A Universal Technique to Hide Traces of Histogram-Based Image Manipulations

M. Barni
Dept.of Information Engineering
University of Siena
Siena, IT
barni@dii.unisi.it

M. Fontani
Dept. of Information Engineering
University of Siena
Siena, IT
marco.fontani@unisi.it

B. Tondi
Dept. of Information Engineering
University of Siena
Siena, IT
benedettatondi@libero.it

ABSTRACT

We propose a universal counter-forensic technique for concealing traces left on the image histogram by any processing tool. Under the assumption that the forensic analysis relies on first-order statistics only (which is true in many practical applications), the proposed scheme allows the attacker to conceal traces left by any processing operation, while maintaining a high fidelity between processed and "cleaned" images.

Categories and Subject Descriptors

I.4 [**Image Processing**]: Miscellaneous

General Terms

Security

Keywords

Universal Counter Forensics, Histogram, Contrast Enhancement

1. INTRODUCTION

Multimedia (MM) Forensics is an emerging discipline that aims at revealing the history of digital contents (image, video, audio) using a blind approach. The idea at the basis of MM Forensics is that when a processing tool is applied to a digital content, a number of footprints are left into the media. Several methods have been proposed that leverage on these footprints to reach some conclusions on the past history of the object under analysis: there are techniques for integrity verification, source identification or classification, analysis of near-duplicates dependencies and many others (see [10] for a recent survey). Together with the continuous development of new forensic techniques, however, counter-forensic (CF) methods are being developed as well. As suggested by the name, counter-forensics aims at concealing the traces introduced by processing tools when the user edits/tampers a MM content. As it will be clarified in Section 2, existing approaches are mostly targeted at deceiving a specific detector: they exploit knowledge of the forensic algorithm and try to erase the traces it looks for. In doing so, they may introduce some other kinds of artifacts, that could be detected using different (perhaps more sophisticated) forensic tools. This can lead to a "cat-and-mouse" game where several iterations of the forensic/counter-forensic loop are carried out. It would be interesting, instead, to devise universal CF methods that give the attacker more warranties about the undetectability of the processing operations, at least under some assumptions.

In this work we propose a universal approach for concealing traces left in the image histogram. This is an extremely useful tool if we assume that the Forensic Analyst (FA) can only consider first order statistics to perform its tests (as, for example, in [16] and [17]), and that the Adversary (AD) must satisfy some requirements in terms of desired image quality. Under these assumptions we developed a counter forensic technique that is "universal" in the sense that:

- the AD does not need to know anything about the FA detection algorithms (apart from the fact that they are based on first-order statistics only);
- the AD can use the proposed technique, without any changes, to hide histogram traces introduced by any kind of processing tool.

In brief, the idea is that the AD will first process the image and then perform slight modifications on the resulting image so to bring the histogram as close as possible to that of another, original, unprocessed image, while respecting strict distortion constraints. Intuitively, if the AD manages to do so, the FA will be forced to classify both the original and the tampered images in the same way, thus committing either a false positive or a false negative error. Of course, this will hold only if the two images are no longer distinguishable based on the statistic the FA relies on (that is, image histogram).

The paper is structured as follows: in Section 2 a brief overview of existing counter-forensic techniques is given; then in Section 3 we sketch and present the proposed CF approach. Experimental results are reported in Section 4, showing the effectiveness of our approach for different kinds of processings.

MM&Sec'12, September 6–7, 2012, Coventry, United Kingdom.

2. COUNTER FORENSICS: A BRIEF OVERVIEW

The origins of counter-forensics trace back to a work by Kirchner et al. [7]: in that paper, the authors introduced the concept of fighting against image forensics, and proposed a method for resampling an image without introducing pixel correlations. It is worth noting that a simple yet important taxonomy was introduced in [7], were a first distinction is proposed between *post-processing* and *integrated* techniques, and between *targeted* and *universal* ones. In a nutshell, a counter-forensic technique belongs the post-processing class if it consists of two steps: first the attacker performs the tampering, thus obtaining a desired modified content, then she processes the content so to conceal/erase the detectable traces left during the first step. On the contrary, an integrated counter-forensic technique modifies the image so that *by construction* it does not expose detectable traces. It is easy to guess that, developing integrated methods is much harder in most cases. The second distinction regards the target of the counter forensic method: if it aims at removing the trace searched for by a specific detector, then it belongs to the targeted family. A universal method, instead, attempts to maintain as many statistical properties as possible, so to make the processed image hard to detect also with tools unknown to the AD.

Some time later, Cao et al. [4] proposed a targeted method to hide traces of contrast enhancement, a common enhancement operator that leaves traces in the histogram of the image, so to deceive the detector developed by Stamm et al. [16]. Cao's method is based on the introduction of local random dithering in the enhancement step, so it can be classified as integrated attack. Nevertheless, the authors also mention the possibility of turning this attack into a post-processing one.

Stamm et al. proposed several works for hiding traces of JPEG compression [18] [19], that also allow to hide some kinds of tampering that are revealed thanks to JPEG compression side effects [14]. The basic idea underlying these works is to remove an important trace left by JPEG compression into the image, namely the quantization of DCT coefficients. Since the goal is pursued by introducing additive noise to remove discontinuities in DCT coefficients values, these methods can be thought of as post-processing CF attacks. Lastly, JPEG counter-forensics incurs in a perceptual cost over the tampering result, that has been studied by Valenzise et al. in [20].

Counter-forensics is also entering the field of video: Stamm et al. faced this topic [13] providing a targeted method that allows to remove/add frames from a MPEG video without introducing statistical artifacts in the prediction error, a trace exploited in the detector introduced by Wang et al. [22] to detect video doctoring.

Very recently, counter forensic is moving towards more theoretical approaches: Barni [1] and Stamm et al. [15] proposed game theoretical formulations. In [1] the source-identification problem with known statistics is modeled as a zero-sum game played by two decision-makers: the FA, whose task is to perform classification through hypothesis testing, and the AD, who wants to perform the attack in such a way that FA's classification is deceived. Under the limited resources assumption for the analyst, the author derives the optimal strategies for the two players and then proves the existence of a Nash equilibrium for the game. Inspired by [1], in Section 3.2 we propose to use the divergence function as the objective function for the AD, since it is well known that the divergence is the statistical function which gives the appropriate measure of distinctiveness between two distributions [5].

As to [15], a framework is proposed to evaluate the probability that a forgery is detected assuming that both the AD and the FA play their optimal strategies.

3. A NEW UNIVERSAL COUNTER FORENSIC TECHNIQUE

When designing counter-forensic methods, it is always necessary to simultaneously consider the presence of, at least, two players: the Forensic Analyst and the Adversary. The goal of the FA is to devise a method (detector) that is able to tell apart untouched images from those that have undergone some (usually very specific) processing. In a realistic scenario, it is reasonable to assume that the FA has limited resources for performing measurements over the signal. In this paper, we focus on the case in which the FA can only consider first order statistics of the observed signal (barely speaking, the histogram of the image) and wants to classify images as original or modified.

The AD has a different goal: she wants to produce a processed image, having some desired characteristics, and do that in such a way that FA's tools will misclassify it as original. As stated in the Introduction, she can follow two possible strategies to achieve this goal: the integrated approach or the post-processing one. The latter scheme however, if correctly interpreted, is much more appealing from the point of view of generality: if the AD finds a general way to make the statistical characteristic of a processed image similar or equal to those of an untouched one, she will be able to re-use the same tool for concealing traces left by different processing tools[1].

Outline of the proposed scheme

Following the arguments given in the previous section, we opted for a post-processing approach, and devised a universal counter-forensic method that conceals traces left in the histogram of the processed image (see Figure 1). From now on, all images will be denoted with the underline notation, e.g. $\underline{x}$. We denote with $\underline{x}(i) \in \mathcal{I}$ the value of the i-th pixel of the image among the set of possible values $\mathcal{I}$, and use h_x to indicate the histogram of $\underline{x}$. To begin with, let us assume that the AD has already created the processed image $\underline{y}$, and that she has access to a set S of histograms of untouched images. Then the AD proceeds as follows:

1. *Histogram retrieval* (Section 3.1): among all histograms in S, find the one that is most similar to h_y, denote it with h_x;
2. *Histogram mapping* (Section 3.2): find the best way to modify h_y so to bring it as close as possible to h_x, while satisfying some constraints on the maximum distortion incurred by $\underline{y}$;
3. *Implementation of the mapping* (Section 3.3): actually change pixels in the image according to the histogram

[1]It is worth observing that the gain in generality may come at the expense of lower performance in terms of trace concealment.

mapping, keeping the perceptual distortion as low as possible.

3.1 Histogram retrieval

The goal of this phase is the following: given a (processed) image $\underline{y}$ with histogram h_y find the most "similar" histogram h_x among a set S. Of course, we also want to maintain the properties induced on h_y by the processing, otherwise the counter-forensic method would remove the benefits the AD is looking for.

To do so, we propose that the AD uses a constrained research over the set S, looking for histograms that minimize a chosen distance to h_y, while respecting a set of constraints that will vary according to the kind of processing. Among the various families of histogram distance functions we choose the χ^2 distance that, given two histograms P and Q, is defined as follows:

$$\chi^2 = \frac{1}{2}\sum_i \frac{(P_i - Q_i)^2}{(P_i + Q_i)}.$$

If we denote with Γ the set of original histograms that satisfy the constraints imposed by the AD, the histogram retrieval problem is solved by searching for an h^* such that:

$$h^* = \arg\min_{h_x \in \Gamma} \chi^2(h_x, h_y). \quad (1)$$

where the metric is always evaluated between normalized histograms (i.e. histograms obtained dividing the population of each bin by the total number of pixels).

Notice that, in this phase of the scheme, we are more interested in retrieving an histogram that is near to h_y from the "shape" point of view than from the statistical one. In fact, besides the constraints in Γ, we would like the retrieved histogram to have, say, the same number of modes of the target one; the statistical distinguishability of the processed and attacked histograms will be treated within the histogram mapping phase. However, it may happen that the best matching histogram according to the χ^2 distance is statistically too different from h_y, thus making the histogram mapping very expensive or even impossible (as will be discussed in Section 3.2). To face this fact, we retrieve the best K matching histograms from the database, and run the histogram mapping on all of them; among these K candidates, the one resulting in the best mapping (based on the value assumed by the objective function in the optimum) will actually be used.

The choice of χ^2 distance has several motivations: firstly this metric weighs the contribution of each bin based on its population, so the contribution due to very populated bins is mitigated. This perfectly suits the needs of the AD, since changing the value of pixels that are sparely present in the image will probably incur a high perceptual cost, while highly populated bins can be managed more smartly, balancing the perceptual impact of the change (see Section 3.3). Secondly, the χ^2 distance can be efficiently evaluated, and this is essential to our application since it allows the AD to search among a sufficiently large dataset of histograms. On the other hand, this metric does not take into account relationships between adjacent bins. More sophisticated cross-bin histogram distances like Earth Mover's Distance [11] or the Quadratic-Chi [9] could improve search results at the cost of a higher complexity; we leave the investigation of the benefits allowed by cross-bin distances for future work.

As a last consideration about histogram retrieval, we point out two important facts. The first is that the search is conducted directly on histograms, and not on images. This considerably reduces the size of the dataset (10.000 histograms can be represented with less than 10MB) and the search routine, since only the histogram of the processed image must be computed on-line. The second observation is that the goal of this phase has nothing to do with content based retrieval: the AD simply wants to know if an original image exists (no matter what its content is) whose histogram is not far from that of the processed one, but she is not interested in what is actually represented in the image.

3.2 Histogram mapping problem

Given the processed image $\underline{y}$ and an original histogram h_x coming from the reference histogram database, the AD wants to create an attacked image $\underline{z}$ that is similar to $\underline{y}$ but has an histogram as close as possible to h_x. This problem is similar to the Optimal Transport problem [21], where the goal is to find a transport map which moves a given distribution into another minimizing some cost function; actually, our case is a bit different since the AD does not need a perfect match between the two histograms.

For sake of clarity, we assume that all images have the same number of pixels n, we will relax this assumption later. Let $h_z(i)$ and $h_y(i)$ be the number of times the $i-th$ pixel value appears, respectively, in $\underline{z}$ and $\underline{y}$, and let $\nu_z(i)$ and $\nu_y(i)$ be the corresponding relative frequencies ($\nu_z(i) = h_z(i)/n$, $\nu_y(i) = h_y(i)/n$). In our framework the ν_y vector is known, since it is computed from the processed image $\underline{y}$, while the ν_z vector has to be found. We introduce a *displacement matrix* $N = \{n(i \to j)\}_{i=0\ldots255, j=0\ldots255}$, whose (i,j)-th element tells how many elements of the histogram should be moved from the i-th to the j-th bin.

The goal of the AD is to find the displacement matrix N^* that minimizes the divergence between h_z and h_x while

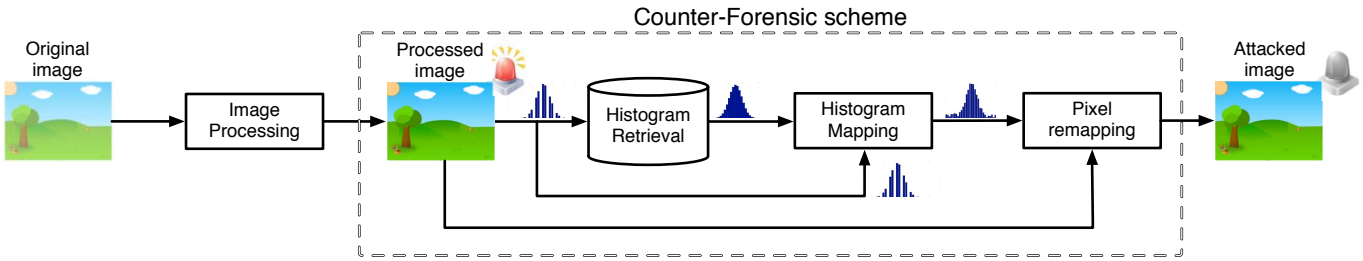


Figure 1: A schematic representation of the proposed universal counter forensic approach. Notice that, at least in the theoretical development, we are not interested about the specific processing carried by AD.

satisfying some constraints on the distance between $\underline{z}$ and $\underline{y}$. The divergence between two histograms is defined as:

$$D(\nu_z||\nu_x) = \sum_{i=1}^{255} \nu_z(i) \log \frac{\nu_z(i)}{\nu_x(i)}. \quad (2)$$

We choose this objective function because, since the divergence measures the statistical distinguishability of two distributions, minimizing this quantity turns out to be the optimal strategy for the adversary (see [1] for a thorough explanation). As to the distance between the images, since huge changes of pixel values would almost surely lead to annoying artifacts, we impose a maximum value D_{max} for the absolute pixel distortion:

$$\max_i |\underline{y}(i) - \underline{z}(i)| \leq D_{max}. \quad (3)$$

Furthermore, we must consider that the AD cannot move from each bin of h_y more elements than those actually available. This results in the following constraint:

$$h_y(i) = n(i{\to}i) + \sum_{k \neq i} n(i{\to}k) = \sum_k n(i{\to}k). \quad (4)$$

Eq. (4) suggests that also h_z can be easily written in terms of the elements of the displacement matrix:

$$h_z(i) = n(i{\to}i) + \sum_{k \neq i} n(k{\to}i) = \sum_k n(k{\to}i). \quad (5)$$

Substituting (5) in (2), we can rewrite the objective function in terms of the $n(i{\to}j)$ variables:

$$D(\nu_z||\nu_x) = \min_{n(i\to j)} \sum_{i=1}^{|\mathcal{I}|} \frac{(\sum_k n(k{\to}i))}{n} \cdot \log \frac{(\sum_k n(k{\to}i))}{n\nu_x(i)}. \quad (6)$$

and we can therefore rephrase the optimization problem as follows:

$$\min_{n(i\to j)} \sum_{i=1}^{|\mathcal{I}|} \frac{(\sum_k n(k{\to}i))}{n} \cdot \log \frac{(\sum_k n(k{\to}i))}{n\nu_x(i)} \quad (7)$$

subject to

$$\begin{cases} \sum_j n(i{\to}j) = h_y(i)\ \forall i \\ n(i \to j) = 0,\ \forall (i,j) \in \mathcal{I} : |i-j| > D_{max} \\ n(i{\to}j) \geq 0 \quad \forall i,j \\ n(i{\to}j)\ \in \mathbb{N} \end{cases} \quad (8)$$

where the second constraint is equivalent to eq. (3) and limits the maximum distortion per pixel.

The above optimization problem can be classified in the MINLP[2] class. Furthermore, since it can be proved (using the log-sum inequality [5]) that the objective function is convex in the $n(i{\to}j)$ variables, the problem is actually a *convex* MINLP problem [3], for which several efficient solvers [2] yielding the global optimum solution exist.

Some observations about the above problem are in order. The first is about the number of optimization variables, that is quadratic in $|\mathcal{I}|$. This means that the complexity of the problem does not depend on the size of the image, but only on its bit-depth (so we will usually have $|\mathcal{I}| = 256$). Furthermore, although it makes sense to consider only solutions for which one between $n(i{\to}j)$ and $n(j{\to}i)$ is equal to 0, it is not necessary to explicitly express this constraint, since the solutions for which this condition does not hold can be easily pruned after the optimization problem is solved. As a last consideration, we notice that in some cases the mapping may not be feasible: for example, suppose we have $D_{max} = 10$, histogram h_x is such that $h_x(j) = 0\ \forall\ j \in [128, 255]$ and histogram h_y has a peak in 250. There is no way to map h_y to h_x respecting the constraint about maximum distortion, so the problem is not solvable, and $D(\nu_z||\nu_x)$ is infinite. However, should this happen for one mapping, the fact that we solve K times this problem with different target histograms makes it very unlikely that we cannot find one admissible mapping.

[2]Mixed integer nonlinear problems

3.2.1 Generalization to arbitrary image size

Before moving to the last step of the CF scheme, we relax the hypothesis that the number of pixels in $\underline{x}$ and $\underline{y}$ is the same. Let $|\underline{x}|$ denote the number of pixels in image $\underline{x}$. As a matter of fact, since histogram retrieval is based on relative frequencies, most of the times we will have $|\underline{x}| \neq |\underline{y}|$. In order to generalize the problem without leaving the MINLP class we re-define h_x and h_y as follows:

$$h_x(i) = \nu_x(i) \times \mathsf{lcm}(|\underline{x}|, |\underline{y}|)$$

$$h_y(i) = \nu_y(i) \times \mathsf{lcm}(|\underline{x}|, |\underline{y}|)$$

where lcm denotes the least common multiple operator. This will simply require, after the optimization, to scale back the elements of the displacement matrix N by dividing each of them by $|\underline{x}|$: doing so, quantities in N will be referring to the number of pixels of $\underline{y}$, which is obviously the same for $\underline{z}$.

3.3 Pixel Remapping

After the target histogram h_z has been obtained, the AD needs to actually modify $\underline{y}$ into $\underline{z}$. All the operations performed in this phase will not affect the result of FA's forensic tools, since we assumed that they only consider the histogram of the image. Nevertheless, the AD is not interested in obtaining an attacked image $\underline{z}$ that is perceptually distant from the processed one $\underline{y}$. In this section we describe an approach that allows the AD to implement the pixel mapping defined by the displacement matrix N^* in a perceptually convenient way.

We begin by recalling that the human visual system (HVS) is known to be less sensitive to noise when this affects highly textured regions. On the contrary, noise in uniform regions, like the sky or a flat wall, is usually much more evident to the observer [23]. Therefore, the first intuition is that, whenever a choice is possible, regions of the image having high variance should be modified first. Furthermore it is useful to iteratively determine which parts of the image are more insensitive to noise through all the computation, using a kind of similarity map between the currently achieved image and $\underline{y}$. To compute this map, we adopt the Structural Similarity (SSIM) metric introduced by Wang et al. in [23]. This metric quantifies and localizes the structural similarity between two images, and provides a similarity value for each pixel; to determine this value, the system considers the contrast, brightness and other perceptually relevant information in the region surrounding the pixel. Since the image changes during pixel mapping, the map is evaluated several times in

order to allow a better (i.e. less perceptible) distribution of noise throughout the image.

Based on the above considerations we propose the following scheme, and comment it next:

1. Set all pixels as admissible
2. Compute a map of local variance[3] of $\underline{y}$;
3. For each couple (i, j):
   (a) find admissible pixels location having value i;
   (b) scan them selecting the first $n(i \rightarrow j)$ with higher values in the map;
   (c) substitute them with j;
   (d) remove selected pixels from the admissible ones[4];
   (e) if no more pixels of value i have to be remapped, compute the SSIM map between the current image and $\underline{y}$;

The first comment we make is about multiple computations of the similarity map: there is a clear tradeoff between computational complexity and perceptual fidelity. If we compute the map only once, then we do not take into account the distortion that is progressively introduced, and experimental results show that this can lead to annoying false-contouring artifacts. On the other hand, computing the SSIM after each single pixel substitution is clearly prohibitive (and useless). We think an excellent tradeoff is obtained by computing the map $|\mathcal{I}|$ times, specifically when no more pixels from the i-th level are left to move. Notice that for the first iteration we cannot resort to SSIM (which is a full-reference metric) to get a similarity map, because no changes have been performed still. Considering the HVS properties introduced before, we simply compute a map of the local variance of the image (working block-wise, with block size 5×5) and use it just for the first step.

While postponing a rigorous experimental validation to Section 4, we report in Figure 2 an example that shows the output of each of the steps described so far: the histogram of a contrast-enhanced image (notice the peak-and-gap artifacts) is fed to the histogram retrieval module, which returns the histogram yielding the lowest χ^2 distance in the DB. After pixel remapping ($D_{max} = 6$), the histogram of the attacked image is close to that of the original one, and the perceptual similarity between processed and attacked images is really satisfactory.

4. EXPERIMENTAL RESULTS

In this section we extensively evaluate the proposed counter forensic technique in a realistic scenario, and show that it yields excellent results in hiding traces (AUC of the detector before and after CF attack is evaluated) while retaining a very high quality for the attacked images (PSNR and SSIM are used for quality assessment).

4.1 Experiments scenario and setup

To provide an experimental validation we need to choose a specific scenario: this consists in selecting a detector for the FA and a (set of) processing operation for the AD. During the whole procedure, the AD can not exploit the knowledge of the detector used by the FA since we are aiming at a universal CF technique.

[3]SSIM cannot be evaluated before applying the first modification (see comments).

[4]This avoids multiple substitutions of the same pixel.

Experiments scenario

Among the image forensic algorithms based on first order statistics, probably the most popular is the one for detecting contrast enhancement, proposed by Stamm et al. in [16]. This tool exploits the fact that typical contrast enhancement techniques leave a characteristic fingerprint in image's histogram, namely the peak-and-gaps artifact. This effect is easily exposed in the frequency domain, where peak-and-gaps behavior results in an anomalous amount of high-frequency components. Therefore, by investigating the Fourier transform of image's histogram, the authors devised a very reliable detector.

From the AD point of view, we choose to implement two different techniques for contrast enhancement of grayscale images: one based on γ-correction and one based on histogram stretching. γ-correction enhancement is very simple, being fully described by the following equation:

$$\underline{y}(i) = 255 \times \left(\frac{\underline{x}(i)}{255} \right)^{\gamma} \tag{9}$$

where $\underline{y}$ denotes the enhanced image and $\underline{x}$ denotes the original one.

To formally define the histogram stretching operation, let us denote with l_{min} the gray level at the 1st percentile of the histogram and with l_{max} the gray level at the 99th percentile: then, we perform histogram stretching as:

$$\underline{y}(i) = 255 \times \frac{\underline{x}(i) - l_{min}}{l_{max} - l_{min}}. \tag{10}$$

Comparing Figure 2(a) and 2(b), the effect of histogram stretching in improving image quality is evident.

Since the AD wants to preserve the benefits induced by processing the image, he must define a constraint that filters the search for the best matching histogram. We adopt the Michelson definition of contrast [8], that for a given image histogram h is

$$c(h) = \frac{(h_{max} - h_{min})}{(h_{max} + h_{min})}$$

where h_{max} is the greatest non-empty bin and h_{min} is the lowest non-empty bin of h. Then, when searching in the set S of available untouched histograms, the AD defines the subset Γ of admissible histograms as:

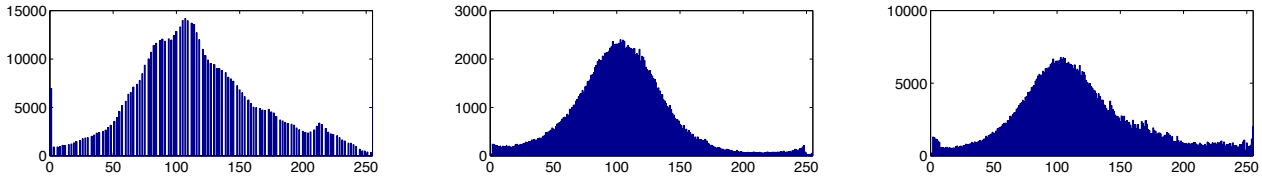
$$\Gamma = \{h \text{ in } S : c(h) \geq c(\bar{h})\}$$

where $\bar{h}$ is the histogram of the contrast-enhanced image, thus preventing the selection of target histograms having lower contrast than the one obtained with processing.

Experiments setup

We conducted our experiments by using images from the UCID dataset [12]. We also used another independent dataset, MIRFLICKR [6] (25.000 images), to prepare the database of untouched histograms. Throughout the experiments, all color images are converted to grayscale using the rgb2gray Matlab function. The only parameters the attacker has to choose are the number of candidates for which the optimization problem is solved (we use $K = 10$, each optimization runs in few seconds) and the maximum distortion per-pixel; of course, allowing a higher distortion will yield a more precise mapping of the attacked histogram to the one coming from the database but will also result in a lower quality.

(a) Untouched (b) Processed (c) Remapped



(d) Hist. of processed image (e) Hist. from DB (f) Hist. of remapped image

Figure 2: Top row: (a) an original image; (b) its processed (contrast stretched) version, and (c) the image resulting from the proposed CF technique. Bottom row: (d) histogram of the processed image, which is compared to those in the DB to find the best matching one (e), then the histogram mapping problem is solved yielding (f). Notice that the peak-and-gap artifacts in the left histogram have been removed in the right one.

We repeated the experiments with $D_{max} = 2$, 4 and 6 in order to investigate the relationship between distortion and effectiveness of the approach.

We evaluate the performance of the forensic method in [16] using the Area Under Curve indicator, and use the Peak Signal to Noise Ratio (PSNR) and SSIM index to assess the quality of the attacked image.

4.2 Results for γ-correction processing

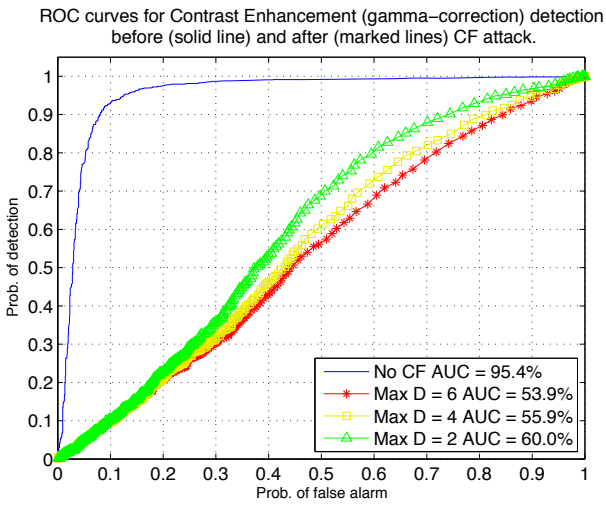
We performed contrast enhancement over all pictures in the UCID dataset according to eq. (9) and run the FA's detector on the resulting images; since values of γ very near to 1 would not result in a sensible modification, γ was chosen randomly from the set $[0.5; 0.8] \cup [1.2; 2]$. Then, we applied the proposed counter-forensic scheme on each processed image, for various D_{max}, and run again the detector. Figure 3(a) shows the ROC curves obtained for different values of maximum per-pixel distortion: we can state that the forensic detector no longer distinguishes untouched images from attacked ones even for $D_{max} = 2$. Experiments also confirm that, by allowing higher distortion, the AD can further hinder the performances of the detector. Of course, this fact alone is meaningless until we also investigate the fidelity of the attacked images to the processed ones: this information is reported in Figure 3(b); notice that PSNR is sufficiently high even for $D_{max} = 6$, and the SSIM index confirms an extremely low perceptual distortion. This confirms that the CF attack does not produce annoying artifacts, nor it removes the benefits introduced by the γ-correction.

4.3 Results for histogram stretching attack

Histogram stretching is a more intensive processing from a forensic point of view, in that it significantly modifies the histogram of the image. As in the previous experiment, we applied the processing described in eq. (10) to images of the UCID dataset, and plotted ROC curves in Figure 4(a). Though AD is using exactly the same CF attack scheme in front of a different processing (also the DB of untouched histograms remains the same), results are almost identical: for the FA's detector histograms of attacked images are no longer distinguishable from those of original images. Figure 4(b) confirms that, also in this experiment, attacked images are perceptually very near to the processed ones.

5. CONCLUSIONS

We have presented a universal counter forensic approach against detectors based on first-order statistics (image histograms). The approach belongs to the post-processing class of the CF attacks: after having processed the image, the AD uses the proposed technique to: i) search the best matching histogram (in a set of untouched ones) for the processed image; ii) solve an optimization problem for mapping the processed histogram into the retrieved one, satisfying some constraints on distortion; iii) actually modify (remap) pixels of the processed image, yielding an attacked image that is perceptually similar to the processed but has an histogram as close to the desired one as possible. Experimental results show the effectiveness of the proposed approach. Future work will focus on investigating the benefits that can be achieved: i) by using more sophisticated histogram similarity functions; ii) by merging the histogram search phase

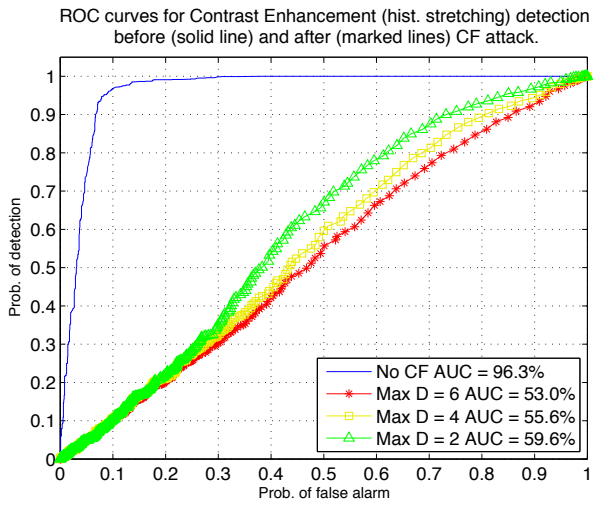


(a)

D_{max}	PSNR(db)	SSIM	AUC
2	44.8	0.994	0.600
4	39.3	0.981	0.559
6	36.2	0.964	0.539

(b)

Figure 3: Results for γ-correction counter-forensics. (a): ROC curves for Contrast Enhancement Detector running on γ-corrected images (solid line) and on attacked ones (marked lines); (b): mean values for PSNR and SSIM between γ-corrected and and attacked images, along with Area Under Curve obtained by the forensic detector. Experiments are carried on the UCID dataset



(a)

D_{max}	PSNR(db)	SSIM	AUC
2	44.9	0.994	0.596
4	39.2	0.984	0.556
6	36.1	0.971	0.530

(b)

Figure 4: Results for histogram stretching counter-forensics. (a): ROC curves for Contrast Enhancement Detector running on γ-corrected images (solid line) and on remapped ones (marked lines); (b): mean values for PSNR and SSIM between γ-corrected and remapped images, along with Area Under Curve obtained by the forensic detector. Experiments are carried on the UCID dataset

with the histogram mapping one; iii) by exploring connections with optimal transportation theory. In facing the mentioned issues, we will also consider the fact that the objective function we are using now (the Kullback-Leibler divergence) is proved to be optimal in the case of "known sources" (see [1]), while our case falls in the class of "known training sequences", since the attacker and the analyst do not know the probability density function of the unprocessed images; we will investigate if more appropriate distances exist for the latter scenario, and, if they exist, evaluate their impact on performances. Finally, we will test the method against different histogram-based detectors.

6. ACKNOWLEDGMENTS

This work was partially supported by the REWIND project funded by the Future and Emerging Technologies (FET) programme within the 7FP of the European Commission, under FET-Open grant number: 268478.

7. REFERENCES

[1] M. Barni. A game theoretic approach to source identification with known statistics. In *Proc. of ICASSP 2012, IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, 2012.

[2] P. Bonami, M. Kilinc, J. Linderoth, et al. Algorithms and software for convex mixed integer nonlinear programs. Technical report, Computer Sciences Department, University of Wisconsin-Madison, 2009.

[3] M. Bussieck and A. Pruessner. Mixed-integer nonlinear programming. *SIAG/OPT Newsletter: Views & News*, 14(1):19–22, 2003.

[4] G. Cao, Y. Zhao, R. Ni, and H. Tian. Anti-forensics of contrast enhancement in digital images. In *Proceedings of MM&Sec 2010, 12th ACM workshop on Multimedia and security (MM&Sec '10)*, 2010.

[5] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[6] M. J. Huiskes and M. S. Lew. The MIR Flickr retrieval evaluation. In *Proc. of MIR '08, ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2008. ACM.

[7] M. Kirchner and R. Böhme. Tamper hiding: Defeating image forensics. In *Proc of IH 2007, Int. Conference on Information Hiding*, pages 326–341, 2007.

[8] A. A. Michelson. *Studies in optics*. University of Chicago Press, 1927.

[9] O. Pele and M. Werman. The Quadratic-Chi histogram distance family. In *Proc. of ECCV 2010, European Conference on Computer Vision*, 2010.

[10] J. Redi, W. Taktak, and J.-L. Dugelay. Digital image forensics: a booklet for beginners. *Multimedia Tools and Applications*, 51:133–162, 2011. 10.1007/s11042-010-0620-1.

[11] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[12] G. Schaefer. An uncompressed benchmark image dataset for colour imaging. In *Proc. of ICIP 2010, IEEE Int. Conference on Image Processing*, pages 3537–3540, 2010.

[13] M. Stamm and K. Liu. Anti-forensics for frame deletion/addition in mpeg video. In *Proc. of ICASSP 2011, IEEE Int. Conference on Acoustics, Speech and Signal Processing*, pages 1876 –1879, 2011.

[14] M. Stamm, S. Tjoa, W. Lin, and K. Liu. Undetectable image tampering through jpeg compression anti-forensics. In *Proc. of ICIP 2010, IEEE Int. Conference on Image Processing*, pages 2109 –2112, 2010.

[15] M. C. Stamm, S. Lin, and K. J. R. Liu. Forensics vs. anti-forensics: A decision and game theoretic framework. In *Proc. of ICASSP 2012, IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, 2012.

[16] M. C. Stamm and K. J. R. Liu. Blind forensics of contrast enhancement in digital images. In *Proc. of ICIP 2008, IEEE Int. Conference on Image Processing*, pages 3112–3115, 2008.

[17] M. C. Stamm and K. J. R. Liu. Forensic detection of image manipulation using statistical intrinsic fingerprints. *IEEE Transactions on Information Forensics and Security*, 5(3):492–506, 2010.

[18] M. C. Stamm and K. J. R. Liu. Wavelet-based image compression anti-forensics. In *Proc. of ICIP 2010, IEEE Int. Conference on Image Processing*, pages 1737–1740, 2010.

[19] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu. Anti-forensics of JPEG compression. In *Proc. of ICASSP 2010, IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, pages 1694–1697, 2010.

[20] G. Valenzise, V. Nobile, M. Tagliasacchi, and S. Tubaro. Countering jpeg anti-forensics. In *Proc. of ICIP 2011, IEEE Int. Conference on Image Processing*, pages 1949–1952, 2011.

[21] C. Villani. *Topics in optimal transportation*. American Mathematical Society, 2003.

[22] W. Wang and H. Farid. Exposing digital forgeries in video by detecting double MPEG compression. In *Proc. of MM&Sec 2006, 8th ACM workshop on Multimedia & Security*, pages 37–47, 2006.

[23] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.