  - An image is a $H \times W$ matrix, $I = \{I_{r,c}\}$, each $I_{r,c}$ containing b bpp
  - Packetization of I in $q = \lceil \frac{H \times W \times b}{m} \rceil$ packets, m = number of bits of image data per packet
  - Communication scheme $\overleftrightarrow{\Gamma}$, of P packets, where each packet has a probability p_l of being lost
  - Error concealment by averaging correctly received neighboring pixels
- Block interleaving
  - Bijective function $\vartheta : I \rightarrow I'$, where I' is a new bitmap with each block $B_{i,j} \rightarrow B'_{i',j'}$
  - Adapted interleaving considering $B'_{i,j} \leftarrow B_{i',j'}$

UNIVERSIDAD DEL BÍO-BÍO
CRAN

Quality assessment

- Subjective assessment (direct visualization)
- Objective assessment (PSNR: Peak Signal-to-Noise Ratio)
  - $PSNR = 10.\log_{10}\left(\frac{255^2}{MSE}\right)$
  - where MSE is the Mean Squared Error: $MSE = \frac{1}{H.W}\sum_{r=0}^{H-1}\sum_{c=0}^{W-1}||I_{r,c} - I''_{r,c}||^2$

Example (with a 128 × 128 8bpp grayscale image)


Original image


PSNR = 29.01 dB


PSNR = 27.56 dB

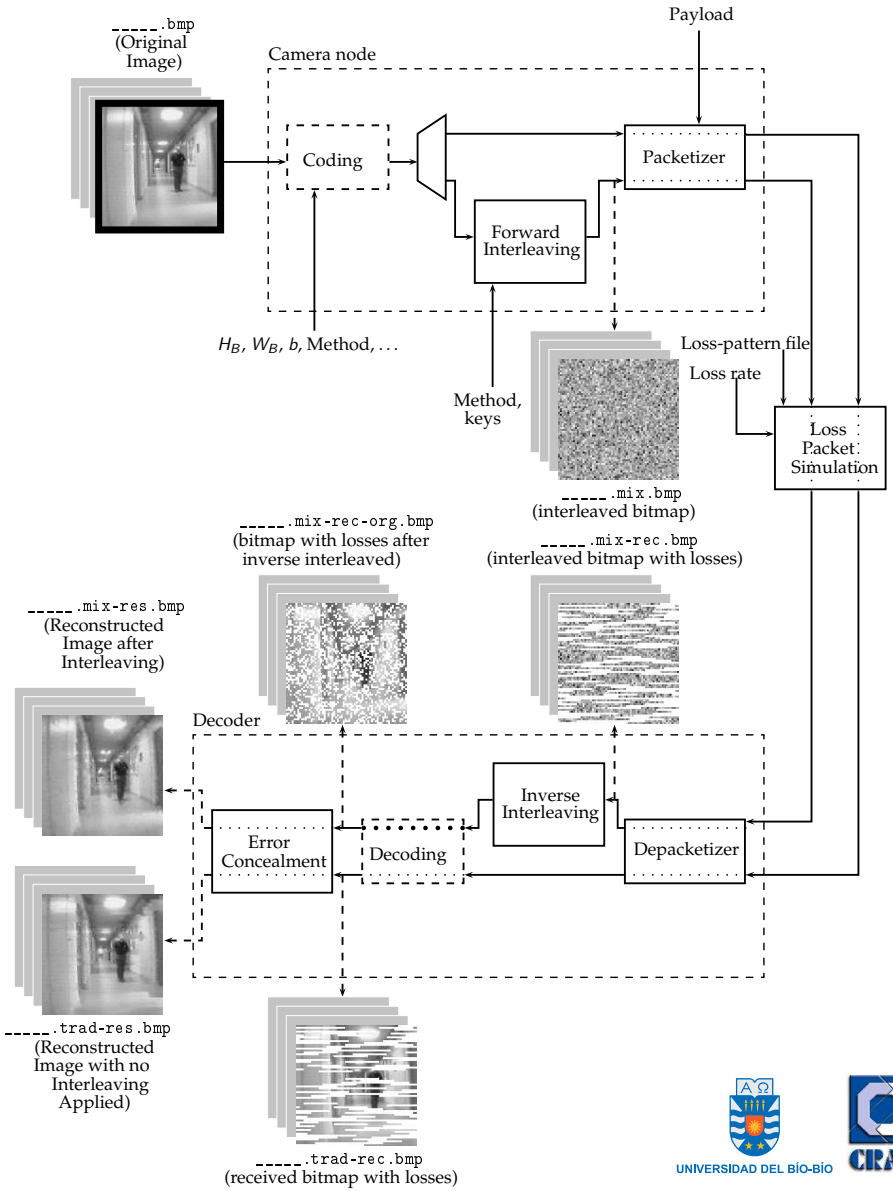

PSNR = 26.17 dB

Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

Figure: Original image

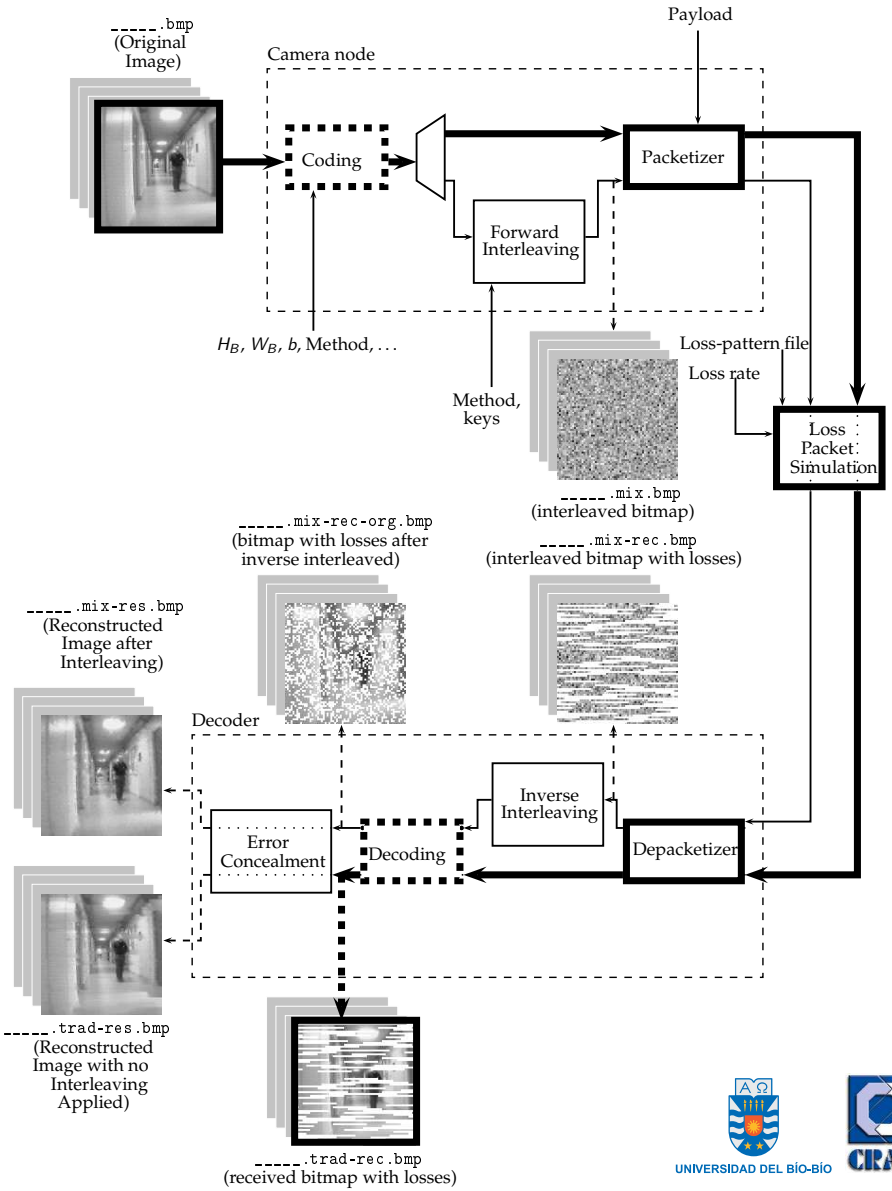


Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

Figure: Received image - S.T.

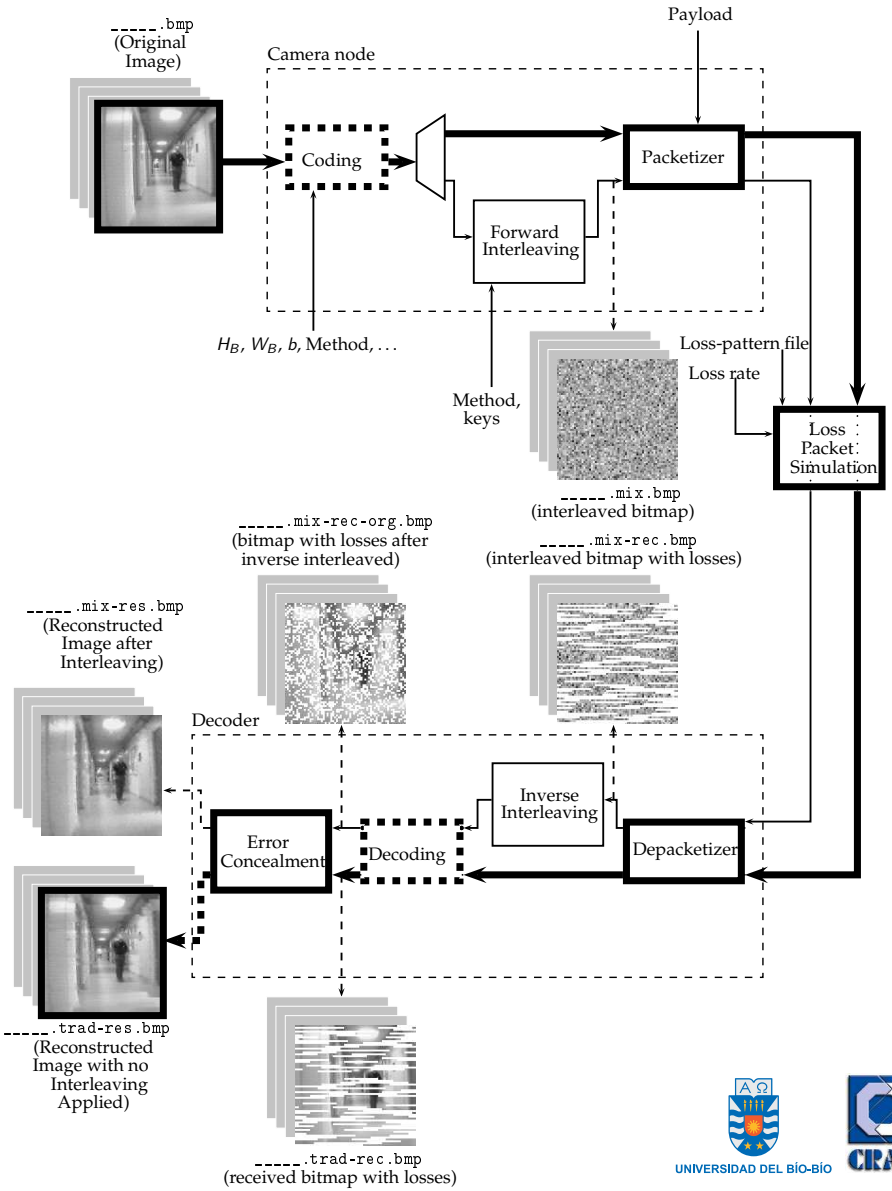


Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

Figure: Reconstructed image - S.T.

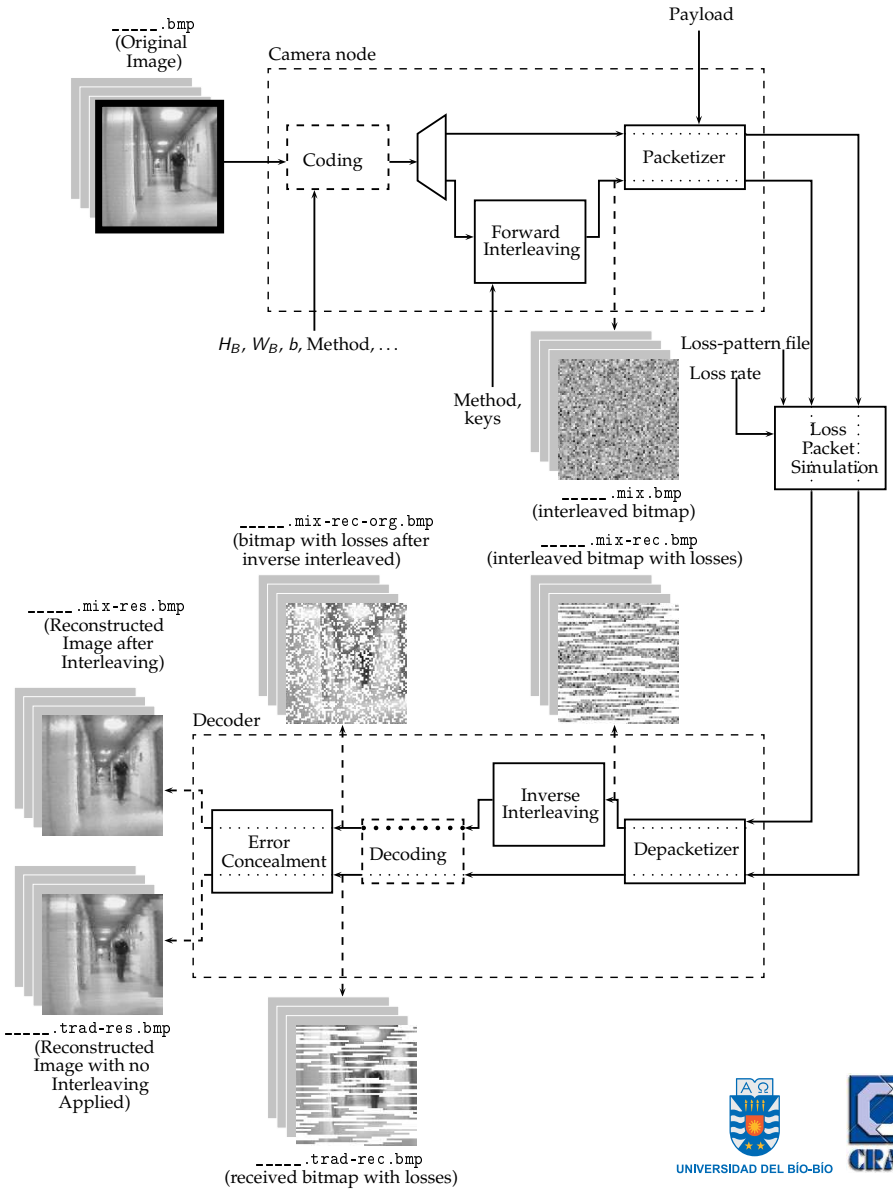


Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

Figure: Original image



Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

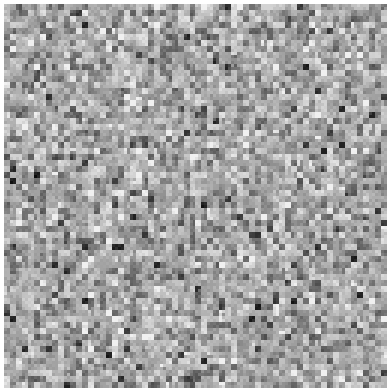
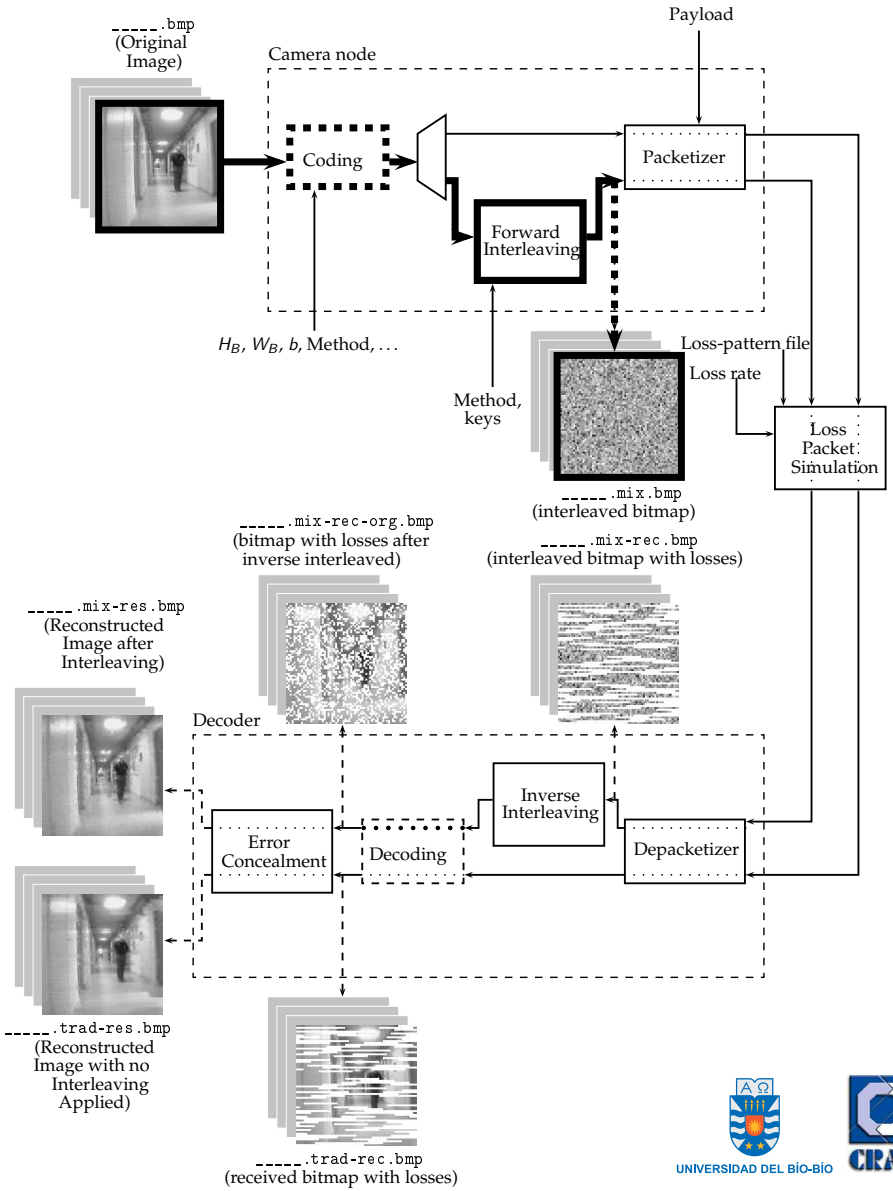


Figure: Interleaving image - I.T.

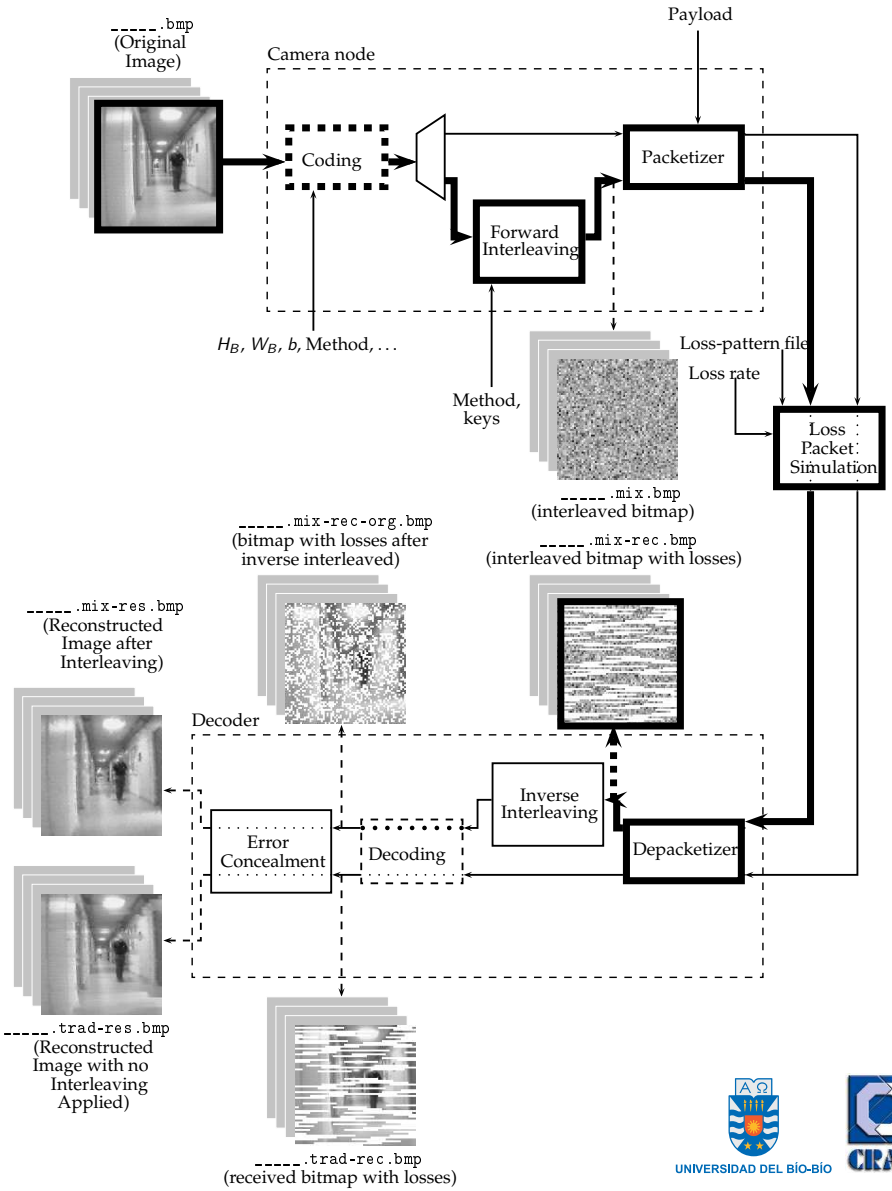


Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

Figure: Received image - I.T.



Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

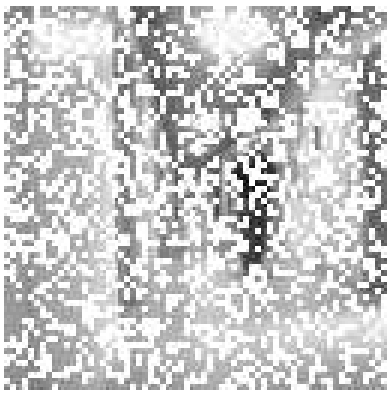
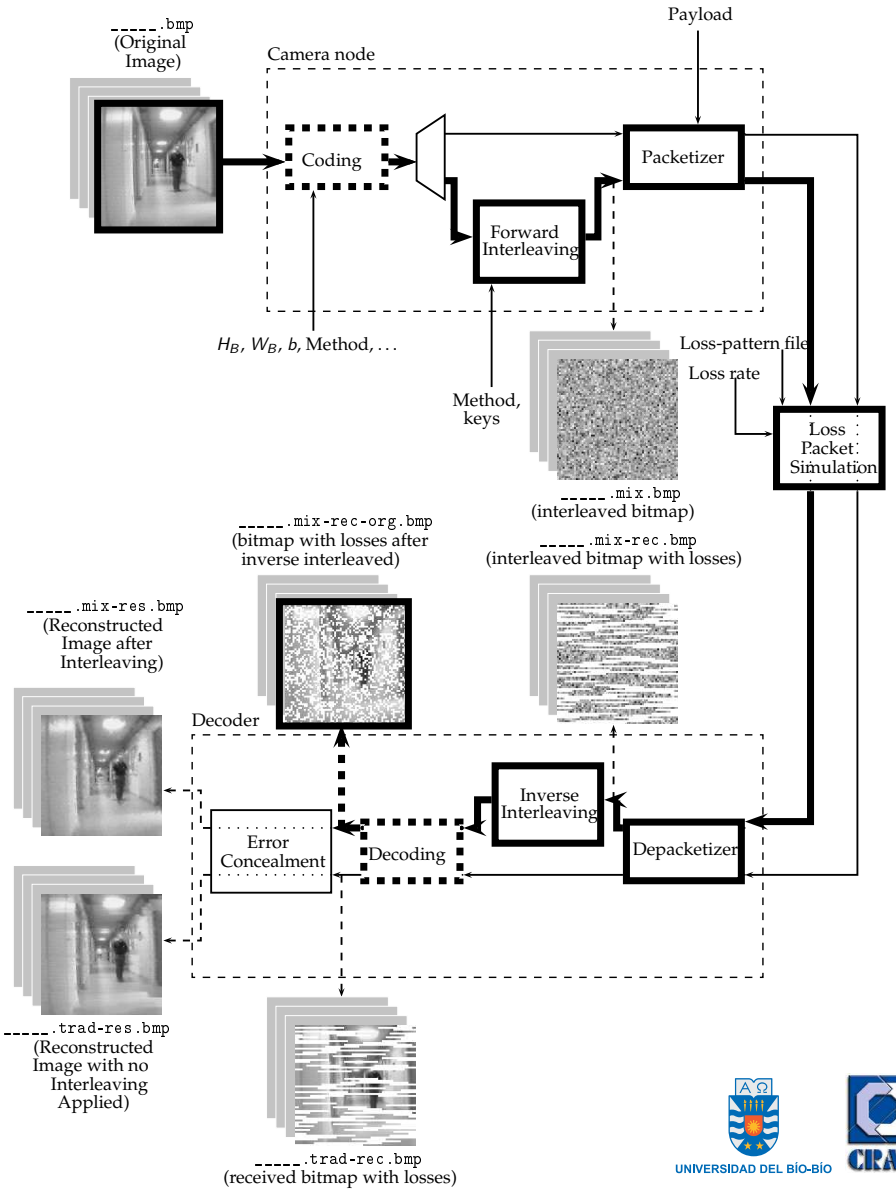


Figure: De-Interleaving image - I.T.

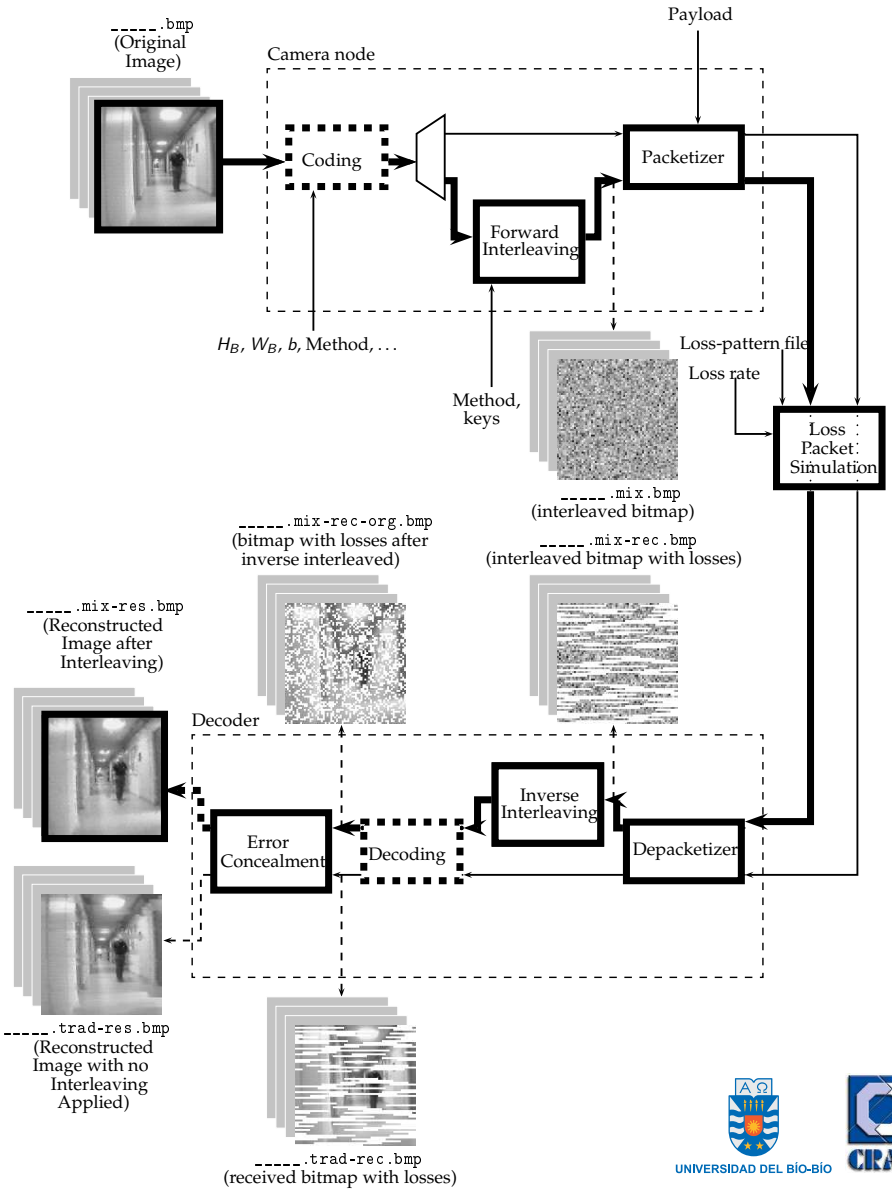


Sample execution

Example of an execution line

```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

Figure: Recontructed image - I.T.



Sample execution

Example of an execution line

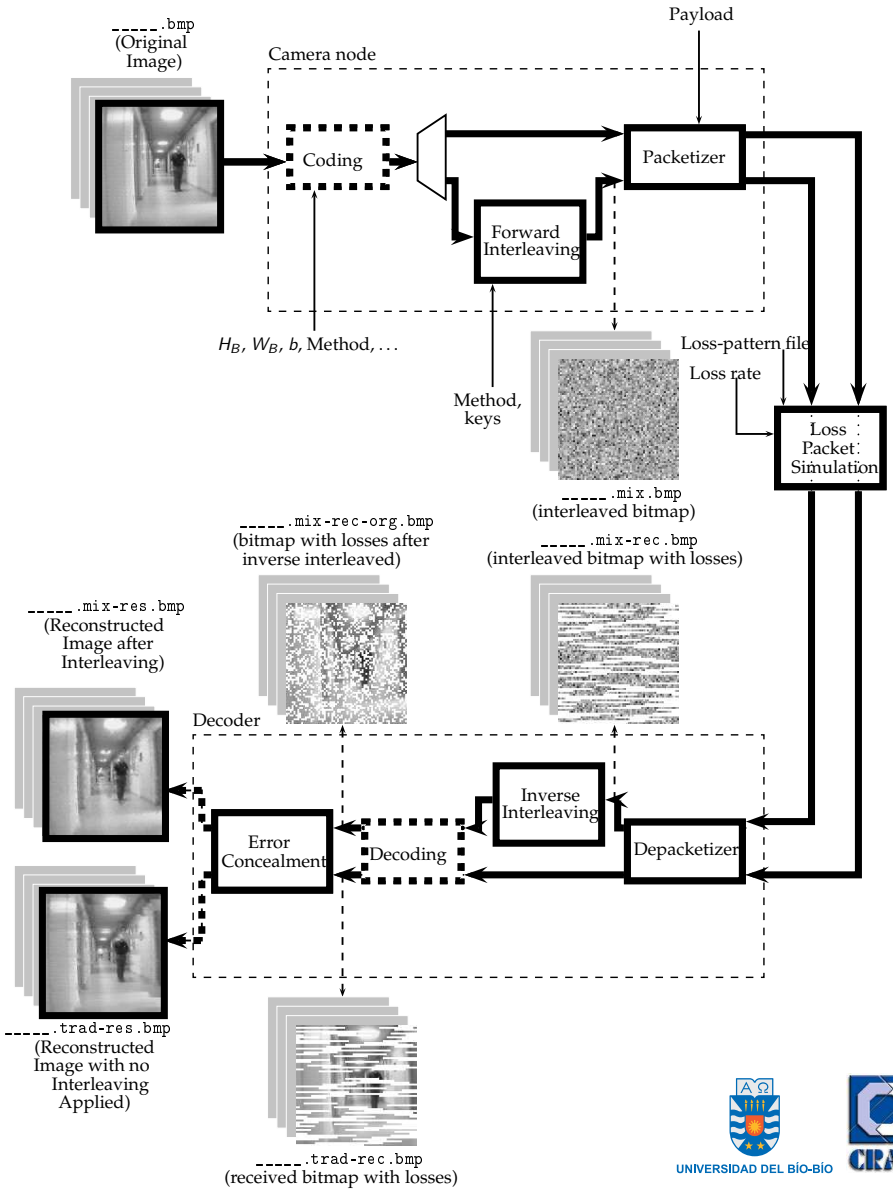
```
$./simlit -ipath /sample/ -csf results -s
27 -hb 2 -wb 2 -b 3 -loss-file
/sample/loss-file_40_1.in -torus-mixer 1
8 -adapted-interleaving
-simulation-rep-file-data-graphic
```

Figure: S.T. (PSNR=27.43dB)

Figure: I.T. (PSNR=30.67dB)

Data output files:

`__.txt`	Simulation report
`__-data.dat`	Simulation report data
`__-nmixdata.dat`	Detail. PSNR vs. loss rates / non-mixed images
`__-mixdata.dat`	Detail. PSNR vs. loss rates / mixed images



Options and complementary tools

Principal options

`-adapted-interleaving`	Interleaving technique during packetization
`-b`	Bits per each pixel during transmission
`-csf`	Change the results folder
`-data-graphic`	Creates files to build graphics of 'Loss-Rate' v/s 'Psnr' for each image, under different loss-rate
`-hb/-wb`	Height/Width block size input
`-help`	Help information
`-ipath`	Work directory
`-loss-file`	Defines loss-file input
`-loss-path`	Defines the path of loss files (For multiple simulation)
`-loss-rate`	Defines loss rate for randomly packet loss
`-no-console-messages`	Do not display console messages
`-no-res-img-store`	Do not create resulting images
`-s`	Payload packet
`-simulation-rep-file`	Output summary data files
`-mohsen-mixer`	Interleaving scheme presented in "An Efficient Chaotic Interleaver for Image Transmission over IEEE 802.15.4 Zigbee Network"
`-torus-mixer`	Interleaving scheme presented in "Error Resilient Image Communication with Chaotic Pixel Interleaving for Wireless Camera Sensors"

UNIVERSIDAD DEL BÍO-BÍO

CRAN

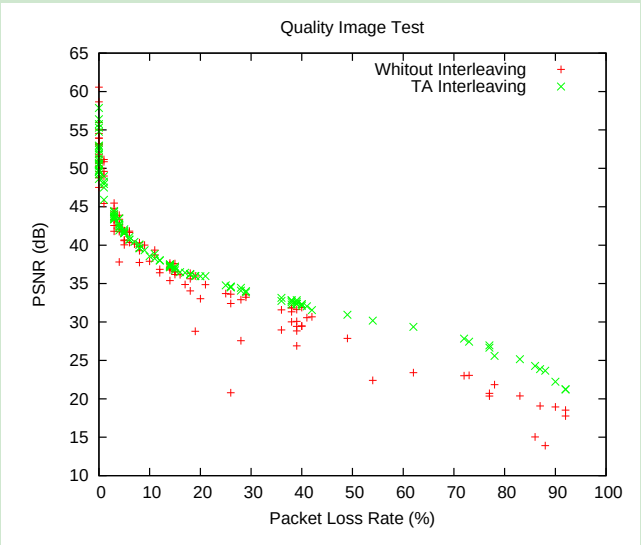
Options and complementary tools

Complementary tools

`genPackLoss` (loss generator) allows to create loss-pattern files both user defined or randomly.

1	1
2	1
3	0
.	.
.	.
.	.

`graphic` Plotter of data graphics, allow graphically display of generated data by Sim-LIT through `-data-graphic` option



Evaluation

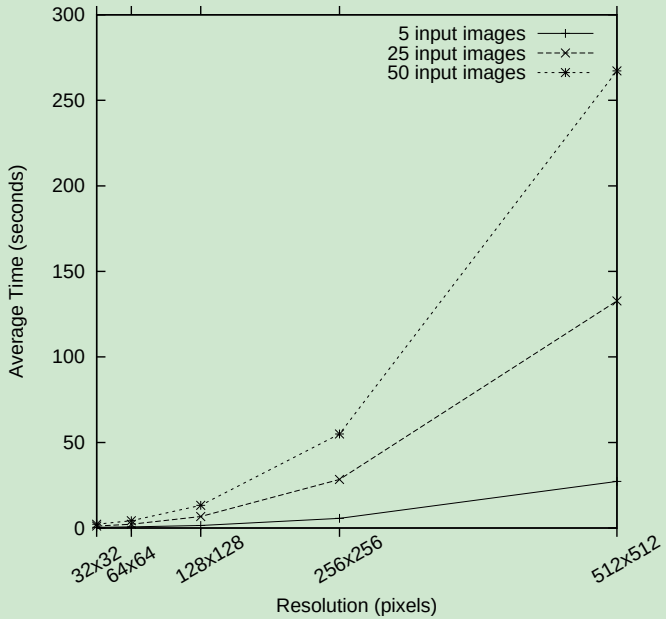
Evaluation parameters

- Used machine
  - Processor: AMD Phenom II x4 955
  - RAM: 6 GB
  - O.S.: GNU/Linux Debian (Squeeze) kernel release 2.6.32-5-amd64
- Measurements collected with the Linux command `time`
- Simulations considered T.A. interleaving scheme applied in [Duran-Faundez and Lecuire, 2008]

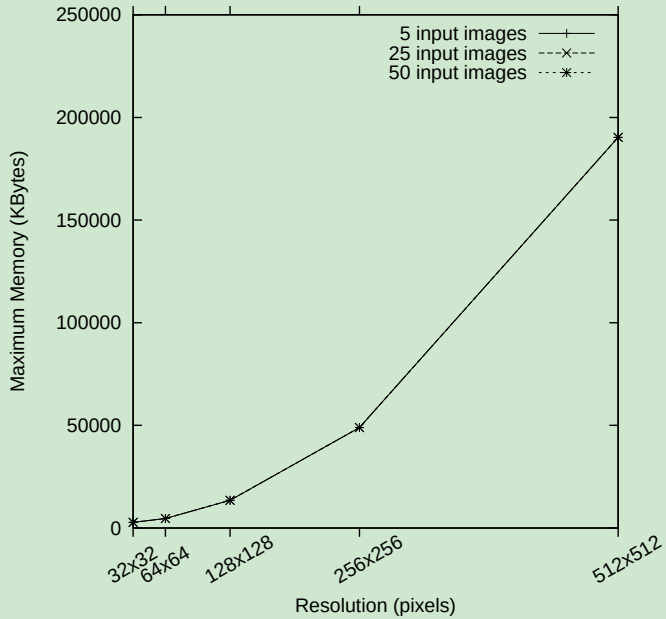
C. Duran-Faundez, V. Lecuire (2008). "Error resilient image communication with chaotic pixel interleaving for wireless camera sensors". In : *REALWSN'2008*.

Evaluation

Example (Performance of simulations)



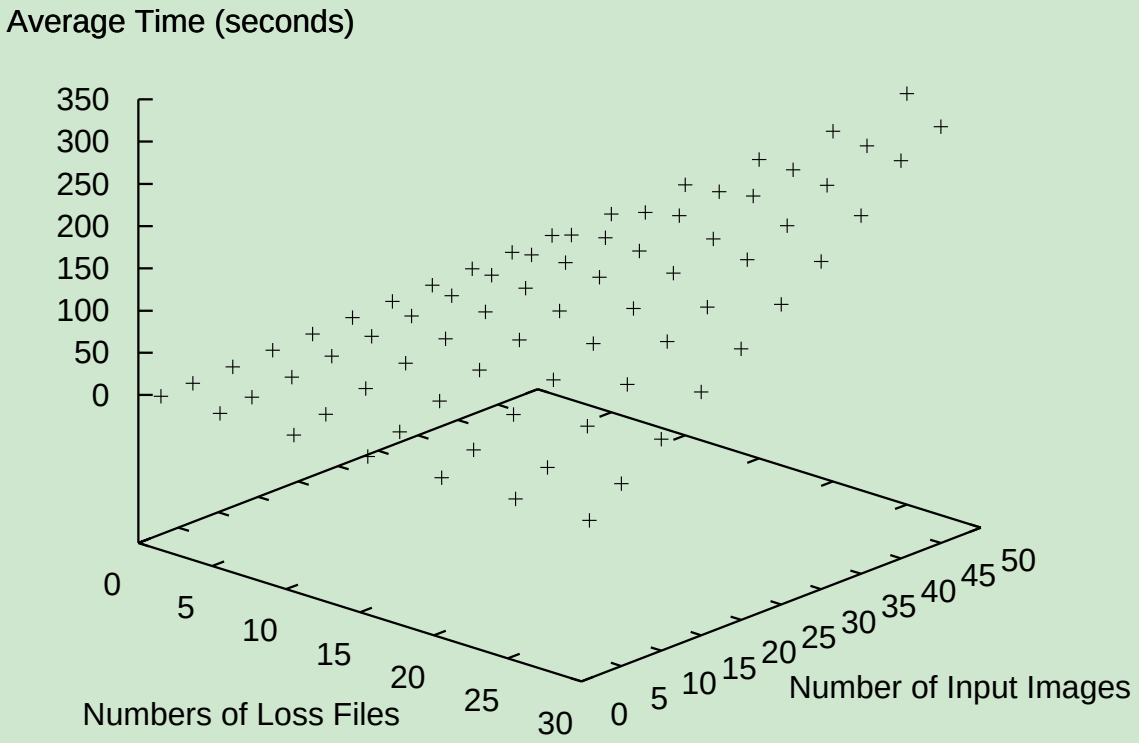
Execution Time



Memory Use

Evaluation

Example (Performance of simulations)



Conclusion and Future Works

Conclusions

- Simulation framework oriented to image quality assessments
  - Oriented-object C++ Programming
  - For now, only `.bmp` support
  - Block interleaving included in the first version
  - Loss models depend on the constructed loss files (it is possible to connect with WSNs simulators)
- Currently, we use Sim-LIT to evaluate existing and new interleaving schemes (possible to find optimal?)
- Available as open-source:
  `http://pegasus.dci.ubiobio.cl/~crduran/software/simlit`

Future works

- Parallel programming
- Incorporation of other error robust methods
- Incorporation of other quality metric measurements
- Other improvements

UNIVERSIDAD DEL BÍO-BÍO

Sim-LIT: A simulation framework for image quality assessment in wireless visual sensor networks under packet loss conditions

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
† Centre de Recherche en Automatique de Nancy (CRAN)
Université de Lorraine, CNRS, Vandœuvre-lès-Nancy, France

XXX International Conference of the Chilean Computer Science Society
(SCCC'2011), Curicó, Chile