FAST PROTECTION OF THE COLOR OF HIGH DIMENSION DIGITAL PAINTING IMAGES

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 **high dimension** image in its corresponding grey-level image. The objective of this work is to allow a free and **rapid** access to the grey-level image and give color image access to secret key owners. This method is made of two steps which are the **fast high-dimension** color image decomposition (in a grey-level image and its associated color information) and the data-hiding. The two main contributions of this paper are the energy function proposed to model the decomposition of the color image and the fast optimization. The optimization of the proposed energy function leads to the achievement of an *index* image and a color palette. The good properties of that decomposition are an *index* image which is similar to the luminance of the color image and a color palette which is well suit for the data-hiding. The obtained results confirm the model quality and the real-time property.

1. INTRODUCTION

Nowadays, only few secure solutions are proposed, to protect digital painting data-base, in order to give both a free access to low-quality images and a secure access to the same images with an higher quality. Our proposed 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 envisage to give a limited access to the private digital painting data-base of the Louvre Museum of Paris, France.

In order to obtain a grey-level image embedding its color information, we decompose a color image in an *index* image and a color palette. The color palette is then hidden in the *index* image. The *index* image should be similar to the luminance of the color image, the embedding process should be of week magnitude and the color palette should be cleverly ordered. The originality of this paper is to propose a solution for this very constrained decomposition and this with real-time possibility. Thus, the main contribution is the energy function proposed to model the decomposition of the color image and the fast optimization of this model.

Many work propose solutions to hide information by using the decomposition of a color image in an *index* image and a color palette. The data-hiding may occur in the *index* image [1] or in the color palette [2, 3]. Nevertheless, none of those techniques tries to protect the color information by hiding the color palette in the *index* image. Only the previous work of [4] protect the color information by hiding the color palette in the *index* image. Authors of [4] sort the colors of the color palette in order to get an *index* image which closed of the luminance of the original color image and in the same time they get a color palette whose consecutive colors are close. In this paper, the approach is completely different and relies on a function optimization of the global problem formulation.

Other works such that [5, 6, 7] based on wavelet decomposition and sub-band substitution propose solutions to embed the color information in a grey-level image. Their areas are perceptive compression and image authentication for [5, 6] and image printing for [7]. Even if those techniques embed the color information, their approach and their purpose are clearly different from that exposed in that paper.

In section 2, we present the proposed energy model. Section 3 deals with the secured data-hiding. In section 4, results are presented and analyzed.

2. ENERGY MODEL

In the section below, one propose a mathematical model for the decomposition of a color image in an *index* image plus an associated color palette. In the section 2.2, a second model close to the first one is introduced in order to find a fast and cheap memory solution for this decomposition. Finally, in section 2.3, we present a *Fast Decomposition Algorithm* which is approaching the minimum of the first model.

2.1. First model

Our goal is to find an *index* image and a color palette with the following three constraints:

- the *index* image should be close from the luminance of the original color image,
- the color quantized image should be close from the color image,
- and the color palette should own consecutive couples of close color.

Those three constraints come mathematically to found K colors $C(k)$ (C is the color palette) and for each pixel i the *index* value $Index(i)$. Thus, we are looking to minimize the above energy model in order to obtain $\forall i \in [1, N]$, $Index(i)$ and $\forall k \in [1, K]$, $C(k)$:

$$\begin{aligned} E1 = & \underbrace{\sum_{i=1}^{N} (C(Index(i)) - I(i))^2}_{first\ term} \\ & + \underbrace{\lambda_1 \sum_{i=1}^{N} (Index(i) - Y(i))^2}_{second\ term} \\ & + \underbrace{\lambda_2 \sum_{k | k \in [1..K]\ and\ k\ is\ odd} (C(k) - C(k+1))^2}_{third\ term}, \end{aligned} \tag{1}$$

with I the color image, Y the luminance image, $\lambda1$ and $\lambda2$ two scalar values.

The first term is expressing the constraint of color quantization. Its role is to exhibit the best representative K colors, i.e find C, knowing the color image I. The second term stand for getting the *index* image the nearest to the luminance image Y. The last term constrain couples of consecutive color from the palette to be close.

The minimization of Equation 1 such that:

$$\{P_{i,k}, C(k)\} = \underset{\{P_{i,k}, C(k)\}}{\arg} \min E1, \tag{2}$$

is not feasible by using derivative approach. Indeed, the function $E1$ of Equation 1 is not derivable since the C function ($C : [1..K] \rightarrow [0..255]^3$) is discrete. Instead of using meta-heuristic approaches (such that evolutionist algorithms) in order to solve the Equation 2, one prefer a less CPU and memory costly solution.

2.2. Second model

In order to solve rapidly the Equation 2, one propose to minimize another equation, whose solution is close, with a pixel sub-sampling of the color image. The new equation, minimized on the pixel sub-sampling, (in order to obtain $\forall i$, $\forall k$, $P_{i,k}$ and $C(k)$) is:

$$\begin{aligned} E2 = & \sum_{i=1}^{N} \sum_{k=1}^{K} P_{i,k}^m (C(k) - I(i))^2 \\ & + \lambda_1 \sum_{i=1}^{N} \sum_{k=1}^{K} P_{i,k}^m (Y(i) - k)^2 \\ & + \lambda_2 \sum_{k | k \in [1..K]\ and\ k\ is\ odd} (C(k) - C(k+1))^2, \end{aligned} \tag{3}$$

with $P_{i,k}$ the ownership values giving the degree of belongingness of a pixel i to the kth color and m$\in]1, \infty[$ the fuzzy coefficient tuning the equi-probability degree[1]. Note that $P_{i,k}$ belongs to [0,1] and are named fuzzy membership values in fuzzy c-mean clustering approach [8]. Also note that the $P_{i,k}$ give indirectly the *index* image such that: $Index(i) = \arg_k \max_k P_{i,k}$.

The minimization of Equation 3 such that:

$$\{P_{i,k}, C(k)\} = \underset{\{P_{i,k}, C(k)\}}{\arg} \min E2, \tag{4}$$

is performed iteratively, on a pixel sub-sampling, in a two steps loop as in conventional fuzzy c-mean algorithms. In the first step, colors $C(k)$ are updated, given $P_{i,k}$, by solving the linear system below:

$$\underline{\forall k\ odd:}$$
$$(\lambda_2 + \sum_{i=1}^{i=N} P_{i,k}^m) \times C(k) - \lambda_2 \times C(k+1) = \sum_{i=1}^{i=N} P_{i,k}^m I(i),$$
$$\underline{\forall k\ even:}$$
$$-\lambda_2 \times C(k-1) + (\lambda_2 + \sum_{i=1}^{i=N} P_{i,k}^m) \times C(k) = \sum_{i=1}^{i=N} P_{i,k}^m I(i).$$

In the second step, $P_{i,k}$ (with m=2) are updated given the colors $C(k)$ with:

$$P_{i,k} = \frac{\left(\sum_{l=1}^{l=K} \frac{1}{2 \times ((C(l) - I(i))^2 + \lambda_1 (Y(i) - l)^2}\right)^{-1}}{2 \times ((C(k) - I(i))^2 + \lambda_1 (Y(i) - k)^2)}$$

2.3. Fast Decomposition Algorithm

The *Fast Decomposition Algorithm* objective is to approach the minimum of Equation 2. As explain above, the optimization of Equation 1 on high dimension image is very CPU and memory costly if using meta-heuristic. Our solution is then to approach the minimum by choosing the best solution among a set of *possible solutions*. Those *possible solutions* are computed by using the minimization of the second model, explain in section 2.2 and with a limited amount of data i.e a random selection of few pixels of the original high-dimension image.

The *Fast Decomposition Algorithm* iteratively repeat those three steps:

[1] m is set to 2 for computational complexity reduction.

- Select randomly few pixels to create a small color image[2],
- Minimize Equation 3; A color palette C is obtained and the *index* image $Index$ is deduced from Equation 1:

$$\forall i, Index(i) = \arg\min_k (C(k) - I(i))^2 + \lambda_1 (k - Y(i))^2;$$

- Compute the fitting value $E1$ with another sub-sampling, knowing C and $Index$, and keep this solution if it is the best one.

3. SPATIAL DATA HIDING METHOD

In this paper, we embed the color palette information in the LSB of the N pixels *index* image. The objective is thus to embed a message W made up of l bits b_j ($W = b_1 b_2 ... b_l$). The embedding factor, in $bit/pixel$, is $E_f = l/N$. The *index* image is then divided in areas of size $\lfloor 1/E_f \rfloor$ pixels. Each area is used to hide only one bit b_j 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 *index* image we need to embed $l = 3 \times 256 \times 8 = 6144$ bits (the number of colors is $K = 256$).

Consequently, the embedding factor E_f, only depends on the image size N. In our process, the PRNG (Pseudo-Random Number Generator) selects randomly, for each region, a pixel $Index(i)$. In order to get a marked pixel $Index_W(i)$, the LSB of this selected pixel $Index(i)$ is then modified according to the message bit b_j[3]:

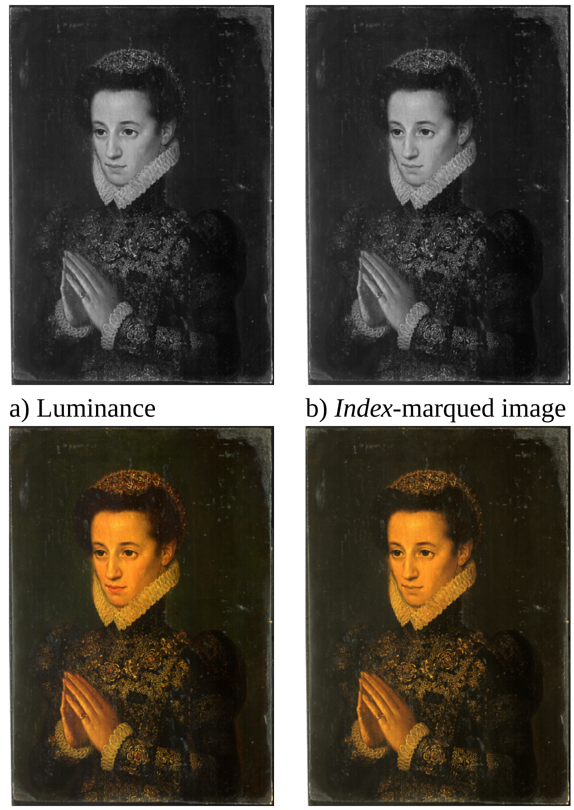
$$Index_W(i) = Index(i) - Index(i) \bmod 2 + b_j.$$

This way to embed the color palette ensure that each marked pixel is at worst modified by one grey-level and in the same time that the rebuilt color pixel would not be very far from the right color value. Indeed, the third term of Equation 1 ensures that consecutive couples of color are close.

4. RESULTS

We have applied our method on High-Dimension digital painting images. For all the experiments, $\lambda_1 = 1$, $\lambda_2 = 0.01 \times N/(K+1)$ and $m = 2$ (see Equation 1). The results obtained show that the approach is efficient whatever the image type. Below, the main steps of our approach are comment on a digital painting image (8248×11816 pixels size) of the data-base of the Louvre Museum Paris. This painting shows a *woman praying* (Anonymous, Flandres, XVI century, Oil on oak).

After proceeding to the *Fast Decomposition Algorithm* on the *woman praying* image with $K = 256$ colors we obtain an *index* image illustrated in Figure 1.b and its color palette in Figure 2. This decomposition would have been impossible in a current desktop computer if proceeded on the full data. Indeed, the memory complexity necessary for the minimization of Equation 3 on the complete image is around 186 GB[4]. With a sub-sampling of 9604 pixels (around 10000 times less pixels) the memory complexity is falling to only 19 MB which is little[5].



a) Luminance b) *Index*-marqued image

c) Color image d) Rebuilt image

Fig. 1. Grey-level image hiding its color palette and the rebuilt color image.

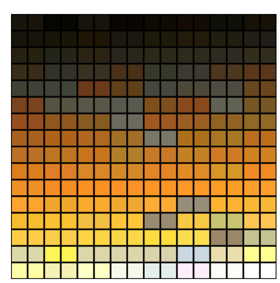


Fig. 2. Color palette.

Lets also note that even if the memory complexity would not have been problematic, the computation complexity would

[2] This randomly sub-sampling image has no visual meaning.

[3] The formula is given for *index* values belonging to [0,K-1].

[4] The memory complexity calculus is a low estimation; it takes into account $P_{i,k}$, $C(k)$, $I(i)$, $Y(i)$, and the linear system.

[5] A relatively cheap but modern desktop computer, for example, a Pentium 4, owns around 512 MB of RAM.

have. Lets take the example of the computation of the $P_{i,k}$ values for an iteration of the algorithm given in section 2.2. With the sub-sampling, it takes 22 127 616 operations and it takes 0.5 second on an Intel Pentium 4, 3.2 GHz with 1GB RAM i.e 44255232 operations per second. Without sub-sampling, the computation of the $P_{i,k}$ values necessitates 224 738 996 608 operations which give 224738996608/44255232 = 1.4 hours. Remember that in order to decompose the color image, one should run more or less 100 iterations to obtain **just one** solution for the minimisation of Equation 3 and **few solutions** should be compute as explain in section 2.3 in order to choose the best one.

The luminance image of the original color image is given in Figure 1.a. One could observe the good similarity between *index* image and luminance image. The good PSNR value of 39.3 dB confirms this subjective feeling. Note that lots of *index* colors are unused which explain the presence of some useless colors on the color palette of Figure 2. Also note that in the color palette in Figure 2, consecutive couples of colors are colorimetricly close as expressed by the third term of Equation 1.

The length of our embedded message (color palette) is $l = 6144$ bits which gives an embedding factor for an image of 8248×11816 pixels of E_f = 6144/(8248×11816) = 6.3$\times$10^{-5} bits/pixel. The *index* image is then cut in block of 150×105 pixels. In each block, a bit of the color palette is embedded at the position selected by the PRNG as explain in Section 3. The secured is obtain through the used of a secret key of 128 bits as a seed for the PRNG. The distribution of the message over the image is then key-related.

Figure 1.d shows the rebuilt color image from the *index*-marked one. This image is not visually far from the original color image and the PSNR value of 34.2094 dB is of good quality. Note that the degradation due to the data-hiding method is weak because it disturb *index* values of a maximum of one. This is made possible thanks to the color palette property to own consecutive couples of close colors. Few other PSNR values are given on the Table 1 for well known images. Rebuilt color images are of midle quality (over 27 dB) but visually pleasant. PSNR values for *index*-marked images are over 30 dB which is a really good result in comparison to the results of [4]. In the same times, our approach allows the treatment of High Dimension Digital Painting. For all those results, the necessary time to compute the decomposition is under the minute on an Intel Pentium 4, 3.2 GHz with 1GB RAM (each time, 10 solutions have been tested).

Table 1. PSNR comparisons

images	PSNR^{lum}	PSNR^{color}
baboon (512×512)	30.9 dB	27.6 dB
airplane (512×512)	38 dB	32.8 dB
pepper (512×512)	33.1 dB	31.8 dB
house (256×256)	36.4 dB	31 dB
barbara (787×576)	32.1 dB	29.8 dB

5. CONCLUSION

In this paper, we have proposed a method to embed securely into a grey level image its color information. This method is built on a fast decomposition of a color image in an *index* image and a color palette. The *index* image is playing the role of the luminance image and the color palette is hidden into this *index* image. The originality of this paper is to model the problem with an energy function whose solution is rapidly extracted even on high dimension images. Obtained results show a real improvement in comparison to [4] and are feasible on high dimension images. Our perspective work will treat of compression possibilities and other more robust data-hiding approaches.

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.

6. REFERENCES

[1] J. Fridrich, “A New Steganographic Method for Palette-Based Images,” in *Proceedings of the IS&T PICS conference*, Apr. 1998.

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

[3] C.H. Tzeng, Z.F. Yang, and W.H. Tsai, “Adaptative Data Hiding in Palette Images by Color Ordering and Mapping With Security Protection,” *IEEE Transaction on Communications*, vol. 52, no. 5, pp. 791–800, 2004.

[4] M. Chaumont and W. Puech, “A Color Image in a Grey-Level Image,” in *IS&T Third European Conference on Colour in Graphics, Imaging, and Vision, CGIV’2006*, Leeds, UK, June 2006, pp. 226–231.

[5] 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*, vol. 2002, no. 2, pp. 152–163, 2002.

[6] Y. Zhao, P. Campisi, and D. Kundur, “Dual Domain for Authentication and Compression of Cultural Heritage Images,” *IEEE Transaction on Image Processing*, vol. 13, no. 3, pp. 430–448, 2004.

[7] R. de Queiroz and K. Braun, “Color to Gray and Back: Color Embedding Into Textured Gray Images,” *IEEE Transaction on Image Processing*, vol. 15, no. 6, pp. 1464–1470, 2006.

[8] J. C. Dunn, “A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-Separated Clusters,” *Journal of Cybernetics*, vol. 3, pp. 32–57, 1974.