A Color Image Hidden in a Grey-Level Image

M. Chaumont and W. Puech
Laboratory LIRMM, UMR CNRS 5506, University of Montpellier II
161, rue Ada, 34392 MONTPELLIER CEDEX 05, FRANCE
marc.chaumont@lirmm.fr, william.puech@lirmm.fr

Abstract

In this paper, we propose a method to embed the color information of an image in a corresponding grey-level image. The objective of this work is to allow free access to the grey-level image and give color image access only if you own a secret key. This method is made of three major steps which are the color quantization, the ordering and the data hiding. The originality of this paper is to build an indexed image *which is, in the same time, a semantically intelligible grey-level image. In order to obtain this particular* indexed image, *which should be robust to data hiding, we propose an original K color ordering algorithm called the* layer running algorithm.

Introduction

Nowadays, only few secure solutions are proposed in order to give both a free access to low-quality images and a secure access to the same images at an higher quality. Our solution is built on a data-hiding method. The image may be freely obtained but its high quality visualization requires a secret key. More precisely, in our solution, a grey-level image is freely accessible but only secret key owners may rebuild the color image. Our aim is thus to protect the color information by embedding this information in the grey level image. Note that this work is though to be used to give a limited access to the private images data-base of the Louvre Museum of Paris, France.

The proposed method is made of three major steps which are the *color quantization*, the *ordering* and the *data hiding*. In the color quantization step, the aim is to found K colors and assigned to each pixel, one of those K colors. In the ordering step, the aim is to organize those K colors in order to build a palette of chromatically regular color and an *indexed image* which is semantically intelligible. In the data hiding step, the aim is to embed the color palette in the *indexed image*. The originality of this paper is to build an *indexed image* which is, in the same time, a semantically intelligible grey-level image. In order to obtain this particular *indexed image*, which should be robust to data hiding, we propose an original K color ordering algorithm called the *layer running algorithm*.

No other previous similar works have been proposed in this field. One could reference solutions such that Wu *and al* who propose to build a new palette in order to embed one message bit into each color of the palette [8], but no one embed the palette in the *indexed image*. In a first part, we describe the three steps of the proposed method. In a second part, we have applied our method on various color images and analyzed the results.

The proposed method

Color quantization

Reducing the color number of a color image is a classical quantization problem. The optimal solution, to extract the K colors, is obtained by solving:

$$\{P_{\imath,k},C(k)\} = \arg\min_{P_{\imath,k},C(k)} \sum_{\imath=1}^{N}\sum_{k=1}^{K} P_{\imath,k}.dist^2(I(\imath),C(k)), \quad (1)$$

where I is a color image of dimension N pixels, $C(k)$ is the k^{th} color of the research K colors, $dist$ is a distance function in the color space (L2 in the RGB color space), and $P_{\imath,k} \subset \{0,1\}$ is the membership value of pixel $\imath$ to color k.

A well known solution to minimize the Equ. (1), and then to obtain the K colors, is to use the ISODATA k-mean clustering algorithm [1]. $P_{i,k}$ is defined as:

$$\forall i,\forall k, P_{i,k} = \begin{cases} 1 \text{ if } k = \arg\{\min_{\{k'\}} \; dist(I(i),C(k'))\}, \\ 0 \text{ else,} \end{cases} \quad (2)$$

with $C(k) = \frac{\sum_{i=1}^{N} P_{i,k}\times I(i)}{\sum_{i=1}^{N} P_{i,k}}$.

Nevertheless, in our approach the K number is significant in comparison to the original number of color. If we proceed with a classical k-mean algorithm, the number of color extracted will often be below K. Indeed, it is the well known problem of *death classes*. To overcome that problem, one could initialized the $P_{\imath,k}$ values by solving the fuzzy c-mean equation below:

$$\{P_{\imath,k},C(k)\} = \arg\min_{P_{\imath,k},C(k)} \sum_{\imath=1}^{N}\sum_{k=1}^{K} P_{\imath,k}^{m}.dist^2(I(\imath),C(k)), \quad (3)$$

where m is the fuzzy coefficient (m is set to 1.6 as proposed in [3]) and $P_{\imath,k} \in [0,1]$ are fuzzy membership values. This equation is solved by a fuzzy c-mean algorithm [4].

Layer running

Once the color quantization has been processed, the obtained K color image, could be represented by an *indexed image* (thanks to $P_{\imath,k}$ values) and a color palette (thanks to $C(k)$ values). Our goal is to solve two constraints; the first constraint is to get an *indexed image* where each grey-level is not too far from the luminance of the original color image; the second constraint is that in the color palette, two consecutive colors should not be far distant. Thanks to the color quantization, we already own an *indexed image* and a color palette. Our problem is then to find a permutation function which permutes in the same time the values of the *indexed image* and the values of the color palette. The best permutation function Φ is found by solving:

$$\Phi = \arg\min_{\Phi} \sum_{\imath=1}^{N} dist^2(Y(\imath),\Phi(Index(\imath))) + \lambda \sum_{k=1}^{K-1} dist^2(Palette(\Phi^{-1}(k)),Palette(\Phi^{-1}(k+1))), \quad (4)$$

where Y is the luminance of the original color image, and λ is the Lagrangian value. The Φ permutation function is a bijective function in $\mathbb{N}$ defined such that $\Phi : [1..K] \rightarrow [1..K]$.

In a first approximation, the Equ. (4) is solved thanks to an heuristic algorithm. The aim of this algorithm is to find an ordering for the K colors such that consecutive colors are not far distant and such that colors are ordered from the darkest to the lightest. This ordering defines for each k^{th} color a k' position which gives us the Φ function such that $\Phi(k) = k'$.

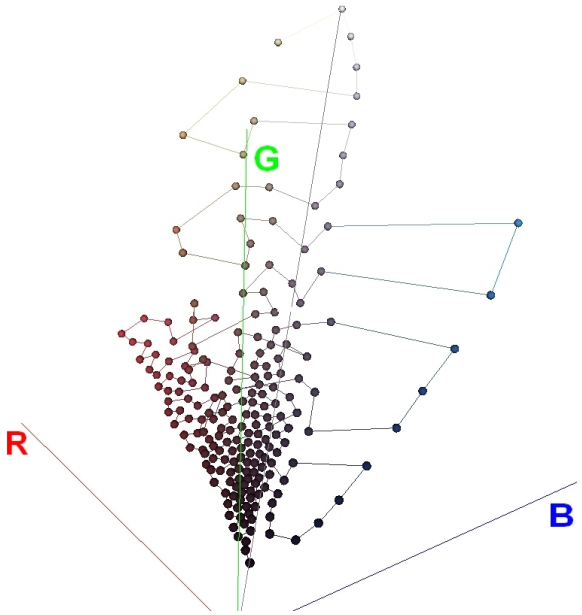


Figure 1. *A view of the layer running in the RGB cube.*

To find an ordering of the K colors, the algorithm (see Listing 1) runs the color space to build the ordered suite of colors, illustrated Figure 1. This running is obtained by jumping from color to color, into the color space, by choosing the closer color from the current one. The first color of this suite is chosen as the darkest one among the K colors. An additional constraint to this running is that we limit the research of colors to colors which are not too far in luminance. This signify that the running in the color space is limited to a layer defined on luminance information. This *layer running algorithm* could then be seen as a kind of *3D spiral run* in the color space.

Spatial data hiding method

The methods in spatial domain embed directly the information into the pixel of the original image. The first techniques embedded the bit message in a sequential way in the LSB (Low Significant Bit) of the pixel image [2, 6]. They have been improved by using a PRNG (Pseudo-Random Number Generator) and a secret key in order to have private access to the embedded information. The PRNG spreads over the image the message and makes hard the steganalyses [5]. Although those spatial hiding methods are not robust against attacks, they enable to embed a great amount of information. The embedding process can be made in the LSB1, LSB2 or even in more significant bits such as SB4 [7] by using the fourth bit of the pixel image.

For this paper, we have used an algorithm to embed the color palette information in the LSB1 of the image of N pixels. The objective is thus to embed a message M made up of m bits b_i

Listing 1. *Layer running algorithm*

```
Integer LAYERSIZE;     //Thickness of the layer.
Array luminance;       //Associates a grey-level to a node
Integer nbNode;        //Number of nodes (path size).

Procedure LayerRunningAlgorithm():List
begin

  Set layer;           //A set of nodes.
  List path;           //A sequence of consecutive nodes.
  Node currentNode;    //The current node.
  Node minLumNode;     //Minimum luminance node.

  //THE CURRENT NODE IS SET TO x0
  currentNode ← x0;

  //THE ORIGIN OF THE PATH IS SET TO x0
  path ← currentNode;

  //COMPUTE THE MINIMUM LUMINANCE NODE
  minLumNode ← arg_node {node|(luminance[node] ≥ luminance[x0]
                        and node ∉ path)};

  //COMPUTE THE LAYER (A SET OF NODE)
  layer ← arg_{node} {{node}|(luminance[node] ≥ luminance[minLumNode]
          and luminance[node] ≤ luminance[minLumNode] + LAYERSIZE
          and node ∉ path)};

  //COMPUTE THE PATH
  for nb from 2 to nbNode
  begin

    //NEAREST NODE FROM THE CURRENT ONE IN THE LAYER
    currentNode ← arg min_{node∈layer} dist(currentNode, node);

    //ADD THIS NODE TO THE PATH
    path ← path ⊕ currentNode;

    //SUPRESS THIS NODE FROM THE LAYER
    layer ← layer − currentNode;

    //IF THE CURRENT NODE IS THE MINIMUM LUMINANCE NODE
    //- UPDATE OF THE MINIMUM LUMINANCE NODE.
    //- AND UPDATE OF THE LAYER (SLIDING WINDOW),
    if ((currentNode = minLumNode) and (nb ≠ nbNode))
    then
    begin
      //COMPUTE THE MINIMUM LUMINANCE NODE
      minLumNode ← arg_node {node|(luminance[node] ≥
                   luminance[minLumNode] and node ∉ path)};
      //COMPUTE THE LAYER (A SET OF NODE)
      layer ← arg_{node} {{node}|(luminance[node] ≥ luminance[minLumNode]
              and luminance[node] ≤ luminance[minLumNode] + LAYERSIZE
              and node ∉ path)};
    end
  end

  //RETURN PATH
  return path;
end
```

($M = b_1 b_2 ... b_m$). The embedding factor, in *bit/pixel*, is:

$$E_f = m/N. \quad (5)$$

The original image is then divided in areas of size $\lfloor 1/E_f \rfloor$ pixels. Each area is used to hide only one bit b_ι of the message. This splitting procedure guarantees that the message is spread homogeneously over the whole image. In order to hide the color palette in the image we need to embed $m = 3 \times 254 \times 8 = 6096$ bits (the number of colors is $K = 254$) in the *indexed image*. Consequently, the embedding factor E_f, Equ. (5), only depends on the image size N. In our process, the PRNG selects randomly, for each region, a pixel $I(n)$. In order to get a marked pixel $I_M(n)$, the LSB of this selected pixel $I(n)$ is then modified according to the message bit:

$$I_M(n) = I(n) - I(n) \; mod \; 2 + b_\iota. \quad (6)$$

Results

We have applied our method on three color images of size 256×256 pixels: the baboon, Figure 2.a, the airplane, Figure 6.a, and the house, Figure 7.a. The baboon image owns a lot of different colors, the airplane image owns very few colors and is light-colored and the house image owns tint areas. The results obtained show that the approach is efficient whatever the image type.

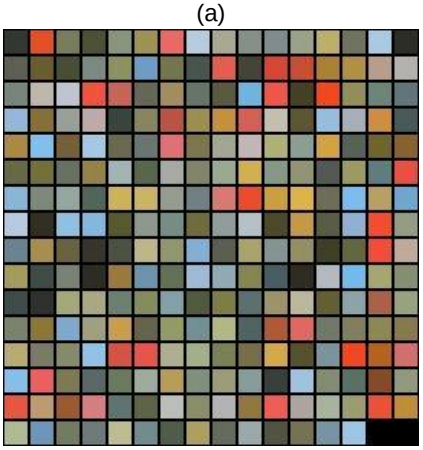
(a)

Figure 2. *a) Original color image of baboon.*

The first step of the approach is the quantization. Figure 3.a shows baboon quantified image obtained with a k-mean quantization and $K = 254$. Once quantization have been proceeded, a color palette and its *indexed image* are obtained, Figure 3.b and Figure 3.c. One could observe that the *indexed image* do not enable to understand its semantic content. With the application of the *layer running algorithm*, we obtain an ordered color palette (Figure 4.a) and its *indexed image* (Figures 4.b). Note that the *layer running algorithm* do not change the informational content i.e the color palette and the *indexed image* allow to rebuild the same color image before and after processing the *layer running algorithm*. Also note that the new *indexed image* (Figure 4.b) is more contrasted than the luminance of the original color image, Figure 4.c, but keep its semantic intelligibility. We could also remark in the color palette, Figure 4.a, that consecutive colors are colorimetricly closed.

The length of our embedded message (color palette) is $m =$ 6096 bits which gives an embedding factor, for an image of $256 \times$ 256 pixels, of $E_f = 0.093 bits/pixels$. The *indexed image* after

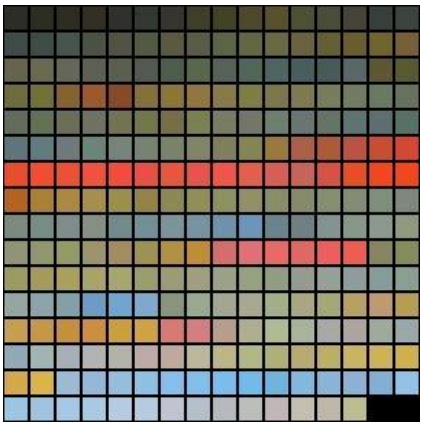
(a)



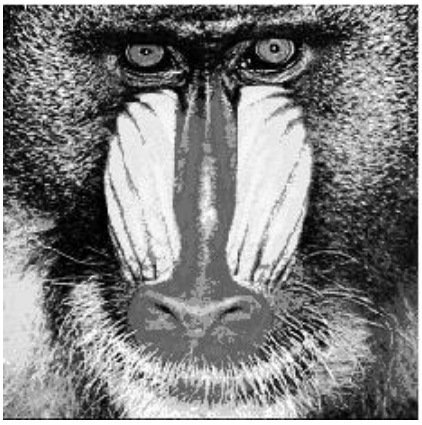
(b)

(c)

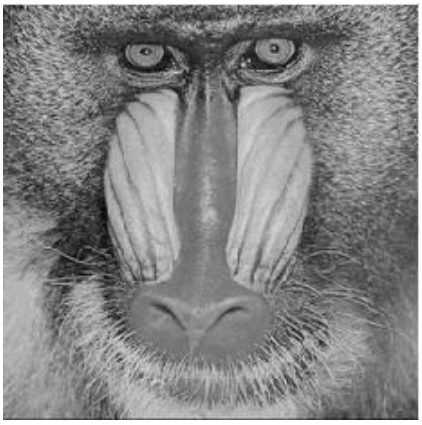
Figure 3. *a) Quantified image, b) Color palette of the quantified image, c) Indexed image from the quantified image.*



(a)

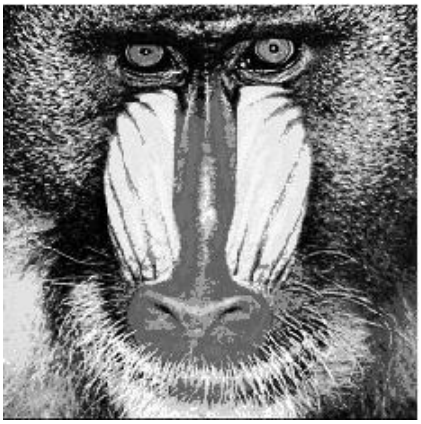


(b)

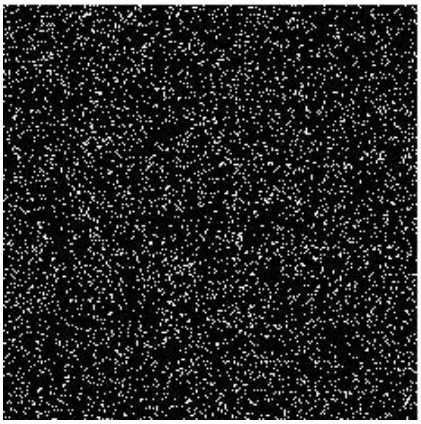


(c)

Figure 4. *Application of the* layer running algorithm *on the baboon image: a) Color palette after color ordering, b) Indexed image after color ordering, c) Luminance of the original color image.*

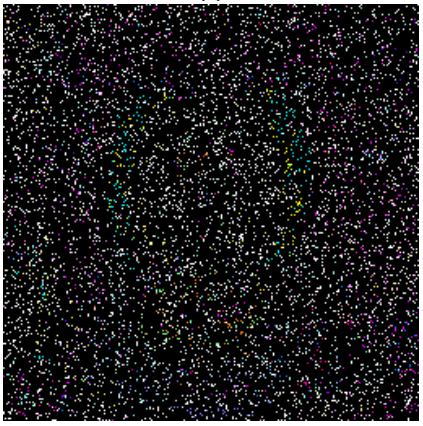


(a)



(b)

(c)



(d)

Figure 5. *Embedding the color palette in the indexed image after color ordering: a) Indexed and marked image, b) Difference image between the indexed image and the indexed marked one, c) Reconstructed color image from the indexed and marked one d) Color difference image between the quantified image and the reconstructed one.*

color ordering is then cut in block of 10 pixels. In each 10-pixel block, a bit of the color palette is embedded in the LSB of a pixel selected by the PRNG. We have used a secret key of 128 bits as a seed for the PRNG, so the distribution of the message over the image is key-related. This secret key has also been used to encrypt the color palette before the embedding.

Figure 5.a illustrates the marked image. Figure 5.b is the difference image between the *indexed image* and the indexed and marked one and shows insertion sites (white pixels) of the color palette. The PSNR computed between *indexed image* and indexed marked one is $PSNR = 58.1\ dB$. We can notice that a palette compression will moreover drastically reduce this distortion. Figure 5.c shows the reconstructed color image from the indexed marked one. It could be observed that this image is not visually far from the original quantified one. The PSNR value of 39.8 dB confirms this quality aspect.

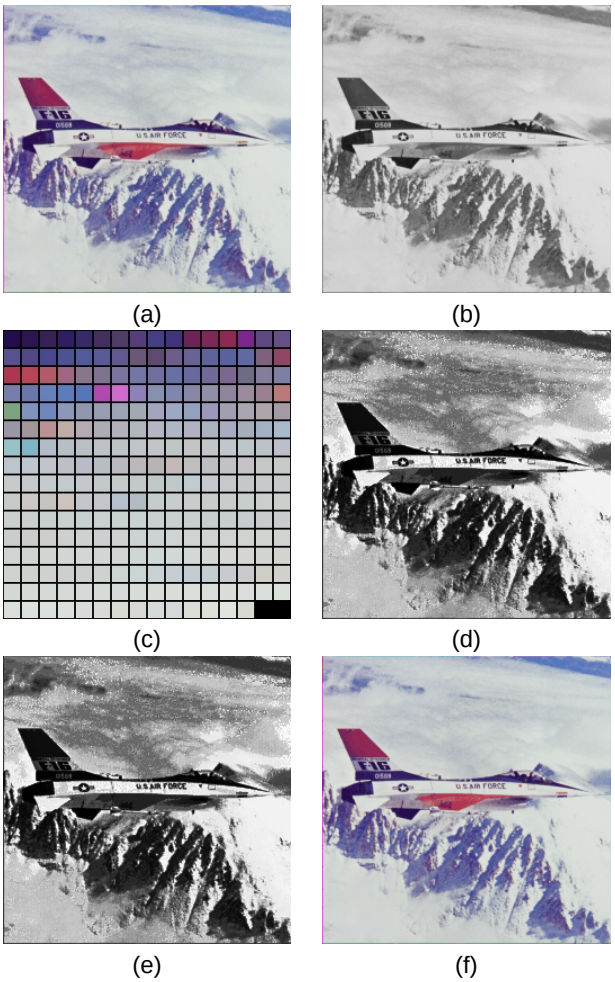


Figure 6. *Application of the* layer running algorithm *on the airplane image: a) Original color image, b) Luminance of the original color image, c) Color palette, d) Indexed image, e) Indexed and marked image, f) Reconstructed color image from the indexed and marked one.*

Results on Figures 6 and 7 similarly show that the color degradation due to color palette embedding is not visually perceptible. Those good results are possible thanks to the colors proximity of consecutive colors in the palette and thanks to the data hiding method. Finally, few PSNR values are given on the table below. One could remark that rebuild color images are of good quality (over 32 dB). PSNR values for indexed image are under 20 dB which is in general a poor result but it is known that PSNR measure is not well adapt in case of strong contrast distortion. Indeed, the indexed images are visually pleasant.

PSNR comparisons

images	$PSNR_{(luminance,index)}$	$PSNR_{(color\ image,rebuild)}$
baboon	16.32	32.89
airplane	12.95	38.89
house	18.56	38.07

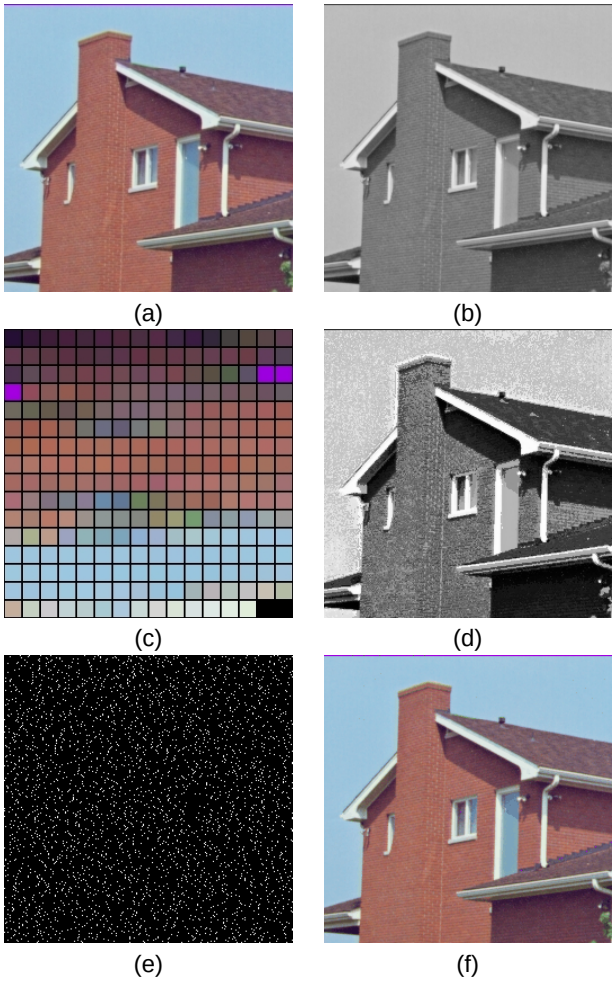


Figure 7. *Application of the* layer running algorithm *on the house image: a) Original color image, b) Luminance of the original color image, c) Color palette, d) Indexed image, e) Insertion sites (white pixels) of the color palette in the indexed image, f) Reconstructed color image from the indexed and marked one.*

Conclusion

In this paper, we have proposed a method to hide the color information in a grey-level image. This method is made of three major steps which are the color quantization, the color ordering and the data hiding. The originality of this paper is to build an *indexed image* which is a semantical intelligible grey-level image. Moreover, to obtain this particular *indexed image*, an original K color ordering algorithm is proposed: the *layer running algorithm*. The results obtained show good performances. One of our perspective work will be to treat other color spaces in order to improve the ordering of the K colors.

Acknowledgments

This investigation was in part supported by the TSAR project which is a french national project ANR-05-SSIA-0017-05 of the ANR ARA SSIA (*Agence Nationale de la Recherche, Action de Recherche Amont, Sécurité Systèmes Embarqués et Intelligence Ambiante*).

We would like also to thank Mr Lahanier Christian of the C2RMF (*Centre de Recherche et de Restauration des Musées de France*) and the Louvre Museum for the digital paintings and for valuable discussions.

References

[1] G. H. Ball and D. J. Hall. ISODATA, a novel method of data analysis and pattern classification. In *In Proceedings of the International Communication Conference*, June 1966.

[2] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for Data Hiding. *I.B.M. Systems Journal*, 35(3-4):313–336, 1996.

[3] R. Castagno and A. Sodomaco. Estimation of image feature reliability for an interactive video segmentation scheme. In *International Conference on Image Processing, ICIP'1998*, volume 1, pages 938–942, Chicago, USA, October 1998.

[4] J. C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.

[5] J. Fridrich and M. Goljan. Practical steganalysis: state-of-the-art. In *Proceeding of SPIE Photonics West, Electronic Imaging 2002*, volume 4675, pages 1–13, 2002.

[6] N. Nikolaidis and I. Pitas. Robust Image Watermarking in the Spatial Domain. *Signal Processing*, 66(3):385–403, 1998.

[7] J.M. Rodrigues, J.R. Rios, and W. Puech. SSB-4 System of Steganography using Bit. In *5th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2004), Lisboa, Portugal*, april 2004.

[8] M.-Y. Wu, Y.-K. Ho, and J.-H. Lee. An Iterative Method of Palette-Based Image Steganography. *Pattern Recognition Letters*, 25:301–309, 2003.

Author Biography

Marc Chaumont *was born in November 1976, in France. He received his Engineer Diploma in Computer Sciences at the INSA (National Institute of Applied Sciences) of Rennes, France in 1999 and his PhD in Computer Sciences at the INRIA of Rennes in 2003. His PhD thesis was on the interest of video objects representations, with dynamic coding and scalability functionality, in the video compression area. He has carry on for one year its research activity at the INRIA and for one another year at the University Technological Institute of Bayonne, France as a Visiting Assistant Professor. During this last year, his research has focus on face tracking using a deformable 3D face model. Nowadays, he is in the LIRMM (Laboratory of Computer Science, Robotic and Microelectronic) of Montpellier as an Assistant Professor. His current interests areas are video compression, data-hiding and segmentation & tracking in videos.*

William Puech *was born in December 1967, in France. He received the diploma of Electrical Engineering from the University of Montpellier, France, in 1991 and the Ph.D. Degree in Signal-Image-Speech from the Polytechnic National Institute of Grenoble, France in 1997. He initialized its research activities in image processing and computer vision. He served as a Visiting Research Associate at the University of Thessaloniki, Greece. From 1997 to 2000, he had been an Assistant Professor at the University of Toulon, France, with research interests include methods of active contours applied to medical images sequences. Since 2000, he is Associate Professor at the University of Montpellier, France. He works now in the LIRMM Laboratory (Laboratory of Computer Science, Robotic and Microelectronic of Montpellier). His current interests are in the areas of security of digital image transfer (watermarking, data hiding, compression and cryptography) and edges detection applied to medical images and road security.*