

A DCT-Based Data-Hiding Method to Embed the Color Information in a JPEG Grey Level Image

Marc Chaumont, William Puech

► **To cite this version:**

Marc Chaumont, William Puech. A DCT-Based Data-Hiding Method to Embed the Color Information in a JPEG Grey Level Image. EUSIPCO: European Signal Processing Conference, Nov 2006, Pise, Italy. 2006. <lirmm-00125715>

HAL Id: lirmm-00125715

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00125715>

Submitted on 22 Jan 2007

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.

A DCT-BASED DATA-HIDING METHOD TO EMBED THE COLOR INFORMATION IN A JPEG 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 an original method to embed the color information of an image in a corresponding compressed grey-level image. The objective of this work is to allow free access to the compressed 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 DCT-based data hiding. The novelty 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. Finally, the DCT-based data-hiding method benefits from the used of an hybrid JPEG coder which allows to compress images with a Word Wide Web standard format and in the same time proposes a data-hiding functionality.

1. 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 build on a data-hiding method. The image may be freely obtained but its high quality visualization require the used of a secret key. More precisely, in our solution, a JPEG compressed 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 thought 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 (section and the DCT-based 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 DCT-based data hiding step, the aim is to embed the color palette in the *indexed image* and to be robust to JPEG compression.

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*. Another originality of that paper is the used of an hybrid JPEG coder which allows to compress images with a Word Wide Web standard format and in the same time proposes a data-hiding functionality.

A lots of works proposed color watermarking or color data hiding methods but few methods use the color palette. Previous works used the color palette to watermark the image and to authenticate it [16, 14]. In particular, Wu *et al* propose to build a new palette in order to embed one message bit into each color of the palette [16]. The problem is to find the best color ordering. In our paper we propose a new method to order the colors of the palette. We use this new ordering to get a index image which is near of the luminance of the original color image . We can then embed this new color palette in the index image in order to protect the colors of images. In the works of Campisi *et al*, the main objective is to compress the color image and to authenticate it [4, 17]. Moreover, their works are different because they do not work with a color palette. De Queiroz and Braun propose to embed the color into the textured gray images [7] for image printing and not for image transmission. In this paper, the presented method is an extension of the proposed method in [6].

In the Section 2, we present the color quantization and the ordering steps. In the Section 3, we explain the choice of the color number and present the used DCT-based data hiding method. Section 4 we have applied our method on a digital painting of the Louvre Museum of Paris, France.

2. THE COLOR QUANTIZATION AND THE LAYER RUNNING ALGORITHM

In this section we present the color quantization and the ordering steps.

2.1 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:

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

where I is a color image of dimension N , $C(k)$ 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 $P_{i,k} \in \{0, 1\}$ is the membership value of pixel i to color k . The constraint for $P_{i,k}$ is, for all i it exists an unique k in order to have $P_{i,k} = 1$.

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:

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

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

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 initialize the $P_{i,k}$ values by solving the fuzzy c-mean equation below:

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

where m is the fuzzy coefficient (m is set to 1.6 as proposed in [5]) and $P_{i,k} \in [0, 1]$ are fuzzy membership values. This equation is solved by a fuzzy c-mean algorithm [9]

2.2 Layer running

Once the color quantification has been processed, the obtained K colors image, could be represented by an *indexed image* (thanks to $P_{i,k}$ 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 quantification, we already own an *indexed image* and a color palette. Our problem is then to find a permutation function which permute 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:

$$\Phi = \arg \min_{\Phi} \left[\sum_{i=1}^N E_i^{\text{ind}} + \lambda \sum_{k=1}^{K-1} E_k^{\text{palette}} \right], \quad (4)$$

$$E_i^{\text{ind}} = \text{dist}^2(Y(i), \Phi(\text{Index}(i))), \quad (5)$$

$$E_k^{\text{pal}} = \text{dist}^2(\text{Palette}(\Phi^{-1}(k)), \text{Palette}(\Phi^{-1}(k+1))), \quad (6)$$

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 $\Phi(k) = k'$.

To find an ordering of the K color, the algorithm 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 window defined on luminance information. This *layer running algorithm* could then be seen as a kind of "3D spiral run" in the color space. More details on this algorithm may be found on [6].

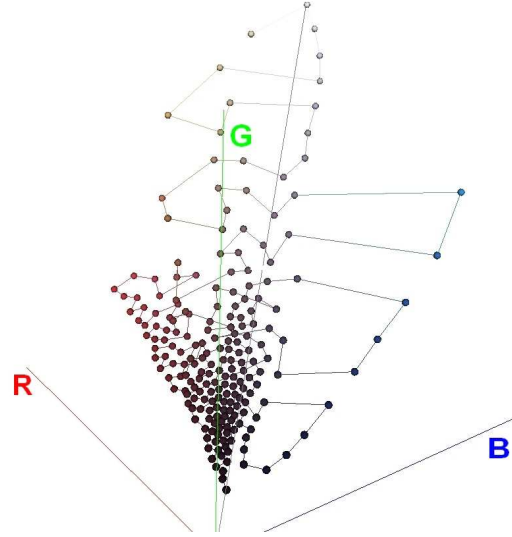


Figure 1: A view of the layer running in the RGB cube.

3. EMBEDDING OF THE COLOR PALETTE

3.1 Color number choice

In the previous section we explain the method used to build an *indexed image* (not too far from the luminance of the original color image), and a color palette (whose consecutive colors are not too far distant). The color number K were supposed known. This subsection gives an empirical way to choose this color number. One could choose a color number equals to 256 but it is not adapted to build an *indexed image* being similar to the luminance image. Indeed, the 256 index values are covered in the *indexed image* whereas it is not often the case in the luminance image. A cleverer solution is to choose a color number been equals to the grey-level range of the luminance image. Lets modify the energy, Equ. (4) to be able to reduce this colors number. Only the first term is changed:

$$E_i^{\text{ind}} = \text{dist}^2(Y(i), t + \Phi(\text{Index}(i))), \quad (7)$$

where t is a translation value.

In order to choose the color number from the luminance histogram we define a *relevance threshold* which is defined as 1% of the maximum histogram value. Grey-level values under this threshold are considered as negligible. One then define the relevant grey-level interval such that the lower bound is the first relevant grey-level *index* and the upper bound is the last relevant grey-level *index*. The interval size gives the colors number (K) and the lower bound gives the translation value (t).

Lets remark that giving a color number equals to the relevant interval size reduce the index range, then gives a less contrasted indexed image. *Indexed image* is then visually better and its PSNR is also better.

3.2 The used DCT-based Data hiding method

Data hiding methods can be an answer to make secure image transmission. For applications dealing with images, the watermarking objective is to embed invisibly message or marks inside the image. The length of the transmitted message can be relatively important, in fact, in this case we talk about

data hiding. The insertion can be made in different ways according to the length of the message or desired robustness. Data hiding can be classified and cataloged in distinct ways [8, 10, 12]. Basically for images, there are two great groups of methods for data hiding. The methods that work in spatial domain [2, 11] and the other group that work in frequency domain in particular in DCT domain [3, 13].

In this section, we describe an hybrid JPEG coder where the data hiding method is processed in the frequential domain. The data hiding method occurs after the DCT transformation. Giving a message M made up of k bits b_i such that $M = b_1 b_2 \dots b_k$, a bit b_i is embedded in the DC coefficient of a DCT block [15] The embedding process is obtained by substituting the Least Significant Bit (LSB) of the DC coefficient by the bit to embed.

Before any data-embedding, we calculate an embedding factor (function of the length of the message and of the image size) to choose the blocks that will be used to embed the message. The embedding factor, in *bit/pixel* is:

$$E_f = m/N. \quad (8)$$

The *indexed image* is then divided in areas of size $\lceil 1/E_f \rceil$. Each area is used to hide only one bit b_i of the message ; this bit been hidden in a block belonging to the area. 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 $m = 3 \times K \times 8$ bits in the *indexed image*. Consequently, the embedding factor E_f , Equ. (8), only depends on the image size N .

Thus, the objective is to embed a message M i.e the color palette. Giving an image I and a square block made up of n^2 pixels, the DCT continuous coefficient $F(0,0)$ is:

$$F(0,0) = \frac{1}{n} \sum_{i=0}^{n^2-1} I(i). \quad (9)$$

In lossy compression approaches such that JPEG, the DC coefficient is quantified and gives a quantified coefficient $F'(0,0)$:

$$F'(0,0) = [F(0,0)/Q(0,0)], \quad (10)$$

where $[\cdot]$ is the nearest integer function, and $Q(0,0)$ is the quantization value.

A classical solution in order to embed the message is to substitute $F(0,0)$ by $F_w(0,0)$. In order to hybridized the JPEG coder, this substitution take care of the JPEG quantization step such that:

$$F_w(0,0) = \begin{cases} \lfloor \frac{F(0,0)}{Q(0,0)} \rfloor \times Q(0,0) & \text{if } \lfloor \frac{F(0,0)}{Q(0,0)} \rfloor \% 2 = b_i, \\ \lceil \frac{F(0,0)}{Q(0,0)} \rceil \times Q(0,0) & \text{if } \lceil \frac{F(0,0)}{Q(0,0)} \rceil \% 2 = b_i, \end{cases} \quad (11)$$

where $\lfloor \cdot \rfloor$ is lowest nearest integer function and $\lceil \cdot \rceil$ is the highest nearest integer function. Note that the substitution of $F(0,0)$ by $F_w(0,0)$ is proceeded before the quantization step. Lets also remark that $F_w(0,0)$ is an integer.

Now, we propose to improve the previous embedding method. Indeed, the modification of the DC coefficient do not care of the spatial information of the corresponding block. This DC modification may then be visually annoying. In order to improve the visual result, the modification of the DC coefficient is obtained trough the modification of the grey-levels of few pixels belonging to the corresponding block. The modified pixels are the pixels owning

the strongest variance. Thus, instead of modifying directly $F(0,0)$ in order to get $F_w(0,0)$, we modify n_w pixels of the corresponding block in order to get the appropriate value for $F_w(0,0)$. The n_w pixels are modified to obtain new pixels $I_w(i)$ such that:

$$I_w(i) = I(i) - \text{sign}(F(0,0) - F_w(0,0)), \quad (12)$$

where $\text{sign}(x) = -1$ if $x < 0$ and $\text{sign}(x) = 1$ if $x \geq 0$. Note that the number of pixels to modify n_w is:

$$n_w = [|F(0,0) - F_w(0,0)| \times n]. \quad (13)$$

Lets remark that in the case of $n_w > n^2$, we apply the Equ. (12) on all pixels of the block and we repeat this operation for the $n_w \text{ modulo } (n^2)$ pixels.

To summarize our embedding method (which add the data-hiding functionality to the JPEG coder), we give the equation of the DC component that embedded a bit of the message, after the quantization step and for a given block:

$$F'_w(0,0) = \frac{1}{n \times Q(0,0)} \left(\sum_{i \in \Omega_w} I_w(i) + \sum_{i \in \Omega_w} I(i) \right), \quad (14)$$

where Ω_w is the set of n_w modified pixels.

4. RESULTS

We have applied our method on a detail of a digital painting of the data base of the Louvre Museum. This detail, of size 523×778 pixels, illustrated Figure 2.a., is from the painting representing Jean-Philippe Saint baptizing the eunuque one of the Candace Queen. The luminance image is shown on Figure 2.b and its histogram on Figure 2.b.

We could observed from the luminance histogram that lots of grey-levels are not relevant. As explain in section 3.1 an automatic thresholding is proceeded and gives a relevant grey-level interval of $[20, 222]$. The color number and the translation are thus automatically deduced: $K = 203$ and $t = 20$. Choosing a color number been equals to the grey-level range ensure a strong reduction of the first term of the energy Equ. (7) without any strong growth of the second energy term of the Equ. (6) and so without any strong growth of the distortion on color image due to data-hiding. A visual comparison between Figures 3.a and 3.b shows a more pleasant (less contrasted) grey-level image with $K = 203$ colors. For the quantified images Figures 3.c and 3.d, not any visual differences could be noticed between $K = 203$ colors and $K = 254$ colors.

Once quantization with $K = 203$ colors have been proceeded, a color palette and its *indexed image* are obtained. Figures 4.a, and 4.c. illustrate this classical result. One could observed that the *indexed image* do not enable to understand its semantic content. With the application of the *layer running algorithm*, we obtain a ordered color palette Figure 4.b and its *indexed image* Figure 4.d which is semantically intelligible. Note that the *layer running algorithm* do not change the informational content i.e the color palette and the indexed image allows to rebuild the same color image before and after processing the *layer running algorithm*.

Before any embedding process the message to embed (the color palette) is predictively and arithmetically coded. The length of the message decreases from 4872 bits to 3183 bits. The embedding factor is then $E_f = 0.0078$ bit/pixel.

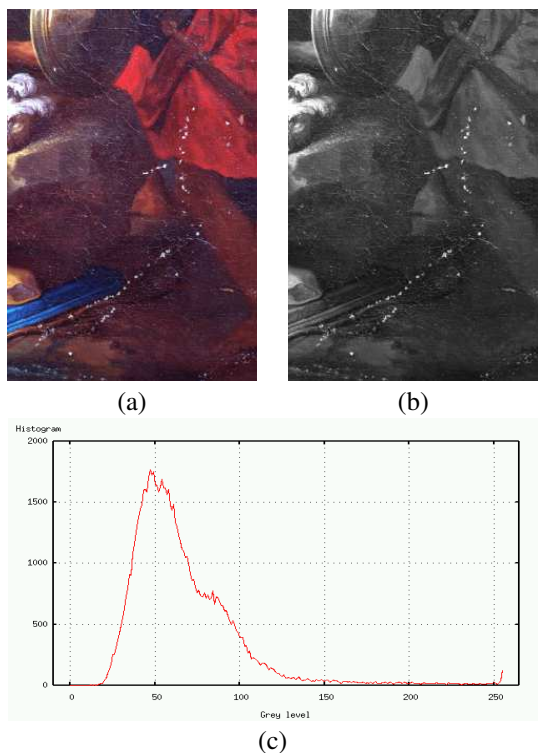


Figure 2: a) Original color image, b) Luminance of the original color image, c) Histogram of the luminance.

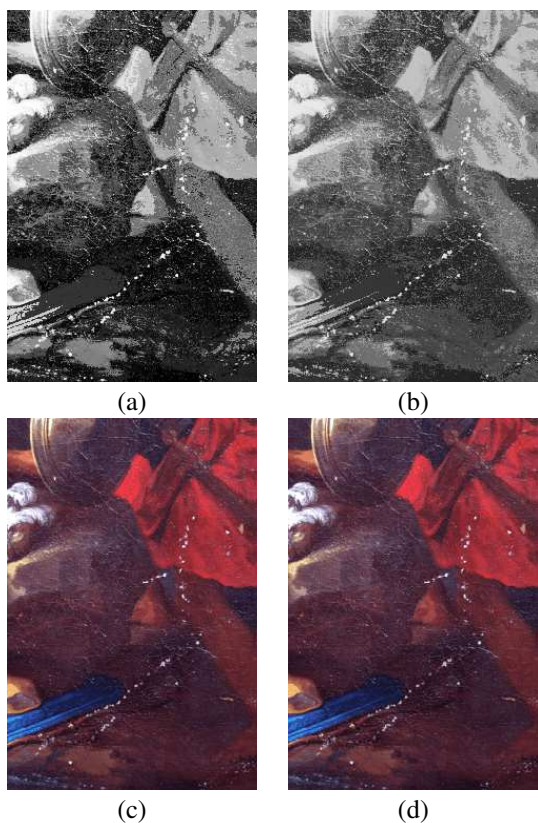


Figure 3: Comparison between cases $K = 254$ colors and $K = 203$ colors. a) Indexed image after color ordering with $K = 254$, b) Indexed image after color ordering with $K = 203$, c) Quantified image with $K = 254$ colors, d) Quantified image with $K = 203$ colors.

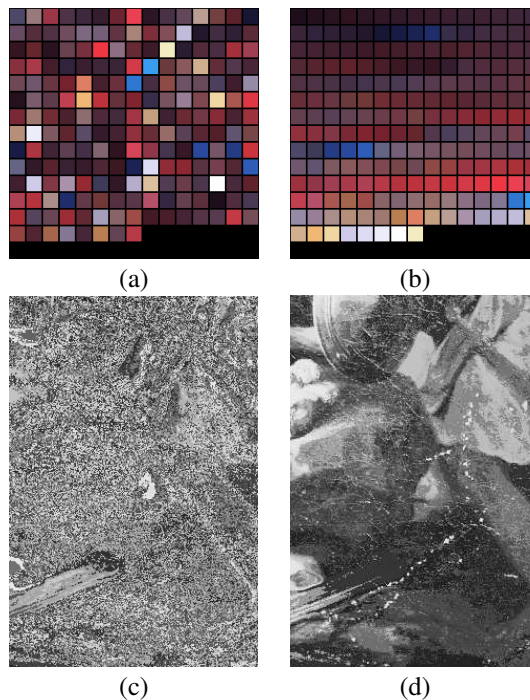


Figure 4: Application of the *layer running algorithm*: a) Color palette of the original quantified image ($K = 203$ colors), b) Color palette after color ordering, c) Indexed image from the original quantified image, d) Indexed image after color ordering.

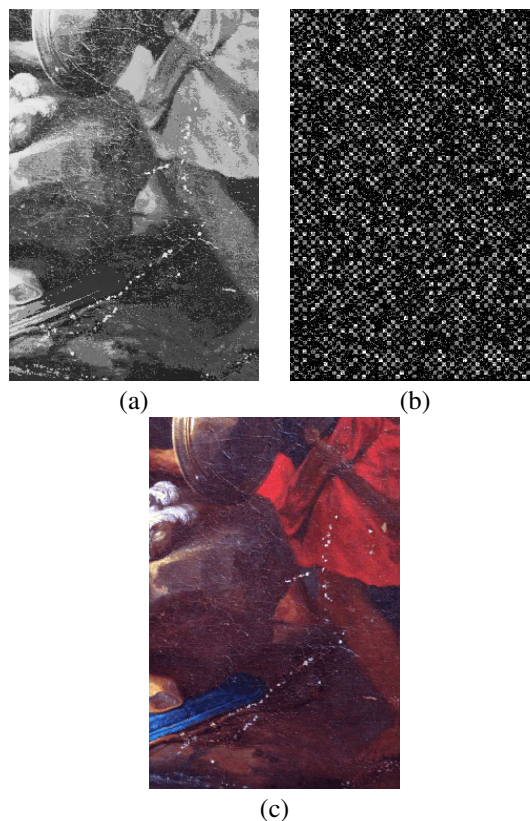


Figure 5: DCT-based data-hiding. a) Indexed and marked image, with $K = 203$ and $m = 3283$ bits, b) Difference image between the indexed image and the indexed marked one, c) Reconstructed color image from the indexed and marked one.

The *indexed image* is then cut in area of $\lfloor 1/E_f \rfloor = 128$ pixels and one bit of the message is embedded in one block belonging to one area. In order to spread the message over the image with a key-related distribution, we use a secret key of 128 bits as a seed for a PRNG. This secret key is also used to encrypt the color palette before the embedding. The image is then compressed with our JPEG hybrid coder (compression and data-hiding) with a quality factor of 100%.

Once our hybrid JPEG coder (compression and data-hiding) has coded the *indexed image*, it is possible for anyone to freely access to the grey-level JPEG image illustrated Figure 5.a. With the secret key, the color palette is extracted and the color image is rebuild. Figure 5.c shows the reconstructed color image from the indexed marked one. It could be observed that the image quality is very good. The PSNR value of 41.2 dB confirms this quality aspect. The data hiding method used and the proximity of consecutive colors in the color palette explain this good result. Figure 5.b shows the difference image computed between the *indexed image* and the indexed, marked and compressed image. One could observed that the pixels modification occur mostly everywhere due to data embedding and to compression. One could also remark that even if the modification is dense, the rebuild image is still visually satisfying.

5. CONCLUSION

In this paper, we have proposed a method to hide the color information in a compressed 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*. An original hybrid JPEG coder allows to embed the color palette inside the *indexed image*. This data-hiding process allows to compress images with a World Wide Web standard format and gives us a ready-to-use solution for securely published the digital paintings of the Louvre Museum.

Acknowledgments

This investigation was in part supported by the TSAR project (Transfert Sécurisé d'images d'Art haute Résolution) 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 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] A.G. Bors and I. Pitas. Image watermarking using block site selection and DCT domain constraints. *Optics Express*, 3(12):512–522, 1998.
- [4] P. Campisi, D. Kundur, D. Hatzinakos, and A. Neri. Compressive Data Hiding: An Unconventional Approach for Improved Color Image Coding. *EURASIP Journal on Applied Signal Processing*, 2002(2):152–163, 2002.
- [5] 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.
- [6] M. Chaumont and W. Puech. A Color Image in a Grey-Level Image. In *CGIV'06, Leeds, UK*, 2006.
- [7] R. de Queiroz and K. Braun. Color to Gray and Back: Color Embedding Into Textured Gray Images. *IEEE Trans. on Image Processing*, 15(6):1464–1470, 2006.
- [8] J.F. Delaigle, C. De Vleeschouwer, and B. Macq. Watermarking algorithm based on a human visual model. *Special Issue on Watermarking, Signal Processing*, 66(3):319–336, 1998.
- [9] 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.
- [10] Z. Duric. *Information Hiding, Steganography and Watermarking - Attacks and Countermeasures*. Kluwer Academic Publishers, Boston, 2001.
- [11] N. Nikolaidis and I. Pitas. Robust Image Watermarking in the Spatial Domain. *Signal Processing*, 66(3):385–403, 1998.
- [12] F.A.P. Petitcolas, R. J. Anderson, and M.G. Kuhn. Information Hiding-A Survey. *IEEE, special issue on protection of multimedia content*, 87(7):1062–1078, July 1999.
- [13] H.-W. Tseng and C.-C. Chang. High Capacity Data Hiding in JPEG-Compressed Images. *Informatic, Institute of Mathematics and Informatic, Vilnius*, 151(1):127–142, 2004.
- [14] C.H. Tzeng, Z.F. Yang, and W.H. Tsai. Adaptive Data Hiding in Palette Images by Color Ordering and Mapping With Security Protection. *IEEE Trans. on Communications*, 52(5):791–800, 2004.
- [15] D. Upham. Jpeg-jsteg, Modification of the Independent Jpeg Group's Jpeg Software for 1-bit Steganography in Jfif Output Files. In *ftp://ftp.funet.fi/pub/crypt/steganography*, 1997.
- [16] 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.
- [17] Y. Zhao, P. Campisi, and D. Kundur. Dual Domain for Authentication and Compression of Cultural Heritage Images. *IEEE Trans. on Image Processing*, 13(3):430–448, 2004.