Robust Watermarking for Images Based on Color Manipulation

Sebastiano Battiato[1], Dario Catalano[1], Giovanni Gallo[1], Rosario Gennaro[2]

[1] Dipartimento di Matematica Universita' di Catania. Viale A.Doria 6, 95125 Catania.
E-mail: {battiato,catalano,gallo}@dipmat.unict.it.
Work done while the second author was visiting the Computer Science Dept. of Columbia University.
[2] IBM T.J.Watson Research Center, PO Box 704, Yorktown Heights, New York 10598, USA.
Email: rosario@watson.ibm.com

Abstract. In this paper we present a new efficient watermarking scheme for images. The basic idea of our method is to alter the colors of the given image in a suitable but imperceptible way. This is accomplished by moving the coordinates of each color in the color opponency space. The scheme is shown to be robust against a large class of image manipulations. The robustness of the scheme is also theoretically analyzed and it is shown to depend on the number of colors in the image being marked.

1 Introduction

The proliferation of the Internet as a new media, opens many new possibilities and opportunities for information and communication. If the new opportunities are many, so are the challenges. It's clear that, in the digital world, making a copy of a document means producing an exact copy of it: there is no degradation in the copying process. Digital watermarking is the embedding of a mark into digital content that can later be, unambiguously, detected to allow assertions about the ownership or provenience of the data. This makes watermarking an emerging technique to prevent digital piracy. There are several parameters and criteria over which categorize watermarking schemes due to their widespread application in several areas. Excellent overviews on various aspects of digital watermarking can be found in [1, 2, 9, 17, 20, 28]. In section 2 of this paper basic notions about watermarks with particular reference to the case of digital images are briefly surveyed. The contribution proposed in this paper is a novel technique to insert an imperceptible mark in digital images. The new method is, in a sense, similar to the technique introduced by Cox et al. [8] (see next Section). These authors propose to edit in an imperceptible way some frequencies in the spectrum of the signal to be marked. Differently than [8] we propose to insert the watermark in a suitable color space of the image (i.e. the *color opponency* space [22]): editing in an imperceptible way the palette of an image allows the insertion of a unambiguous mark. In the paper we show how this can be done efficiently and robustly. The main advantage of the proposed technique is to avoid costly computation of DCT or FT. On the other hand it could be argued that LUT (Look up Table) operations can easily destroy the proposed watermark: our claim is that for these attacks to be effective they must introduce perceptible degradation of

the original picture. In the paper we support this claim through a statistical analysis and experimental evidences.

2 Watermarking basics

Here we summarize some general parameters for categorizing watermarking schemes.

First a watermark can be *perceptible* or *imperceptible*. A perceptible watermark is typically used to encode information that should be known to the final user (i.e. ownership information or usage instruction etc.) An imperceptible watermark is more useful in those context in which the content owner doesn't want the user to know about the watermark (or at least not to know where the watermark is). Roughly speaking, by applying an imperceptible watermark to a digital image, we get a marked copy that is "almost" identical to the original unmarked one, where the meaning of "almost" depends on several parameters, one of which could be how much degradation is allowed in the marked data (with respect to the original).

Also a watermarking technique can be *fragile* or *robust*. Fragile watermarks can be easily corrupted by almost any kind of manipulation on the data (i.e. in the case of images, any image processing operation could damage the mark). Fragile watermarks are useful to preserve the integrity of the marked document, and more generally for authentication purposes (since small changes in the data will cause the corruption of the watermark). On the other hand robust watermarks have to resist the most common trasformations on the data (again in the case of images, typical procedures of image manipulation are filtering, JPEG compression, resizing, etc.) A desirable property would be also to be able to resist malicious attacks aimed at the removal of the watermark. Clearly, robust watermarks are useful in contexts where ownership has to be proved or preserved.

PREVIOUS WORK Several techniques have been proposed in the last few years. One of the first watermarking scheme has been proposed by [26]. Here the basic idea is to insert the watermarking signal (a sequence of pseudo-random bits) into the LSB position of the original image, pixel by pixel. Clearly this scheme is not robust: being embedded in the least significant bit of the pixel it can be easily removed.

Other somehow non-robust techniques are proposed in [3, 5, 7, 15, 19, 18, 27].

Cox et al. [8] pointed out that a watermarking scheme to be robust has to be placed in the perceptually most significant components of the digital data (their method can also be used in audio, video and multimedia in general). This approach can seem contradicting the requirement of imperceptibility of the watermark. However a reasonable tradeoff between imperceptibility and robustness can be obtained using some properties of spread-spectrum communication. This scheme is secure against almost all common signal processing and geometric distortion operations and even against some more malicious attacks.

3 Color spaces

The watermarking technique introduced in this paper operates an editing over a color space of an image. The proposed algorithm requires, for this color space the following properties:

i) a simple and compact way to describe geometrically in the color space a small, perceptively constant region;
ii) the transformation from a standard RGB space to the color space should be fast and easily and robustly inverted.

The algorithm performs small imperceptible perturbations of the palette according to some simple geometrical rules described in the next Section. In particular a colour is *moved* inside a sphere of almost equal colors. The common RGB space, unfortunately, is not well suited to these operations: even small sphere in such a space could be perceptively not very homogeneous. Computer Graphics has, since long, recognized the need for perceptively uniform color spaces (see [12], [13]) and several color models addressing this issue are known: L*u*v*, L*a*b*, etc. The transformation from RGB to such spaces are generally non-linear, hard to invert, and what is worse for our application introduce a requantization of the color. In short they violate the requirement ii) above. For this reason, although these spaces are ideally the best alternative to RGB, we choose to operate in the CO, *Color Opponency* space [22]. This choice grants a linear, easy to invert mapping from RGB to CO. The mapping is realized through the the equations:
RGB $\rightarrow$ Co:

$$A = R + G + B; B/Y = 2B - R - G; R/G = R - 2G + B. \tag{1}$$

CO $\rightarrow$ RGB:

$$R = \frac{A + R/G - B/Y}{3}; G = \frac{A - R/G}{3}; B = \frac{B/Y + A}{3}. \tag{2}$$

Perceptual studies strongly suggest, moreover, that this space is probably the best choice to hide small perturbations in colors, at least to an human observer. *Color Opponency* model is, indeed, very close to the chromatic channels of human visual system. Another, indirect, proof, of the suitability of *Color Opponency* in dealing with human perception is that many computer vision algorithms aimed to recognize specific texture/colors (like for example human skin [11]) adopt this model. In the following Sections it is assumed that the color are already represented in the CO space. Conversion from RGB is a preprocessing step that is taken for granted.

4 The algorithms

A watermarking mechanism is composed by two algorithms. The watermark *insertion* algorithm which takes as input the original image and outputs the marked image and a mark that is stored in a database with the identity of the acquiring user. The watermark

detection algorithm takes as input the original image, a watermarked image and the database of marks and outputs the mark which is associated with the input marked image.

The crucial tool used by the algorithm is to associate to each color in the image (thought of as a point in the CO space) a sphere in the CO space centered around it. The points of the sphere represent colors that are indistinguishable to the human eye from the original color.

At the moment we do not claim that our scheme is resistant against collusion attacks. Indeed, notice that if all rays (on each color sphere) have the same length in all marked images, an obviuos collusion attack can be mounted. As soon as an adversary gets at least four different marked images the mark can be removed with overwhelming probability (since four different points not belonging to the same plane, uniquely determine a sphere in the space and this would allow the adversary to reconstruct the ray and remove the mark). An easy countermeasure to this particular attack would be to choose the length of the ray randomly (in a suitable range) for each color sphere and each marked image. However it should be kept in mind that adopting this solution does not necessarily protect against more clever collusive attacks.

4.1 The Insertion Algorithm

The insertion algorithm starts by classifying the pixels of the image according to their color. Then we modify each color in the image in a random, yet impercetible, fashion. That is, for each color in the image, we move the color space point associated with it to the border of its color sphere. The direction in which the color space point is moved is chosen at random for each color.

More in detail, we are given an image I represented as three matrices in the color opponency space, $A[i,j], R/G[i,j], B/Y[i,j]$ (the coordinates of the color of pixel (i,j)). If the image has N colors, we call COL_k, with $1 \leq k \leq N$, the set of pixels (i,j) of I that have the same color (i.e. the same coordinates $A[i,j] = x_k, R/G[i,j] = y_k, B/Y[i,j] = z_k$ in the color space.) For each k we randomly choose a ray of the color sphere centered in (x_k, y_k, z_k) and consider the plane $\pi[k]$ perpendicular to the ray and passing through its middle point. We "move" all the pixel in COL_k to the end of the ray on the opposite side of the plane $\pi[k]$ (that is, we change the representation of the colors of pixels $(i,j) \in COL_k$ to $A'[i,j] = x'_k, R/G'[i,j] = y'_k, B/Y'[i,j] = z'_k$ where (x'_k, y'_k, z'_k) is the other endpoint of the ray).

The result of this process is the watermarked image. The stored mark [1] is the vector of planes $\pi[k]$

A pseudocode description of the insertion algorithm appears in Figure 1.

[1] It is possible to reduce the size of a mark to a short random string by using strong *pseudo-random number generators* (see [21]). We use a PRNG seeded with a short random string s to generate plane coefficients in the color space. Clearly the stored mark can just be s. When the detection algorithm is run the vector $[\pi[1], \ldots, \pi[N]]$ can be reconstructed using the same process.

Mark-Insert

Input: An Image I, given as three matrices in the color opponency space $A[i,j], R/G[i,j], B/Y[i,j]$ where (i,j) is a single pixel.

Output:
A marked image I' given as $A'[i,j], R/G'[i,j], B/Y'[i,j]$.
A *mark* given as a vector $\pi[k]$, $k = 1, \ldots, N$ where N is the number of colors in the image. Each $\pi[k]$ is a plane in the color space.

1. For each $k = 1, \ldots, N$ where N is the number of colors
   (a) *Classify pixels by color*
   Set COL_k the set of pixels (i,j) that have color k
   Let x_k, y_k, z_k be the point in the color space associated with color k.
   (b) *Select a random direction of motion*
   Select a random ray in the color sphere centered in (x_k, y_k, z_k). Let (x'_k, y'_k, z'_k) be the other endpoint of the ray.
   (c) *Move all pixels in COL_k to the same point in the color opponency space*
   For each pixel $(i,j) \in COL_k$
   Set $A'[i,j] = x'_k, R/G'[i,j] = y'_k, B/Y'[i,j] = z'_k$
   End For.
   (d) *Save the plane normal to the direction of motion*
   Set $\pi[k]$ to be the normal plane to this ray and passing through its middle point.
   End For.
2. Return $I' = [A'[i,j], R/G'[i,j], B/Y'[i,j]]$ as the marked image.
   Save $MARK = [\pi[1], \ldots, \pi[N]]$ as the mark.

Fig. 1. Watermark Insertion Algorithm

4.2 The Detection Algorithm

Recall that a detection algorithm takes as input a received marked image, the original image and the list of stored marks.

Let us assume for the moment that the received image has the same number of colors as the original image. The detection algorithm compares the image to each stored mark. When comparing against $MARK = [\pi[1], \ldots, \pi[N]]$, the basic idea is to look at each class COL'_k of the received marked image and check if the color point x'_k, y'_k, z'_k associated with it is located on the opposite side of the plane $\pi[k]$ (where opposite is defined with respect to the original center for COL_k, i.e. (x_k, y_k, z_k)). If this is the case we increase a counter.

At the end, after comparing the image with all the stored marks, we output the mark that scored the highest counter.

A pseudocode description of the detection algorithm appears in Figure 2. In Section 5 we present a statistical analysis that proves that with very high probability this algorithm identifies the correct mark.

Remark 1. Note that the detection algorithm could receive an image manipulated by a malicious adversary. Such an adversary is not required to keep the same number of colors in the image. In particular the adversary could "move" each pixel in COL_k to a different location in the color space. However notice that in order not to deteriorate the image the adversary must keep each pixel in COL_k inside the color sphere associated with it. Then it is easy to reduce this case to the case that the received image still has the same colors. One possibility is to "recompact" the pixels moved by the adversary to their "baricenter". Another possibility is to increment the counter whenever a large enough quorum of pixels belonging to the same class is on the correct side of the plane. So in the following we can safely assume that the received image will have N colors and the classes COL_k are the same as in the original image. The only difference is that the representative coordinate point for the class COL_k will be different in the marked image.

5 Statistical Analysis

In this section we show that the proposed algorithms work. We first show that a marked imaged that has undergone no transformation at all will be recognized uniquely with very high probability (that is we show that our algorithms do not create false positive or false negative errors).

We then consider an adversarial model in which the marked image is processed by an adversary who is trying to erase the watermark. We first argue that such an adversary does not have enough information to mount an effective attack and the only thing that it can do is to move the pixels of the marked image in a random fashion inside their color spheres. Then we show that with very high probability our algorithms will resist such adversarial strategy and the marked image will still be uniquely identified.

Remark 2. For simplicity's sake we first carry on the analysis in a two-dimensional color space rather than the three-dimensional one. That is we assume that each color

Mark-Detect

Input:
The original image $I = [A[i,j], R/G[i,j], B/Y[i,j]]$.
A received marked image $I' = [A'[i,j], R/G'[i,j], B/Y'[i,j]]$.
The list of stored marks $MARK_\ell = [\pi_\ell[1], \ldots, \pi_\ell[N]]$ for $\ell = 1, \ldots, M$ where M is the total number of images originally marked.
N the number of colors in the image.

Output: A mark $MARK_{id}$.

1. Set $max = 0$ and $id = 0$
2. For each $\ell = 1, \ldots, M$
   (a) Set counter $C_\ell = 0$.
   (b) For each color $k = 1, \ldots, N$
   Let (x'_k, y'_k, z'_k) be the coordinates of the color of the pixels of the marked image belonging to COL_k.
   Let (x_k, y_k, z_k) be the coordinates of the color of the pixels of the original image belonging to COL_k.
   If (x_k, y_k, z_k) and (x'_k, y'_k, z'_k) are on opposite sides of the plane $\pi_\ell[k]$ then increase C_ℓ by 1.
   End For
   (c) If $C_\ell > max$ then set $max = C_\ell$ and $id = \ell$
   End For.
3. Output $MARK_{id}$.

Fig. 2. Watermark Detection Algorithm

is a point in the plane and that it will be moved to the edge of a circle centered on it. The "planes" $\pi[k]$ will become straight lines normal to the ray along which the point has been moved and passing through its middle point. Although this gives us slightly weaker bounds on the error probability of our algorithm, it is much easier to understand the geometric intuition behind the analysis. In section 5.3 we show how to improve the error bounds by using the full three-dimensional model.

5.1 Identifying non-manipulated images

Let's assume that the marked image I' given to detection algorithm is the exact result of the application of the insertion algorithm to the original image I. That is, we assume that no other manipulation has been applied to the image.

Let $MARK_i$ be the correct mark that generated I' from I and let $MARK_j$ be any of the other incorrect marks.

Since the image was not manipulated in any way, the counter C_i resulting from comparing I' with $MARK_i$ will reach the value $C_i = N$. The detection algorithm will output $id = j$ only if also $C_j = N$.

When comparing I' with $MARK_j$, for each k the counter C_j will be increased iff the color (x'_k, y'_k, z'_k) in I' ended up on the opposite side of $\pi_j[k]$. This happens with probability 1/3 since the lines intersect the circles in a way that 1/3 of the edge is on the opposite side (see Figure A case 1). For each $k = 1, \ldots, N$ these are independent events, thus we conclude that $Prob[C_j = N] = 3^{-N}$.

The algorithm fails if there exists a $j \neq i$ such that the above will happen. Thus since there are $M - 1$ incorrect marks to be examined we have that the total probability of failure is bounded by

$$Prob[\ \textbf{Mark-Detect}\ \textit{fails}\ ] \leq (M-1)\ 3^{-N}$$

5.2 Identifying manipulated images

In this section we assume that an adversary has manipulated a marked image I' in an effort to remove the watermark embedded in it.

First of all let's try to understand what kind of attack can the adversary mount. When given a marked image, the adversary does not know in which direction colors have been moved. Indeed this information is part of the secret key used to generate the mark.

Thus the only thing that the adversary can do is to move each color in a random fashion trying to "undo" the effect of the watermark. As a first approximation let us assume that the adversary moves pixels with the same color in the same way. That is, it makes the same changes for each pixel in COL_k inside its associated color sphere (indeed the adversary is limited to move things inside the sphere, otherwise the image is deteriorated). The interesting thing is that now for each color the adversary "sees" a different color sphere associated with it, namely the one centered in (x'_k, y'_k, z'_k). For each color the probability of "undoing" the mark is 1/3. This is because the biggest part of the "new" color circle (actually 2/3 of it) lies on the other side of the line w.r.t. to the original color location, thus the probability that a random motion brings the point to the correct side of the plane is 1/3 (see Figure A case 1).

Another thing that we have to make sure is that by "moving" these colors the adversary does not cause an increase of the counter for an incorrect mark. As we will see, this event will also happen with sufficiently small probability for each color.

Remark 3. It is sufficient to consider the above adversarial strategy for our purposes. Indeed (as we already mentioned in Remark 1, Section 4.2), the adversary could also "move" each pixel in COL_k in a different direction of the color space. However all these locations should be inside the color sphere. Thus it would be sufficient for the detection algorithm to look at the "baricenter" of all the pixels inside this sphere associated with COL_k and increase the counter if this point is on the opposite side of the plane. An equivalent approach would be to increase the counter only if a large quorum of the pixels fall on the opposite side of the plane. Roughly speaking such an attack will be effective if it concentrates as many pixels as possible on the correct side of the plane $\pi[k]$. Which is basically as hard as guessing where the plane is and moving *all* the pixels on the other side of it.

SOME PROBABILITY TOOLS. Our analysis uses heavily the following inequality due to Hoefding (see [14]).

Let $Z_1, \ldots, Z_N$ be independent, identically distributed (i.i.d.) random variables, each ranging over the interval $[a, b]$, and let μ denote their expected value. Then,

$$Prob\left[\left|\frac{\sum_{i=1}^{N} Z_i}{N} - \mu\right| \geq \delta\right] < 2e^{-\frac{2\delta^2 N}{b-a}} \tag{3}$$

Again let $MARK_i$ be the correct mark and $MARK_j$ be any of the other incorrect marks. The statistical analysis will work by considering the counters C_i and C_j kept by the detection algorithm as random variables. The algorithm fails when $C_j \geq C_i$, i.e. when $S = C_j - C_i \geq 0$. We show that S can be written as the sum of N i.i.d. random variables whose expected value is a negative number μ. Then we can just apply Equation (3) with $\delta = -\mu$ to get a bound on the probability that $S \geq 0$.

Consider the random variable X_k defined as follows: $X_k = 1$ if when analyzing color k in the detection algorithm we add 1 to the counter C_i, otherwise $X_k = 0$. Clearly $C_i = \sum_{k=1}^{N} X_k$. Similarly define random variable Y_k as follows: $Y_k = 1$ if when analyzing color k in the detection algorithm we add 1 to the counter C_j, otherwise $Y_k = 0$. Clearly $C_j = \sum_{k=1}^{N} Y_k$. Thus $S = \sum_{k=1}^{N} Z_k$ where $Z_k = Y_k - X_k$. All we are left to do is to estimate the distributions of X_k and Y_k.

ESTIMATE FOR X_k. When analyzing the correct mark $MARK_i$ we claim that for each color the counter C_i is increased with probability $2/3$, that is $Prob[X_k = 1] = \frac{2}{3}$. Indeed in order for $X_k = 1$ we need that the adversary moved the marked color on a location that is still on the opposite side of the line $\pi_i[k]$. But since the color circle associated with this color lies for 2/3 on that side of $\pi_i[k]$ (see Figure A case 1) and the adversary chooses the direction of motion at random we have the above probability.

ESTIMATE FOR Y_k. In this case the analysis is slightly more complicated. In order for $Y_k = 1$ we need that the adversary moved the marked color on a location that is still on the opposite side of the line $\pi_j[k]$ (where $MARK_j$ is the incorrect mark being

analyzed). We analyze the probability that this happens based on the location of the line $\pi_j[k]$.

Referring to Figure A, we partition the original color circle in four areas:

1. Assume that the line $\pi_j[k]$ is ortogonal to a ray contained inside the angle a. This happens with probability $1/3$ since $a = \frac{2\pi}{3}$. In this case at most $2/3$ of the color circle of the "marked" point lies on the opposite side of $\pi_j[k]$. Thus the probability that the adversary moves the point to the opposite side of $\pi_j[k]$ in this case is at most $2/9$.
2. Assume that the line $\pi_j[k]$ is ortogonal to a ray contained inside the angle b. This happens with probability $1/6$ since $b = \frac{\pi}{3}$. In this case at most $1/2$ of the color circle of the "marked" point lies on the opposite side of $\pi_j[k]$. Thus the probability that the adversary moves the point to the opposite side of $\pi_j[k]$ in this case is at most $1/12$.
3. Symmetrically the same probability $1/12$ is obtained when the line $\pi_j[k]$ is ortogonal to a ray contained inside the angle d.
4. Assume that the line $\pi_j[k]$ is ortogonal to a ray contained inside the angle c. This happens with probability $1/3$ since $c = \frac{2\pi}{3}$. But in this case the line $\pi_j[k]$ does *not* intersect the color circle so the adversary will never manage to move the point on the opposite side of it. Thus the probability in this case is 0.

The above cases are mutually exclusive and cover all possibilities, thus we can say that $Prob[Y_k = 1] \leq \frac{2}{9} + 2\frac{1}{12} = \frac{7}{18}$. In the following we assume that the adversary is actually stronger and gets $Y_k = 1$ exactly with probability $7/18$ (this means that in practice the failure probability is smaller than the obtained bounds)

PUTTING IT ALL TOGETHER. Now we can estimate the distribution of Z_k. Given that X_k and Y_k are indendent random variables we have that

$$Z_k = \begin{cases} -1 & \text{with prob. } 11/27 \\ 0 & \text{with prob. } 25/54 \\ 1 & \text{with prob. } 7/54 \end{cases}$$

Thus $\mu = E[Z_k] = -5/18$. We can now apply Equation (3) with $b = 1$, $a = -1$, and $\delta = -\mu$:

$$Prob[S \geq 0] \leq e^{-\mu^2 N}$$

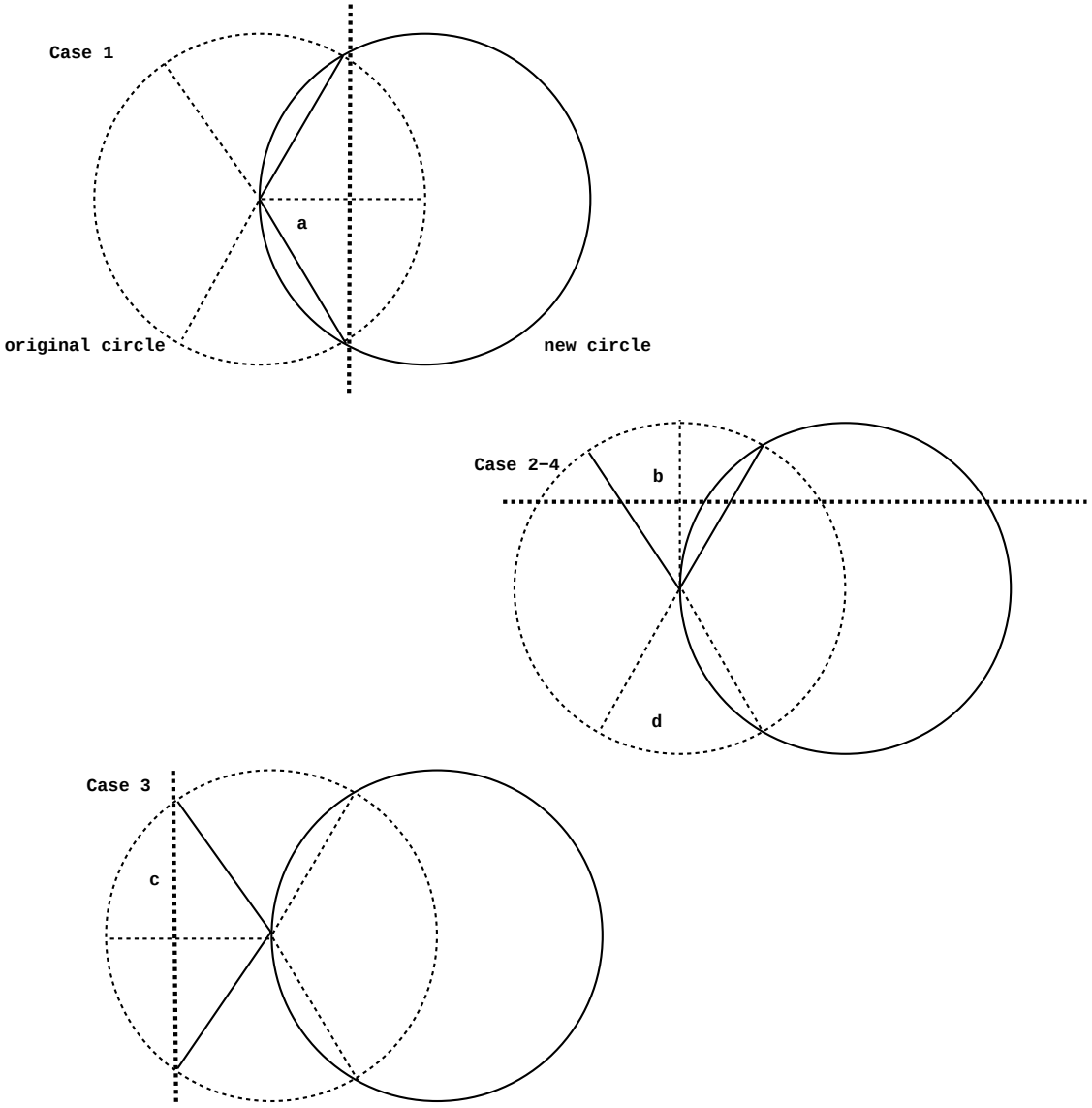
As in the previous case the probability of failure is that there exists at least one incorrect mark that causes $S \geq 0$. Thus

$$Prob[\ \textbf{Mark-Detect}\ \textit{fails}\ ] \leq (M-1)\ e^{-\mu^2 N}$$

5.3 Improving the analysis

The statistical analysis in the previous section was simplified to a two-dimensional model in order to make it more intuitive. However the full three-dimensional model allows us to obtain stronger bounds on the probability of error.

FIGURE A



In this section we simply sketch how to generalize the two-dimensional arguments to the case in three dimension and provide the stronger bounds. The final version of this paper will contain all the missing details.

First of all let's recall some fact of basic geometry. A sphere of ray R has area $4\pi R^2$. If we take a plane perpendicular to a ray and distant h from the center, the area of the spherical cap on the opposite side of the plane from the center is $2\pi R(R-h)$. Since in our case we choose a plane passing through the middle point of the ray, we obtain that the area of the spherical cap on the opposite side of the plane is πR^2, i.e. $1/4$ of the total.

This immediately generalizes the bound on the probability of error in the case of images non-manipulated. In that case we can strenghten the upper bound on the probability of failure to $(M-1)4^{-N}$.

For the case of images manipulated by an adversary we use again the same notation as before. In this case $Prob[X_k = 1] = 3/4$ since to a random motion of the marked point "undoes" the mark only if it brings the point on the above spherical cap. When analyzing incorrect marks we can argue that $Prob[Y_k = 1] < 7/16$, by generalizing the case-by-case analysis done in the two-dimensional case. Thus $\mu = E[Z_k] < -5/16$ in this case (and the probability of failure still bounded by $(M-1)e^{-\mu^2 N}$.)

THE PRACTICAL MEANING OF THIS ANALYSIS. The first thing to notice is that the error probability goes down exponentially with the number of colors in the image. For non-manipulated images the theoretical bounds are already very strong. For manipulated images the theoretical bounds start becoming meaningful for images ranging around one thousand colors.

For images with, say, $N = 256$ colors the theoretical bound says that the probability of failure is $(M-1)e^{-25}$ which is clearly not strong enough. However one must remember that in the analysis we were quite "generous" with upper bounding. In practice the probability of failure is much smaller than that and our experimental results seem to confirm this.

THRESHOLD DETECTION. The algorithms assume that the image being analyzed by the detection algorithm has been generated by one of the marks in the list MARK. If this is not the case (for example the image arrives from another vendor) then the algorithm, as is, will still identifies a mark from the list as the correct one. In order to avoid this problem it is sufficient to modify the algorithm so that it accepts an identification as correct only if the counter C_{id} of the selected mark is higher than a given threshold T. A statistical analysis similar to the one described above (and not reported here for space limitations) show that by setting the threshold to $T \approx N/2$ one obtains bounds on the failure probability comparable to the current ones. The value $T \approx N/2$ is also justified by the experimental results described in the next section.

6 Experimental results

A first version of the proposed marking and detection algorithm has been implemented in MATLAB 5.0. This has been done for sake of fast prototyping. On the other side MATLAB image processing libraries works only on 256 colors pictures.

The first kind of experiments that we have performed concerned the invisibility of the watermark inserted in the picture. The overall quality of a picture is not affected, both for photographs and for synthetic pictures.

Example images for this section can be found at this URL
`http://www.dipmat.unict.it/~anile/FuzzyArith/home/papers.htm`

Small differences comes out only on very high quality monitors and just at the highest magnifications.

The second kind of experiments that we have performed have been aimed to tune the detection algorithm. We call a color that has been recognized as marked a *positive color*. For a given mark M, we are interested in determining a threshold value T such that if more than T positive colors have been found then, with a high probability, the image comes, perhaps after some manipulation, from an image containing mark M. To this aim we have taken an ensemble of about 200 unmarked pictures and submitted them to the detection algorithm in search of a mark M. The algorithm gave an average of about 38% of positive colors, with a variance of $\sigma^2 = 7.5\%$. As a safe rule of thumb, hence, we suggest to classify as definitively suspect an image such that more than $T = (38 + 2\sigma^2)\%$ colors have been recognized as marked. With this safely set threshold we have observed no false positive (unmarked images detected as marked) in our set.

The third kind of experiments that we have done uses a small "library" of 15 marks. We have submitted to the detection algorithm an ensemble of 100 pictures. Of these, 33% were marked with marks from the library, 33% were unmarked, 33% were marked with a mark outside of the library. All the unmarked images have been recognized as such using the threshold discussed above relatively to every marks in the library. Of the 33% of images that have been stained with a mark outside of the library all have been classified as unmarked. Of the remaining group all the pictures have been recognized as marked because the number of colors in agreement with a watermark exceeded the threshold T. Moreover, in total agreement with the theoretical analysis reported in the previous Section, the threshold has been trespassed only relatively to the original inserted watermark. Results are also summarized in the following table, where the number n indicates the number of positive colors detected both on images marked with marks outside the library (denoted with M1) and marked with marks of the correct library (denoted with M2).

	$0 < n \leq T/2$	$T/2 < n < T$	$n \geq T$
Unmarked	25	8	0
Marked M1	20	13	0
Marked M2	0	0	33

We have repeated the previous experiment using the same marks library, with another ensemble of 200 pictures where, this time, 50% of them were unmarked, and 50% of them were marked with a random mark from the library and successively manipulated with a random combination of 4 of the following operators: trimming/cropping, geometrical distortion, scaling and rotation, equalization, contrast stretching, median and gaussian filtering (with a small kernel). All this operation has been realized using the standard *Stirmark 3.0* package (see [16], [24] for more details). The detection algorithm has correctly classified all the unmarked images as such. Of the marked images

almost all have been correctly recognized as marked with the correct mark, although the value observed for the counters were much closer (from above) to the threshold than in the previous experiment. Only in one case a marked image produced a counter value slightly below the threshold, without attaining, in such a case, the maximum value of the counter for the correct mark. Results are summarized in the next table.

	$n < T$	$T < n < 3/2T$	$n > 3/2T$	Correct Mark identified
Marked	8	70	22	99
Unmarked	100	0	0	—

Finally we have simulated some malicious attacks oriented to remove the mark from an image. In order to do so we are presented with the following alternatives:

i) To randomly change all colors in the picture palette by a large amount;
ii) To randomly change all colors in the picture palette by a small amount;
iii) To randomly change a portion of all colors in the picture palette by a large amount;
iv) To randomly change a portion of all colors in the picture palette by a small amount;

The mark becomes difficult to detect in case i), but in this case the picture is visibly degraded from the original: although almost mark free the picture is now close to useless if fidelity to the original is an important issue.

The strategy in case ii) seems to produce the best results for a malicious attack. In this case, indeed, we observed an attenuation of the watermark with an almost imperceptible degradation of the picture. In all of our experiments, however,the correct watermark has always been recognized. In case iii) the mark has been still recognized in all our experiments and at the same time frequently the image could be visibly degraded. In case iv) the mark is recognized by our detection algorithm with a slight decrease of the maximum number of positive colors detected.

Lately, we have implemented the two algorithms in C++ to better test the method with images having more than 256 colors.
Early experimental results showed its robustness against requantization attacks [2]. In particular we noticed that the number of recognized colors is "proportional" to the number of colors of the final, requantized image (the less is the number of colors of the requantized image, the less is, clearly, the number of recognized colors).
However, we point out, that the number of positive colors remains more than T even for "strong" requantizations (from 16 million to 256 colors).

7 Conclusions and future works

In this paper we have presented a new watermarking scheme that works moving the colors of a picture inside the CO color space according to a predefined scheme. The mark inserted in this way can be easily detected. Its statistical properties have been analyzed and the dependence of its robustness to malicious attack is shown to be dependent on the number of colors in a picture. Early experimental results show that the proposed method is resistant to several attacks, under different strategies, moreover the mark is

[2] Requantization attacks have been simulated using Stirmark software

resistant to geometrical transformation, equalization and smoothing. Future works includes to explore the importance of the color space in granting a transparent and robust mark also under collusive-attacks.

Acknowledgments. The authors wish to thank the anonymous referees for their helpful suggestions and comments.

References

1. J.M.ACKEN How Watermarking Adds Value to Digital Content *Communications of the ACM* July 1998/Vol.41, No.7 pp.75-77
2. A.E.BELL, G.W.BRAUDAWAY, F.MINTZER Opportunities for Watermarking Standards *Communications of the ACM* July 1998/Vol.41, No.7 pp.57-64
3. W.BENDER, D.GRUHL, N.MORIMOTO Techniques for data hiding. *Proc. of SPIE* Vol. 2420, pag.40 1995
4. D.BONEH, J.SHAW Collusion-secure fingerprinting for digital data. *Proc. Advances in Cryptology - Crypto'95*, Springer Verlag LNCS no.963 pp.452-465, 1995
5. J.BRASSIL, S.LOW, N.MAXEMCHUK, L.O'GORMAN Electronic marking and identification techniques to discourage document copying. *Proc. of Infocom '94* pp.1278-1287, 1994
6. D.H.BALLARD, C.M.BROWN *Computer Vision*, Prenctice Hall, Inc. 1982
7. G.CARONNI Assuring ownership rights for digital images. *Proc. of reliable IT Systems, VIS'95* Vieweg Publishing Company, 1995
8. I.COX, J.KILIAN, T.LEIGHTON, T.SHAMOON A secure, robust watermark for multimedia. *IEEE Transaction on Image Processing*, Vol.6(12) pp.1673-1687, 1997
9. S.CRAVER, BOON-LOCK YEO, M.YEUNG Technical Trials and Legal Tribulations *Communications of the ACM* July 1998/Vol.41, No.7 pp.45-56
10. F.ERGUN, J.KILIAN, S.R.KUMAR A note on the limits of collusion-resistant watermarks *Proc. Advances in Cryptology - Eurocrypt '99*, Springer Verlag LNCS no.1592 pp.140-149, 1999
11. M.M. FLECK, D.A. FORSYTH, C. BREGLER *Finding Naked People*, Lectures in Computer Science, 1996
12. J.D.FOLEY, A.V. DAM, S.K. FEINER, J.F. HUGHES *Computer Graphics, Principles and Practice*, Addison Wesley, 1990
13. A.S. GLASSNER *Principle of Digital Images*, Morgan Kaufmann Publishers Inc., 1995
14. O.GOLDREICH Foundations of Cryptography (Fragments of a Book). Available on-line from `http://theory.lcs.mit.edu/~ oded/frag.html`.
15. E.KOCH, J.RINDFREY, J.ZHAO Copyright protection for multimedia data *Proc. of the Int. Conf. on Digital Media and Electronic Publishing* 1994
16. M.KUTTER, F.A.P. PETITCOLAS *A fair benchmark for image watermarking systems*, To in E. Delp et al. (Eds), in Vol.3657, proceedings of *Electronic Imaging'99, Security and Watermarking of Multimedia Contents*, San Josè, CA USA, 1999
17. C.LUO, E.KOCH, J.ZHAO In Business Today and Tomorrow *Communications of the ACM* July 1998/Vol.41, No.7 pp.67-72
18. B.M.MACQ, J.J.QUISQUATER Cryptology for digital TV broadcasting. *Proc. of IEEE*, 83(6) pp.944-957, 1995
19. K.MATSUI, K.TANAKA Video-steganography *IMA Intellectual Property Project Proceedings*, Vol.1, pp.187-206, 1994
20. N.MEMON P.WAH, WONG Protecting Digital Media Content *Communications of the ACM* July 1998/Vol.41, No.7 pp.35-43

21. A.MENEZES, P.VAN OORSCHOT, S.VANSTONE Handbook of Applied Cryptography.
22. A.N.NETRAVALI, B.G. HASKELL *Digital Pictures:Representation and compression*, Application of Communication Theory, Plenum Press, NY, 1988
23. C.I.PODILCHUK, W.ZENG *Image Adaptive Watermarking Algorithm Based on a Human Visual model*, *Signal Processing*, Vol.66, No.3, 1998, pp.337-355
24. F.A.P. PETITCOLAS, R.J. ANDERSON, M.G. KUHN *Attacks on copyright marking systems*, in David Aucsmith (Ed): Proceedings of *Information Hiding, Second International Workshop, IH'98*, LNCS 1525, Springer-Verlag, pp.219-239.
25. I.PITAS, G.VOYATZIS *Protecting Digital-Image Copyrights: A Framework*, IEEE Computer Graphics and Applications, Jan. 1999,Vol.19, No.1, pp.18-24
26. R.VAN SCHYNDEL, A.TIRKEL, C.OSBORNE A Digital watermark *Proceedings of ICIP* IEEE Press, 1994 pp.86-90
27. K.TANAKA, Y.NAKAMURA, K.MATSUI Embedding secret information into a dithered multi-level image. *Proc. 1990 IEEE Military Communications Conference*, pp.216-220, 1990.
28. M.YEUNG Digital Watermarking *Comm. of the ACM* July 1998/Vol.41, No.7 pp.31-33