

A COLOR IMAGE HIDDEN IN A GREY-LEVEL IMAGE

Marc CHAUMONT and William PUECH

marc.chaumont@lirmm.fr and william.puech@lirmm.fr

LIRMM, UMR CNRS 5506, Université de Montpellier II



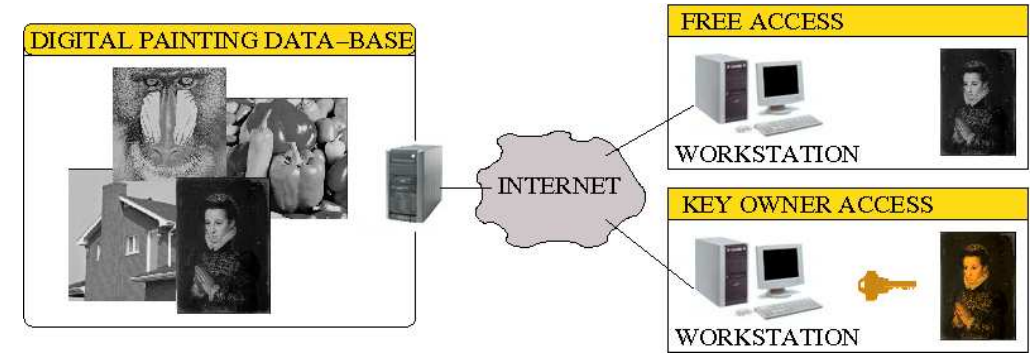
OBJECTIVE :

COLOR SECURED of image database.

- ↳ free access to grey-level images,
- ↳ key-manage access to the color information.

PRINCIPLE :

A solution based on a DATA-HIDING method where COLOR PALETTE is the message and INDEX IMAGE is the cover.



THREE MAIN STEPS:

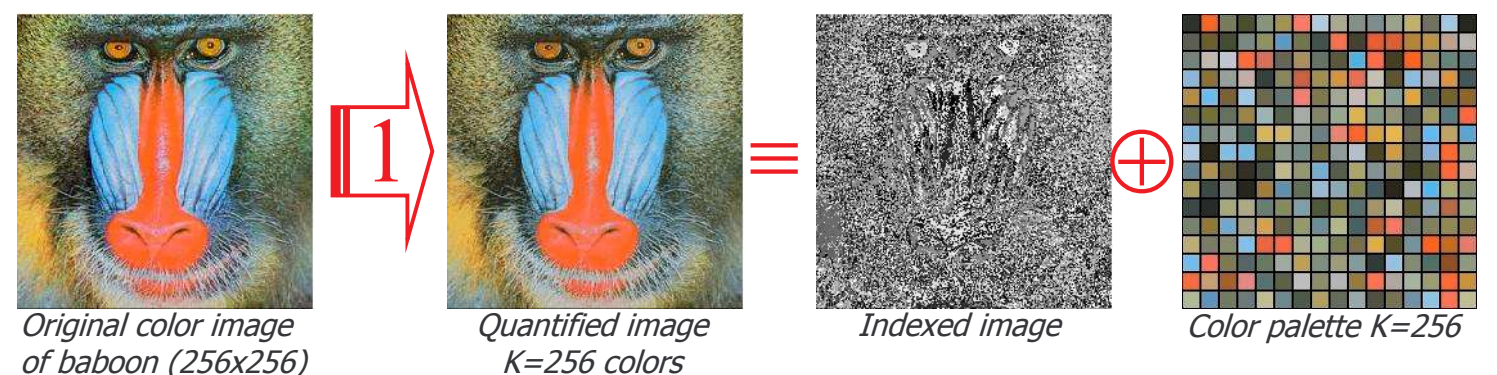
1- A fast and efficient color quantization:

Reduction of the color number to **K=256 colors**:

Two steps using :

- 1- A **fast but non optimal*** algorithm (octree color quantization algorithm)
- 2- An **optimal k-mean algorithm** (use step 1 as initialization).

* optimal in Quadratic Error



2- Optimized color ordering; the Layer Running Algorithm:

2 constraint objectives:

- Get an **indexed image closed to the luminance**,
- Get a **color palette owning closed consecutive colors**.

Mathematical model:

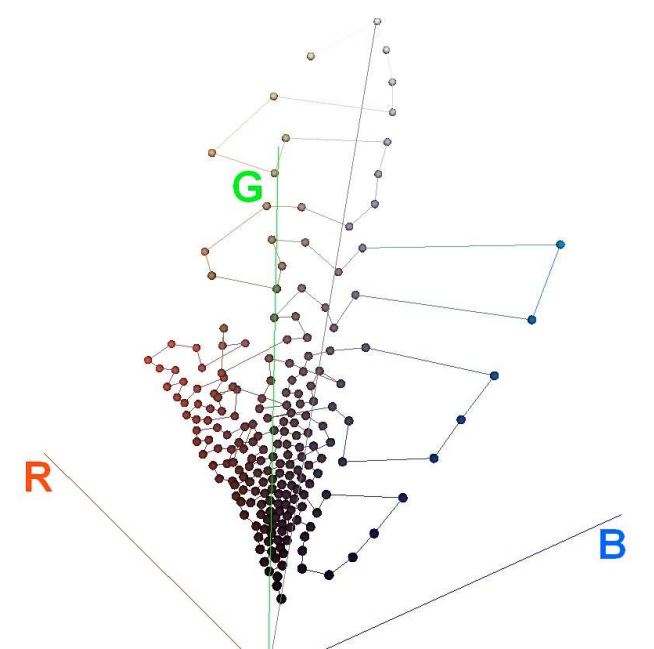
Find the best permutation function Φ such that:

$$\Phi = \arg \min_{\Phi} \sum_{i=1}^N \text{dist}^2(Y(i), \Phi(\text{Index}(i))) + \lambda \sum_{k=1}^{K-1} \text{dist}^2(\text{Palette}(\Phi^{-1}(k)), \text{Palette}(\Phi^{-1}(k+1))),$$

The layer running algorithm: find an ordering for the K colors i.e. a path in the K color graph.

To each k' node in the path, a k color is associated;

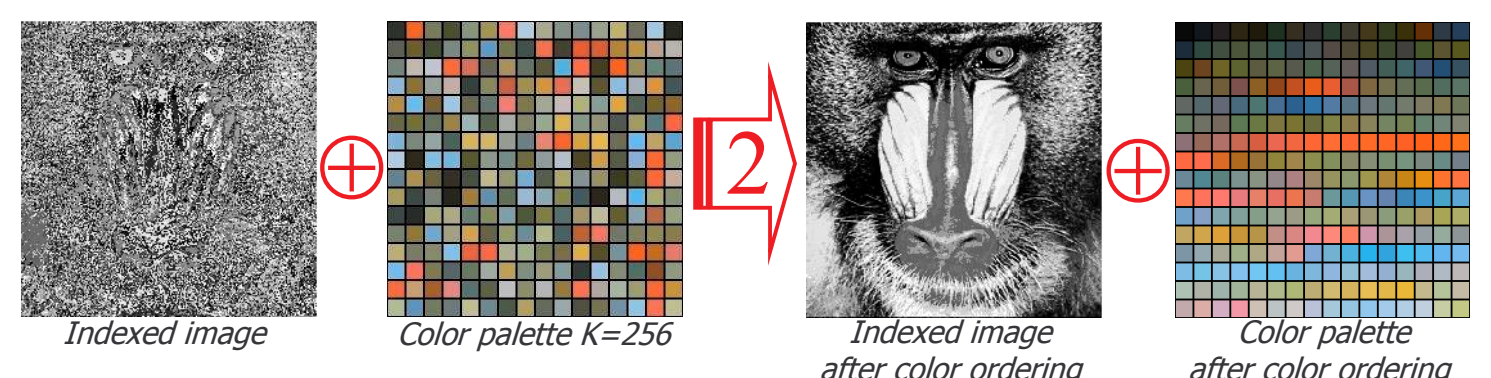
Φ is defined such that $\Phi(k) = k'$



→ Approximate thanks to an heuristic algorithm:

the layer running algorithm

Hidden parameter: the size of a layer.
Choose the size minimizing the energy.



3- Color palette compression & Data-Hiding:

1- Compression of the color palette
differential + entropy encoding

Message header

l	Min	Max	Topology	fs	frequency	frequency	...
8 bits	8 bits	(Max + lMin + 1) bits	4 bits	fs bits	fs bits	...	
Min	: minimum prediction error <= 0						
Max	: maximum prediction error >= 0						
Topology	: if bit = 1 then frequency > 0						
frequency	: appearance frequency						

2- Data-Hiding



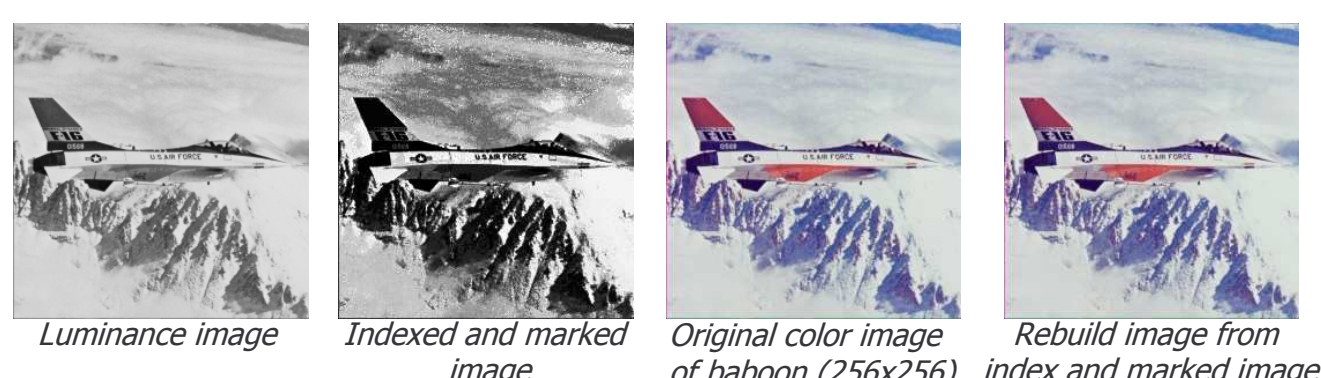
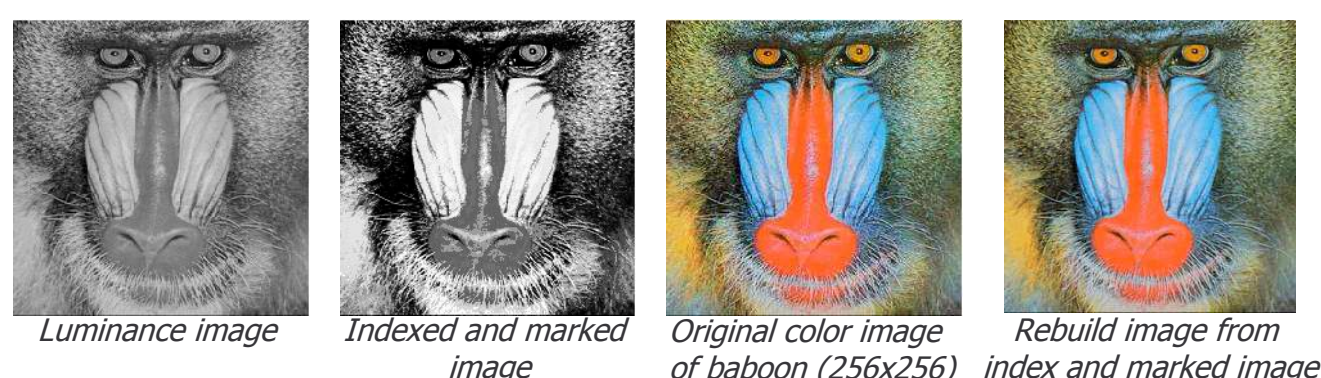
b_j : the bit to embed,

$k = \text{Index}(i)$: the index value of pixel i :

$$\text{Index}_m(i) = \begin{cases} k & \text{if } b_j = k \bmod 2, \\ \arg \min_{k' \in \{k-1, k+1\}} (\text{Palette}(k') - \text{Palette}(k))^2 & \text{otherwise.} \end{cases}$$

RESULTS & CONCLUSION:

Results:



images	PSNR _(luminance.index-marked)		PSNR _(color image.rebuild)		color palette coding cost
	previous method ¹	proposed method	previous method ¹	proposed method	
baboon	16.31 dB	16.75 dB	32.89 dB	33.31 dB	header: 693 bits source: 4417 bits sum: 5110 bits
airplane	12.95 dB	12.87 dB	38.89 dB	39.90 dB	header: 761 bits source: 3552 bits sum: 4313 bits
house	18.56 dB	18.64 dB	38.07 dB	39.27 dB	header: 728 bits source: 3833 bits sum: 4561 bits

Conclusion:

- An original **representation of an indexed image**,
- An original K color ordering: **the layer running algorithm**,
- An efficient way to **securely hide color information in an image**.
- Improvement in complexity and in quality:
 - acceleration of the quantization,
 - layer running algorithm optimization,
 - better data-hiding,
 - color palette compression.

Acknowledgments: TSAR French Project ANR SSIA 2006-2008