A Universal Attack Against Histogram-Based Image Forensics

Mauro Barni, Marco Fontani*, Benedetta Tondi

Dept.of Information Engineering and Mathematical Sciences,
University of Siena,
Via Roma 56, 53100 Siena, IT

barni@dii.unisi.it - marco.fontani@unisi.it - benedetta.tondi@unisi.it

In this paper we propose a universal image counter-forensic scheme that contrasts any detector based on the analysis of the image histogram. Being *universal*, the scheme does not require knowledge of the detection algorithms available to the forensic analyst, and can be used to conceal traces left in the histogram of the image by *any processing tool*. Instead of adapting the histogram of the image to fit some statistical model, the proposed scheme makes it practically identical to the histogram of an untouched image, by solving an optimization problem. In doing this, the perceptual similarity between the processed and counter-attacked image is preserved to a large extent. The validity of the scheme in countering both contrast-enhancement and splicing- detection is assessed through experimental validation.

Keywords: Image Forensics, Universal, Counter Forensics, Tamper Hiding, Histogram, Contrast Enhancement

Images have always played an important communicative role, probably due to their immediacy and presumed objectivity. The advent of digital imaging further increased this trend, since acquiring and sharing photos is nowadays cheap and fast. However, creating forgeries and editing photos has also become much easier, thus raising some doubts about the reliability of what we see through digital images.

As an answer to this growing inconvenience, Multimedia (MM) Forensics is emerging as a discipline that aims at revealing the history of digital contents (image, video, audio) using a blind approach. Unlike active techniques like digital watermarking or fingerprinting, MM Forensics does not assume that the content is generated or controlled by the subject that will have to ensure its authenticity. Instead, the idea at the basis of MM Forensics is that almost every step typically undergone by a digital content (e.g. acquisition, encoding, or application of processing operators) leaves a number of footprints into the media. By leveraging on these footprints, several methods have been proposed 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 (Redi, Taktak, & Dugelay, 2011) 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, the goal of counter-forensics is to conceal the traces introduced by processing tools when the user edits/tampers a MM content, so to make his actions undetectable. As it will be clarified in the following section, existing approaches are mostly targeted at deceiving a specific detector: they exploit knowledge of the forensic algorithm to erase the traces it looks for, of course limiting the perceptual impact of the modifications. In doing so, however, they may introduce new artefacts, 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 paper we extend our previous work in (Barni, Fontani, & Tondi, 2012), proposing a universal approach for concealing traces left in the image histogram by any processing operator. Compared to that work, we cast the counter-forensic scheme into the more appropriate theoretical framework provided in (Barni & Tondi, 2012), that better fits the realistic scenario we are considering.

The proposed tool is extremely useful whenever we can assume that the Forensic Analyst (FA) only considers first order statistics to perform its tests, as, for example, in (Stamm & Liu, 2008) and (Stamm & Liu, 2010), and that the Adversary (AD) must satisfy some requirements in terms of desired image quality. Under these assumptions we develop 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), and that the AD can use the proposed technique, without any changes, to hide histogram traces introduced by any kind of processing tool. Specifically, the AD will first process (or tamper with) 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 content 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 organized as follows: first we give a brief overview of existing counter-forensic techniques; then we sketch and present the proposed CF approach. In the experimental result section we evaluate the performance of the method both from a universal point of view and, as a case study, in countering a specific state of the art forensic algorithm (Stamm & Liu, 2008). Finally, we also evaluate the impact of the proposed scheme in hiding traces left in the pixel histogram during the creation of realistic forgeries.

Previous works in counter forensics

Counter-forensics was firstly introduced in a seminal work by Kirchner and Böhme (2007), where the concept of fighting against image forensics was introduced together with a practical application, namely a method for resampling an image without introducing pixel correlations. Furthermore, a simple yet important taxonomy was introduced in (Kirchner & Böhme, 2007) distinguishing between *post-processing* and *integrated* techniques, and between *targeted* and *universal* ones. Briefly speaking, counter-forensic techniques in the post-processing class consist 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 opposite, an integrated counter-forensic technique modifies the image so that *by construction* it does not introduce detectable traces. Of course, developing integrated methods is much harder in most cases. A second distinction is based on 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.

Cao et al. (2010) proposed a targeted method to hide traces of contrast enhancement, a common image processing operator that leaves traces in the image histogram, so to deceive the detector developed by Stamm and Liu (2008). The method is based on the introduction of local random dithering in the enhancement step, so it can be classified as an integrated attack. Nevertheless, the authors also mention the possibility of turning this attack into a post-processing one.

Several works have been proposed by Stamm et al. to hide traces of JPEG compression (Stamm, Tjoa, Lin, & Liu, 2010; Stamm et al., 2010), that also allow to hide some kinds of tampering that are revealed thanks to JPEG compression side effects. The basic idea is to remove the most 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. However, introducing noise has obviously a cost in terms of quality, as Valenzise et al. (2011) show.

Counter-forensic methods for video have also been proposed: Stamm and Liu proposed a method (2011) 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 and Farid (2006) to detect video doctoring.

With the goal of devising a more theorethical and general formulation for counter-forensics, Stamm et al. (2012) and Barni and Tondi (2013) recently proposed frameworks based on game theory. Stamm et al. (2012) propose a framework to evaluate the probability that a forgery is detected assuming that both the AD and the FA play their optimal strategies. In (Barni & Tondi, 2013) the source-identification problem with known statistics is modelled as a zero-sum game played by the AD and the FA: the task of the FA is to perform classification through hypothesis testing, while the AD wants to perform the attack in such a way that FA's classification is deceived. Under the limited resources assumption for the analyst, the authors derive the optimal strategies for the two players and prove that the correspondent profile is the Nash equilibrium for the game. A considerable step forward in this direction is made in (Barni & Tondi, 2012), in which the known statistics assumption is removed. This provides the appropriate theoretical framework for casting the problem faced in (Barni et al., 2012), allowing us to derive the approach proposed in this paper.

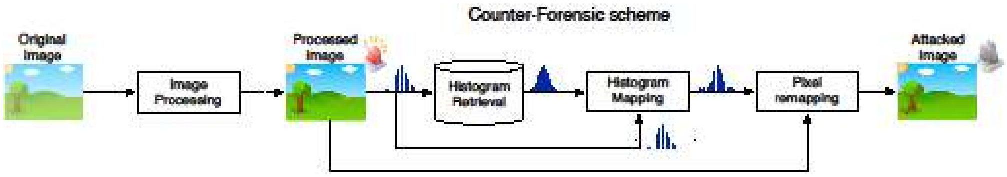


Figure 1: a schematic representation of the proposed universal counter forensic approach. Notice that, at least in the theoretical development, we are not interested about the specific processing carried by AD.

The proposed Universal Counter Forensic technique

Whenever an adversary aiming at impeding the forensics analysis is present, the forensic problem can be seen as a struggle between the analyst and the adversary. Then, the interplay between these two players should be modeled and studied for designing proper forensics and counter-forensics methods (Barni & Tondi, 2013). As suggested by its name, the goal of the FA is to devise a detector that is able to tell apart untouched images from those that have undergone some processing. In a realistic scenario, we can reasonably assume that the FA has limited resources for analyzing the signal. In this paper, we focus on the case in which the FA can only consider first order statistics of the observed signal, i.e. the histogram of the image, and wants to classify images as original or modified.

On the other side, AD's goal is to produce a processed image which has some desired characteristics, and to do that in such a way that the FA will misclassify it as original. As stated in the introduction, the AD can follow either an integrated strategy or a 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 characteristics 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. On the other hand, it is worth observing that the gain in generality may come at the expense of lower performance in terms of trace concealment.

In the following we give a brief overview of the proposed counter-forensic scheme. 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). We stress again that the proposed method is a candidate attack against *any* forensic detector relying on the analysis of the histogram. 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 according to the following steps:

1. *Retrieval*: among all histograms in S, find the one that is most similar to h_y, denote it with h_x.
2. *Mapping*: 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. *Mapping implementation*: change pixels in the image according to the histogram mapping, keeping the perceptual distortion as low as possible.

Before explaining each step in more details, some comments are in order. First, we propose to use the generalized likelihood function $\mathscr{h}$ (Gutman, 1989) both for performing histogram retrieval and as the objective function for the histogram mapping problem. This choice is justified by the fact that minimizing the $\mathscr{h}$ function between the attacked distribution and the original one is theoretically proved to be the optimum strategy for the AD against first-order detectors (Barni & Tondi, 2012). The $\mathscr{h}$ function is a similarity measure between two empirical distributions P_1 and

P_2 defined on the same alphabet, expressed as

$$\hbar(p_1, p_2) = \mathrm{D}(p_1 \| u) + \frac{n_2}{n_1}\mathrm{D}(p_2 \| u), \tag{1}$$

where the small letters p_1 and p_2 denote the normalized distributions and n_1 and n_2 are the correspondent normalization factors (that is $p_1 = P_1/n_1$ and $p_2 = P_2/n_2$). The empirical probability mass function u is obtained as linear combination of p_1 and p_2 with proportionality constants $n_1/(n_1+n_2)$ and $n_2/(n_1+n_2)$, i.e. $u = n_1/(n_1+n_2)p_1 + n_2/(n_1+n_2)q_2$. The function $\mathrm{D}(\cdot\|\cdot)$ which appears in (1) is the Kullback-Leibler divergence (Cover & Thomas, 1991). Given two probability distributions p and q defined over the same finite alphabet X, the Kullback-Leibler distance or divergence between them is defined as

$$\mathrm{D}(p \| q) = \sum_{x \in \mathsf{X}} p(x) \log \frac{p(x)}{q(x)}. \tag{2}$$

The resort to the $\hbar$ function is motivated by the fact that the $\hbar$ function is the optimum log-likelihood function of the Neyman-Pearson test when the FA knows only a training sequence drawn from the target source and not the real probability distribution (Barni & Tondi, 2012). This is the case when working with images, as in this paper.

The histogram retrieval phase also deserves a comment: according to (Barni & Tondi, 2012), if the AD had only one training image available (let us call it $\underline{t}$), the optimum way of proceeding for her would be to look for the histogram h_z which minimizes $\hbar(v_z, v_t)$, where v_z and v_t denote the normalized versions of the histograms h_z and h_t, possibly subject to a distortion constraint. In our scheme, the AD has access to a *database* of training images, thus differing from the scenario in (Barni & Tondi, 2012). Nevertheless, noting substantially changes: indeed, by extending the results in that paper, it is easy to argue that for an analyst who relies on more than one training sequences to make the decision, the optimum log-likelihood function is the minimum of the $\hbar$ function over all the training histograms in the dataset.

Histogram retrieval phase

In the scope of this work, the histogram retrieval phase can be formalized as follows: given a (processed) image $\underline{y}$ with histogram h_y find the most "similar" histogram h_x among a set S. Furthermore, the AD may choose to impose some constraints on the search, so that the desired properties induced on h_y by the processing are maintained, otherwise the counter-forensic method could remove the benefits the AD is looking for. Let us denote with $\Gamma \subset S$ the set of original histograms satisfying the constraints imposed by the AD. Then, we propose that the AD uses a constrained research over the set Γ, looking for histograms that minimize a chosen distance to h_y. As we said, we use the $\hbar$ function as similarity measure between a processed histogram and a target one. The search for the best possible target histogram can be carried out by computing the following minimization:

$$\min_{h_x \in \Gamma} \hbar(v_x, v_y), \tag{3}$$

where v_x and v_y denote the normalized versions of the histograms. We observe that the histogram resulting from (3) could not be the one that minimizes the $\hbar$ function *after* the histogram mapping phase is performed. This contrasts with the fact that the optimum attack has to guarantee the achievement of the minimum possible value for $\mathscr{h}$ among all the possible training sequences and mappings. To face this problem, 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) will actually be used. Strictly speaking, the number of K which ensures to obtain the minimum at the end of the mapping stage is not known a priori and, above all, it usually depends on the shape of the modified histogram h_y. From a theoretical point of view, for ensuring the optimality of the procedure we should consider all the histograms in the database. Although this is possible, it would be computationally too expensive, so we retain the first K histograms for the mapping phase. In practice, it turned out in our experiments that taking K equal to 10 is usually sufficient.

Notice that, the use of the $\mathscr{h}$ function allows to retrieve, in this phase, an histogram h_x that is near to h_y even from the "shape" point of view as well as from the statistical one. By looking at (1), it is straightforward to see that histograms having many bins with considerably different occurrences lead to large values for both terms in (1) and then they will not be chosen among the best K target histograms. It is also proper to stress how the $\mathscr{h}$ function does not take into account relationships between adjacent bins. The use of cross-bin histogram distances like Earth Mover's Distance (Rubner, Tomasi, & Guibas, 2000) or the Quadratic-Chi (Pele & Werman, 2010), designed to improve the search results for retrieval applications, could in principle give different results for the retrieval phase, at the cost of a higher complexity. However, for our purposes, according to the theoretical results achieved in (Barni & Tondi, 2012), the use of different target functions in the retrieval and mapping stages is (at least for images with a large size) suboptimum.

As a last consideration about histogram retrieval, we point out two important facts. The first is that the search is carried out 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.

Histogram mapping phase

Starting from a processed image $\underline{y}$ and an original histogram h_x, the AD aims at creating an attacked image $\underline{z}$ that is similar to $\underline{y}$, in the sense that keeps the desired properties, but has an histogram which is as close as possible to h_x. This problem is similar to the Optimal Transport problem (Villani, 2003), where the goal is to find a transport map which moves a given distribution into another minimizing some cost function, but it differs from it since: i) the AD does not need a perfect match between the two histograms; ii) the AD does not want to minimize the cost of the transport but rather a different functional, subject to a constraint on the maximum cost.

For the sake of generality, we assume that the image of the database from which the histogram h_x has been drawn, let us name it $\underline{x}$, has a different number of pixels than that of the processed image $\underline{y}$ (for any image of the database the number of pixels of the image is preserved by storing the histogram instead of its normalized version). Let n_x be the number of pixels of image $\underline{x}$, and let n indicate the number of pixels of the processed image $\underline{y}$, which reasonably will be the same of the attacked one $\underline{z}$. 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 vector ν_y is known, since it is computed from the processed image $\underline{y}$, while the vector ν_z has to be found. Similarly, let ν_x denote the normalized target histogram, i.e. $\nu_x(i)=h_x(i)/n_x \quad \forall i$. We introduce a *displacement matrix* $N=\{n(i\to j)\}_{i=0\dots255, j=0\dots255}$, whose (i,j)-th element tells how many elements of the histogram h_y 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 $\mathscr{h}$ function between the normalized versions of the histogram h_z and the target histogram h_x while satisfying some constraints on the distance between $\underline{z}$ and $\underline{y}$. According to (1), by using the definition of the Kullback-Leibler divergence, the $\mathscr{h}$ function between ν_z and ν_x is defined as:

$$\mathscr{h}(\nu_z,\nu_x) \;=\; \sum_{i=1}^{|I|}\nu_z(i)\log\frac{\nu_z(i)}{\nu_r(i)} \;+\; \frac{n_x}{n}\sum_{i=1}^{|I|}\nu_x(i)\log\frac{\nu_x(i)}{\nu_r(i)}, \tag{4}$$

where $\nu_r(i)=\dfrac{n}{n+n_x}\nu_z(i)+\dfrac{n_x}{n+n_x}\nu_x(i) \quad \forall i$. To simplify the notation, we define $c=\dfrac{n}{n+n_x}$ and $c_x=\dfrac{n_x}{n+n_x}$ where $c+c_x=1$, and rewrite eq. (5) as follows

$$\mathscr{h}(\nu_z,\nu_x)=\sum_{i=1}^{|I|}\nu_z(i)\log\frac{\nu_z(i)}{c\nu_z(i)+c_x\nu_x(i)} \;+\; \frac{c_x}{c}\sum_{i=1}^{|I|}\nu_x(i)\log\frac{\nu_x(i)}{c\nu_z(i)+c_x\nu_x(i)}. \tag{5}$$

As previously said, the choice of this objective function is justified in (Barni & Tondi, 2012), where it is proved that minimizing $\mathscr{h}$ is the optimal strategy for the adversary. As to the distance constraint between the images, since large pixel changes would almost surely lead to annoying artifacts, we impose a maximum value D_{max} for the absolute pixel distortion:

$$\max_i |\underline{y}(i)-\underline{z}(i)| \le D_{max}. \tag{6}$$

Then, according to (6), we put a constraint on the infinity (max) norm between $\underline{y}$ and $\underline{z}$.

We now rewrite eq. (5) and condition (6) directly as a function of the $n(i\to j)$ variables. First, we must consider that the AD cannot move from each bin of h_y more elements than those actually available. As a consequence, the following equality must be satisfied:

$$h_y(i)=n(i\to i)+\sum_{k\neq i}n(i\to k)=\sum_k n(i\to k). \tag{7}$$

Eq. (7) suggests that h_z can also be written in terms of the elements of the displacement matrix as follows:

$$h_z(i)=n(i\to i)+\sum_{k\neq i}n(k\to i)=\sum_k n(k\to i). \tag{8}$$

Substituting (8) in (5), we can rewrite the objective function in terms of the $n(i\to j)$ variables:

$$\min_{n(i\to j)} \sum_{i=1}^{|\mathcal{I}|} \frac{(\sum_k n(k\to i))}{n}\cdot\log\frac{(\sum_k n(k\to i))/n}{c(\sum_k n(k\to i))/n + c_x \nu_x(i)} \;+\; \frac{c_x}{c}\sum_{i=1}^{|\mathcal{I}|} \nu_x(i)\cdot\log\frac{\nu_x(i)}{c(\sum_k n(k\to i))/n + c_x \nu_x(i)}. \tag{9}$$

As a second step, we express constraint (6) as a function of $n(i\to j)$ yielding

$$n(i\to j)=0,\ \forall(i,j)\in\mathcal{I}\times\mathcal{I}:|i-j|>D_{max}. \tag{10}$$

We can therefore rephrase the optimization problem as follows:

$$\min_{n(i\to j)} \sum_{i=1}^{|\mathcal{I}|} \frac{(\sum_k n(k\to i))}{n}\cdot\log\frac{(\sum_k n(k\to i))/n}{c(\sum_k n(k\to i))/n + c_x \nu_x(i)} + \frac{c_x}{c}\sum_{i=1}^{|\mathcal{I}|} \nu_x(i)\cdot\log\frac{\nu_x(i)}{c(\sum_k n(k\to i))/n + c_x \nu_x(i)}, \tag{11}$$

subject to

$$\begin{cases} \sum_j n(i\to j)=h_y(i)\ \forall i \\ n(i\to j)=0,\ \forall(i,j)\in\mathcal{I}\times\mathcal{I}:|i-j|>D_{max} \\ n(i\to j)\geq 0 \quad \forall i,j \\ n(i\to j)\in\mathbb{N}. \end{cases} \tag{12}$$

The above optimization problem belongs to the MINLP (Mixed integer nonlinear problems) class (Bussieck & Pruessner, 2003). By looking at the objective function, we can argue that it is a convex function in the $n(i\to j)$ variables. For sake of brevity, the proof of such a claim is omitted being similar to the one given in (Barni & Tondi, 2013) for the convexity of the $\mathcal{D}$ function in the same variables. Since the constraint functions defining the feasible set are also convex in the $n(i\to j)$ variables and upper bounded, the problem is actually a *convex* MINLP (Bussieck & Pruessner, 2003), for which a global optimum solution exists. For convex MINLPs there are several efficient solvers yielding the optimum solution (Bussieck & Vigerske, 2011). Among the most common algorithms for solving convex MINLPs, a remarkable candidate is the branch and bound method, according to which we solve the NLP (nonlinear programming) relaxation of the problem obtained by removing the constraint that the $n(i\to j)$ variables must assume integer values (Bonami, Kilinc, Linderoth, & others, 2009). In our application, we used the BONMIN (Basic Open-source Nonlinear Mixed Integer programming) solver in the BB mode (Bussieck & Vigerske, 2011) which implements the NLP-based branch and bound algorithm. By default, it resorts to the software package IPOPT (Wächter & Biegler, 2006) to solve the NLP relaxation.

As to the computational complexity, we notice that the number of optimization variables is quadratic in $|\mathcal{I}|$. This means that the complexity of problem (11)-(12) does not depend on the size of the image, but only on the bit-depth of the images (so we will usually have $|\mathcal{I}|=256$). We underline that the actual number of optimization variables is even lower than $|\mathcal{I}|^2$. In fact, some further considerations can be done regarding formulation (11)-(12) First of all, we can remove the second constraint in (12) by properly restricting the sums in the objective function and in the first constraint. For notational simplicity let us define $\forall i\in\mathcal{I}$ the set $\mathcal{A}(i)=\{k\in\mathcal{I}:|k-i|\leq D_{max}\}$. Accordingly, we can rewrite the optimization problem in the following equivalent form:

$$\min_{n(i\to j)} \sum_{i=1}^{|\mathsf{I}|} \frac{\left(\sum_{k\in \mathsf{A}(i)} n(k\to i)\right)}{n} \cdot \log \frac{\dfrac{\left(\sum_{k\in \mathsf{A}(i)} n(k\to i)\right)}{n}}{\dfrac{c\left(\sum_{k\in \mathsf{A}(i)} n(k\to i)\right)}{n} + c_x v_x(i)}$$

$$+\frac{c_x}{c}\sum_{i=1}^{|\mathsf{I}|} v_x(i)\cdot\log\frac{v_x(i)}{\dfrac{c\left(\sum_{k\in \mathsf{A}(i)} n(k\to i)\right)}{n} + c_x v_x(i)}. \qquad (13)$$

subject to

$$\begin{cases} \sum_{j\in \mathsf{A}(i)} n(i\to j) = h_y(i)\ \forall i \\ n(i\to j)\geq 0 \quad \forall i,j \\ n(i\to j)\in \mathsf{N} \end{cases} \qquad (14)$$

obtaining a slightly simplified set of constraints. Furthermore, by looking at the first constraint in (12) (and (14)), we notice that all the optimization variables $n(i\to j)$ describing displacements from empty bins to any other bin will have a zero value, that is $h_y(i)=0$ implies $n(i\to j)=0$ for all j. Let E be the set of the empty bins, with $\mathsf{E}\subset \mathsf{I}$. It is easy to argue that the actual complexity of the problem is $2D_{max}\cdot(|\mathsf{I}|-|\mathsf{E}|)$ which is often much less than $|\mathsf{I}|^2$.

By referring to the problem rewritten as in (13)-(14), the optimization is very fast and, on average, the time taken by the solver to find the optimum mapping is less than one second (tests have been performed on a computer equipped with a Intel i7 CPU, 8GB RAM, under Windows 7 operating system). Two final observations about the above problem are in order. Firstly, 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. Secondly, we notice that in general there are many cases in which 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 bring h_y close to h_x respecting the constraint about maximum distortion, so the problem is infeasible, and the resulting $\mathfrak{k}(v_z,v_x)$ is very high. However, in our particular case, a target histogram h_x which is so different from h_y should already have been discarded during the retrieval phase. In fact, if the dataset is large enough it is very unlikely that a better target histogram and, therefore, a feasible mapping is not found.

Pixel remapping phase

After the attacked 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 (Z. Wang, Bovik, Sheikh, & Simoncelli, 2004). 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 (Z. Wang et al., 2004). 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 of $\underline{y}$;
3. For each couple $(i, j)$:
    a. find admissible pixels location having value $i$;
    b. scan them selecting the first $n(i \to j)$ with highest values in the map;
    c. substitute them with $j$;
    d. remove selected pixels from the admissible ones;
    e. if no more pixels of value $i$ must be remapped, compute the SSIM map
       between the current image and $\underline{y}$;
```

One first comment regards 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). A good tradeoff is obtained by computing the map $|\mathbf{I}|$ 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 yet. 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 the next section, 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 K histograms yielding the lowest $\mathcal{h}$ distance in the dataset. 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 the processes processed and attacked images is satisfactory.

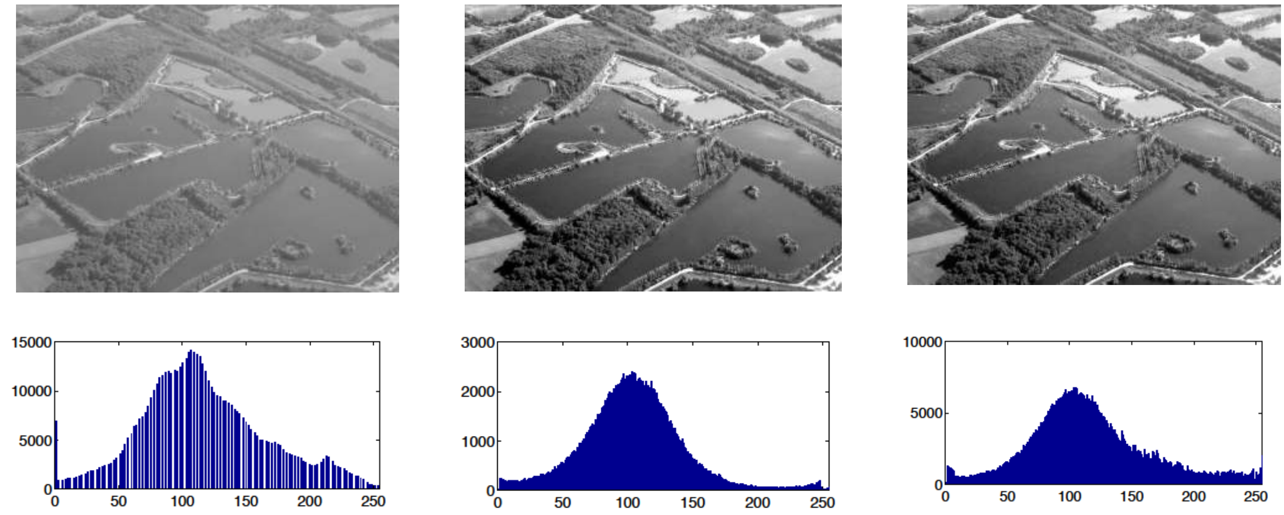


Figure 2: Top row, from left to right: an original image, its processed (contrast stretched) version, and the image resulting from the proposed CF technique. Bottom row: the histogram of the processed image (left), the one retrieved from the database (center), and the one obtained applying the proposed method (right).

Experimental results and practical applications

In this section we evaluate the performance of the proposed counter forensic technique in two different applications. It is important to notice that evaluating a universal counter forensic method means

The first application deals with *enhancement detection*, where the goal of the FA is to discover whether an image has been globally enhanced with some processing operator or not. Therefore, we apply a histogram-based enhancement to a given image and then we use the proposed technique to remove traces from the processed histogram. The second application deals with *splicing detection*, where the FA wants to understand whether a given region of an image has been pasted from another picture or not. Creating a splicing usually requires to adapt the histogram of the cut region to that of the host picture, thus leaving traces in some areas of the forgery. In this case, therefore, we use the proposed technique to remove the processing traces only from the spliced content, that will usually be much smaller than the whole image.

In order to test the effectiveness of the proposed scheme, we implemented a state-of-the-art algorithm for the detection of histogram based image enhancement proposed by Stamm and Liu (Stamm & Liu, 2008). This tool exploits the fact that most histogram-enhancement techniques leave a characteristic fingerprint in the image histogram, namely the peak-and-gaps artifact. This effect is easily exposed in the frequency domain, where the mentioned behavior results in an anomalous amount of high-frequency components. Therefore, by investigating the Fourier transform of the image histogram, the authors devised a very reliable detector. Since this detector considers only the histogram of the image, the proposed universal counter-forensic scheme should be able to fool it. To check if this is the case, we used Stamm's algorithm for distinguishing processed and attacked images from untouched ones. Performances are measured in terms of the Area Under Curve (AUC) of the detector before and after the attack, while the quality of the attacked images is evaluated using PSNR and Structural Similarity (SSIM).

Attacking enhancement detection

In this first application, the enhancement and the attack are performed over the whole image. To generate the enhanced images, we employed two different techniques: one based on γ -correction and one based on histogram stretching. γ-correction enhancement is very simple, being fully described by the following equation:

$$\underline{y}(i) = 255 \times \left(\frac{\underline{x}(i)}{255} \right)^{\gamma} \tag{15}$$

where $\underline{y}$ denotes the enhanced image and $\underline{x}$ denotes the original one. Since values of γ very near to 1 would not result in a sensible modification, in our experiments γ is always randomly chosen from the set $[0.5;0.8] \cup [1.2;2]$.

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:

$$\underline{y}(i) = 255 \times \frac{\underline{x}(i) - l_{min}}{l_{max} - l_{min}}. \tag{16}$$

Comparing the left-most and center images in Figure 2, the effect of histogram stretching in improving image quality is evident.

Since the AD wants to preserve the benefits obtained 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 (Michelson, 1927), that for a given image histogram h is

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

where h_{max} is the largest 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:

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

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.

We conducted our experiments by using images from the UCID dataset (Schaefer, 2010). We also used another independent dataset, MIRFLICKR (Huiskes & Lew, 2008), composed by 25.000 images of the same size, to prepare the database of untouched histograms. The use of a database containing images with the same number of pixels is motivated by the fact that this number plays a role in $\mathscr{h}$, and consequently affects the histogram retrieval phase. If all the images in the database share the same size, the function $\mathscr{h}$ will only account for their shape, thus favouring histograms that are easier to map in the subsequent phase. Throughout the experiments, all color images are converted to grayscale. The only parameters the attacker has to choose are the number of candidates for which the optimization problem is solved (we used $K = 10$) and the maximum per-pixel distortion; of course, allowing a higher distortion will yield a more precise mapping of the attacked histogram into the one selected from the database but will also result in a lower quality of the attacked image. 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 performed, separately, contrast enhancement and histogram stretching over all pictures in the UCID dataset and run Stamm's detector on the resulting images; then, we applied the proposed counter-forensic scheme on each processed image, for various D_{max}, and run again the detector. Figures Figure 3 and Figure 4 show, respectively, the results obtained by hiding traces of contrast-enhancement and histogram stretching operations. In both figures, ROC curves obtained for different values of maximum per-pixel distortion are plotted: 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. Compared to our previous work (Barni et al., 2012), we can observe a little improvement in deceiving Stamm's detector. Nevertheless, we stress that Stamm's detector has been considered as a specific case study: the sole fact that the proposed method successfully deceives a specific detector does not prove its universality.

Since what we are proposing is a *universal* counter forensic method, it is important to investigate how much the histogram of the attacked image deviates from the one coming from the database. In order to better highlight the fidelity of the remapped histogram h_z to h_x, we calculated for each experiment the χ^2 distance between their normalized versions, defined as:

$$\chi^2(\nu_x, \nu_z) = \frac{1}{2}\sum_{i=0}^{255} \frac{(\nu_x(i) - \nu_z(i))^2}{\nu_x(i) + \nu_z(i)},$$

and reported its average value on in the right-most column of Tables 1 and 2, also comparing the obtained values to those achieved by our previous method (Barni et al., 2012).

The choice of the χ^2 distance allows us to show that the obtained histogram is similar to the untouched one also according to measures that were not directly considered in our scheme. We see that, on average, the χ^2 distance between the histograms takes values in the order of 10^{-2} for the proposed method, which can be considered definitely small. This fact strongly supports the universality claim, because devising an histogram-based forensic detector capable of discriminating between such similar histograms would be extremely difficult. Finally, Tables 1 and 2 show that our new formulation results in remapped histograms that are, on average, slightly closer to the target ones compared to those obtained with the method in (Barni et al., 2012). Notice that, since we are proposing a universal approach, even a small improvement in the similarity between the attacked and target histograms is of noticeable importance.

Of course, the above performance measures would be meaningless if we do not investigate the fidelity of the attacked images to the processed ones: also this information is reported in Tables 1 and 2 for contrast enhanced and histogram stretched images respectively. 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 counter-forensic attack does not produce annoying artifacts, nor it removes the benefits introduced by the processing carried by the AD.

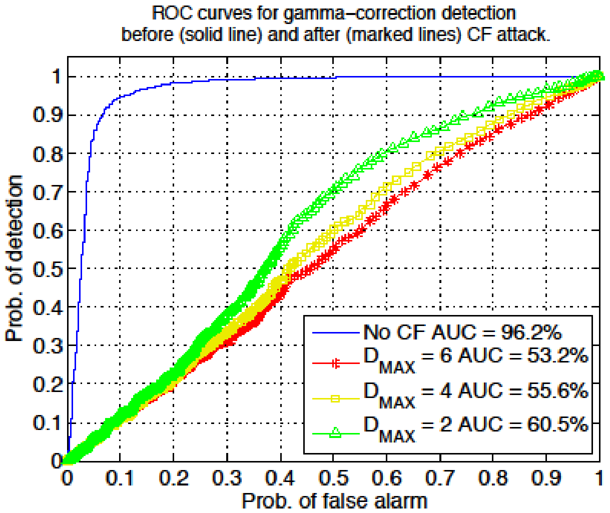


Figure 3: ROC curves for Contrast Enhancement Detector running on gamma-corrected images (solid line) and on attacked images (marked lines).

D_{max}	PSNR (db)	SSIM	AUC	Previous χ^2	Proposed χ^2
2	44.8	0.993	0.605	0.113	0.092
4	39.2	0.979	0.556	0.080	0.058
6	36.2	0.962	0.532	0.061	0.040

Table 1: mean values for PSNR and SSIM between processed and attacked images, along with the Area Under Curve obtained by the forensic detector. The last two columns show the average X^2 distance between the remapped histogram and the one coming from the database obtained, respectively, with our previous method in (Barni et al., 2012) and the proposed one.

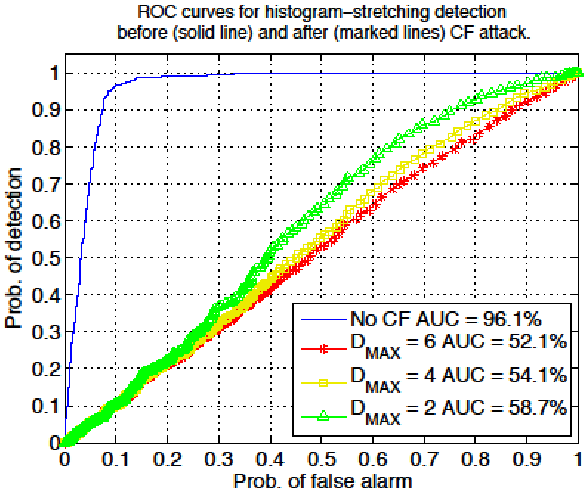


Figure 4: ROC curves for Contrast Enhancement Detector running on histogram-stretched images (solid line) and on attacked ones (marked lines).

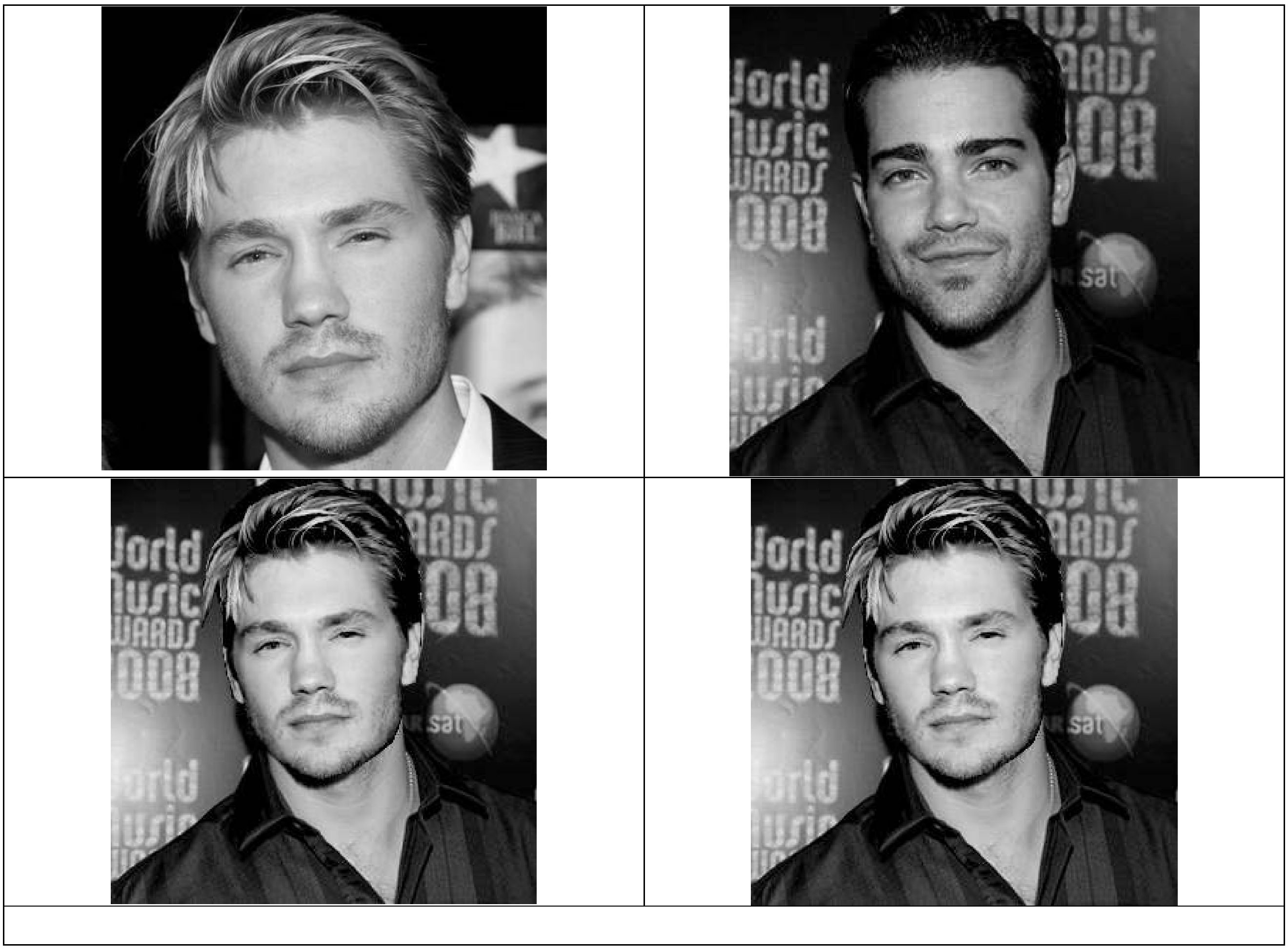
D_{max}	PSNR(db)	SSIM	AUC	Previous χ^2	Proposed χ^2
2	44.8	0.994	0.587	0.060	0.060
4	39.2	0.981	0.541	0.033	0.032
6	36.1	0.964	0.521	0.021	0.019

Table 2: mean values for PSNR and SSIM between processed and remapped images, along with the Area Under Curve obtained by the forensic detector. T The last two columns show the average X^2 distance between the remapped histogram and the one coming from the database obtained, respectively, with our previous method in (Barni et al., 2012) and the proposed one.

Attacking splicing detection

Splicing detection is one of the most important tasks in image forensics. A splicing is the composition of two (or more) images into one, and is usually accomplished by cutting a portion of a *source* image and pasting it into a *host* image, thus creating a forgery. During this process, it is usually necessary to adapt the brightness and contrast of the inserted patch of pixels so that they match with the destination content. In general, the presence of this patch of "enhanced pixels" may not be revealed by observing the overall histogram of the forged image; on the other hand, if the forensic analyst has a suspect about a specific region of the image, he may investigate the histogram of that sub-part to understand if it has been altered. For this reason, we show how the proposed counter-forensic scheme can be used to create realistic forgeries that do not show artifacts in the histogram of the spliced region. Since creating this kind of forgeries is a time-consuming task, we limit this experiment to a set of 7 hand-made splicings, and we report the obtained results along with a specific example.

The experiments have been carried out as follows: given a source and a host image, we selected a part of the source image to be pasted into the host image. After being cut, and possibly resized and/or rotated, the selected patch has been adjusted by using histogram stretching techniques, so that its contrast/brightness fits the destination picture. Then, we created two forgeries: one by simply pasting the processed patch of pixels, the other by first applying the proposed counter-forensic scheme and then pasting the pixels. Finally, we played the role of the FA: we analyzed the histogram of the pixels belonging to the pasted region using the Stamm and Liu algorithm (2008), and compared the obtained results for the two mentioned forgeries. By comparing the output of the detector in the two cases we could evaluate the effectiveness of the proposed counter-forensic scheme. Furthermore, we also computed the structural similarity between the processed patch of pixels and its counter-attacked version. Figure 5 shows an example of the experiment, where the SSIM between the patch of pixels before and after counter-forensic is 0.976, but the output of the forensic detector drops from 1.191 to 0.604. Since the threshold corresponding to the cut-off point for the detector lays around 0.78, the algorithm would classify the image as non tampered.



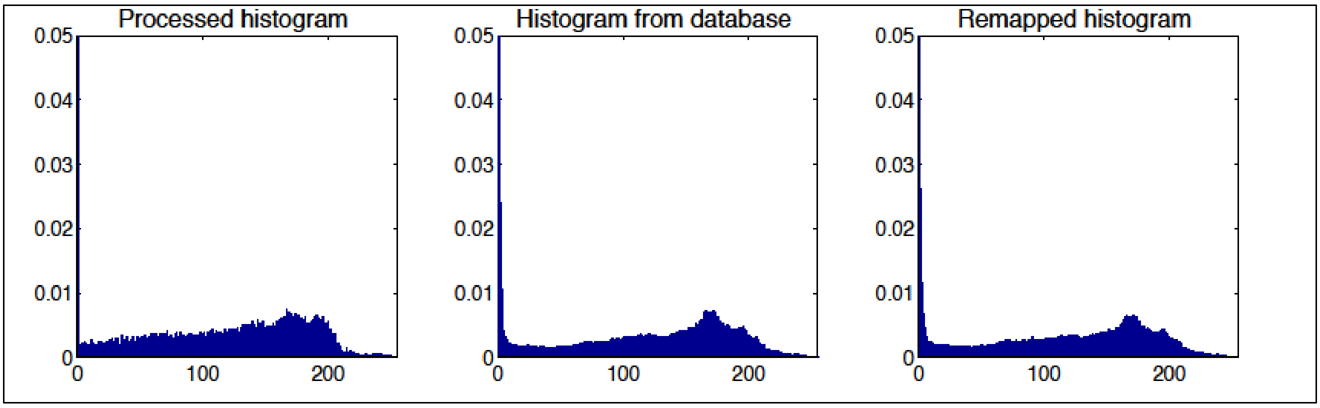


Figure 5: typical procedure for the creation of a splicing: given the source image (top left) and the host image (top right), part of the source is cut and adapted (resized and histogram-stretched) to be pasted into the host, thus creating the fake (middle left). By applying the proposed counter-forensic scheme, a perceptually identical fake is obtained (middle right) where the histogram of the pasted region has been mapped to a non-suspicious one coming from a database of untouched images, as shown at the bottom.

We report in Table 3 the detector output before and after counter-forensic, along with the SSIM values between the two suspect regions, for other hand-made forgeries available at the web address http://clem.dii.unisi.it/vipp/index.php/download/imagerepository. Also in this application, we clearly see the benefits of the proposed attack: the detector output drops significantly after the application of the proposed scheme, while the perceptual fidelity is fully preserved.

File	Before CF	After CF	SSIM
leaders	1.19	0.60	0.97
river	1.86	0.58	0.98
singer	1.06	0.24	0.98
smiling	0.61	0.21	0.97
missmondo	1.08	0.13	0.97
parachute	0.89	0.40	0.97
car	0.95	0.48	0.99

Table 3: results obtained on 7 hand-made splicing. Notice the significant drop of the detector output before and after application of the counter-forensic scheme. In the right-most column, the SSIM value between the processed and attacked version of the image is reported.

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 CF attacks: after an image has been processed, 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 remap the 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. Leveraging on the theoretical results proved in (Barni & Tondi, 2012), the h function is used for retrieving and mapping the histograms. Being a not-targeted procedure, the proposed tecnique can be used to counter any histogram-based detector. Experimental results showed the effectiveness of the proposed approach both for countering enhancement- and splicing- detection; furthermore, an increase in performance is obtained compared to our method in (Barni et al., 2012). Future work will focus on investigating how the proposed method can be extended to color images and how it can be used to remove traces left in the histograms of DCT coefficients, thus widening the applicability of the approach to a broader set of forensics tasks.

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.

References

Barni, M., Fontani, M., & Tondi, B. (2012). A universal technique to hide traces of histogram-based image manipulations. In *Proc. of MM&Sec 2012, 14th ACM workshop on Multimedia & Security* (pp. 97–104). New York, NY, USA: ACM. doi:10.1145/2361407.2361424

Barni, M., & Tondi, B. (2012). Optimum forensic and counter-forensic strategies for source identification with training data. In *Information Forensics and Security (WIFS), 2012 IEEE International Workshop on* (pp. 199 –204). doi:10.1109/WIFS.2012.6412649

Barni, M., & Tondi, B. (2013). The Source Identification Game: an Information-Theoretic Perspective. *Information Forensics and Security, IEEE Transactions on*, *PP*(99), 1. doi:10.1109/TIFS.2012.2237397

Bonami, P., Kilinc, M., Linderoth, J., & others. (2009). *Algorithms and software for convex mixed integer nonlinear programs*. Computer Sciences Department, University of Wisconsin-Madison.

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

Bussieck, M. R., & Vigerske, S. (2011). MINLP Solver Software.

Cao, G., Zhao, Y., Ni, R., & Tian, H. (2010). Anti-forensics of contrast enhancement in digital images. In

Proceedings of MM&Sec 2010, 12th ACM workshop on Multimedia and security (MM&Sec '10).

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York, NY, USA: Wiley-Interscience.

Gutman, M. (1989). Asymptotically Optimal Classification for Multiple Tests with Empirically Observed Statistics. *IEEE Transactions on Information Theory*, *35*(2), 401–408.

Huiskes, M. J., & Lew, M. S. (2008). The MIR Flickr Retrieval Evaluation. In *Proc. of MIR '08, ACM International Conference on Multimedia Information Retrieval*. Vancouver, Canada: ACM.

Kirchner, M., & Böhme, R. (2007). Tamper Hiding: Defeating Image Forensics. In *Proc of IH 2007, Int. Conference on Information Hiding* (pp. 326–341).

Michelson, A. A. (1927). *Studies in optics*. University of Chicago Press.

Pele, O., & Werman, M. (2010). The Quadratic-Chi Histogram Distance Family. In *Proc. of ECCV 2010, European Conference on Computer Vision*.

Redi, J., Taktak, W., & Dugelay, J.-L. (2011). Digital image forensics: a booklet for beginners. *Multimedia Tools and Applications*, *51*(1), 133–162.

Rubner, Y., Tomasi, C., & Guibas, L. J. (2000). The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, *40*(2), 99–121.

Schaefer, G. (2010). An uncompressed benchmark image dataset for colour imaging. In *Proc. of ICIP 2010, IEEE Int. Conference on Image Processing* (pp. 3537–3540). doi:10.1109/ICIP.2010.5651245

Stamm, M. C., Lin, S., & Liu, K. J. R. (2012). FORENSICS VS. ANTI-FORENSICS: A DECISION AND GAME THEORETIC FRAMEWORK. In *Proc. of ICASSP 2012, IEEE Int. Conference on Acoustics, Speech, and Signal Processing*.

Stamm, M. C., & Liu, K. J. R. (2008). Blind forensics of contrast enhancement in digital images. In *Proc.*

of ICIP 2008, IEEE Int. Conference on Image Processing (pp. 3112–3115).

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

Stamm, M. C., & Liu, K. J. R. (2011). Anti-forensics for frame deletion/addition in MPEG video. In *Proc. of ICASSP 2011, IEEE Int. Conference on Acoustics, Speech and Signal Processing* (pp. 1876 –1879). doi:10.1109/ICASSP.2011.5946872

Stamm, M. C., Tjoa, S. K., Lin, W. S., & Liu, K. J. R. (2010). Undetectable image tampering through JPEG compression anti-forensics. In *Proc. of ICIP 2010, IEEE Int. Conference on Image Processing* (pp. 2109 –2112). doi:10.1109/ICIP.2010.5652553

Valenzise, G., Nobile, V., Tagliasacchi, M., & Tubaro, S. (2011). Countering JPEG anti-forensics. In *Proc. of ICIP 2011, IEEE Int. Conference on Image Processing* (pp. 1949–1952).

Villani, C. (2003). *Topics in optimal transportation*. (A. M. Society, Ed.). American Mathematical Society.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*(1), 25–57. doi:10.1007/s10107-004-0559-y

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

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. doi:10.1109/TIP.2003.819861