THE ROLE OF PERMUTATION CODING IN MINIMUM-DISTORTION PERFECT COUNTERFORENSICS

Félix Balado

School of Computer Science and Informatics
University College Dublin, Ireland

ABSTRACT

This paper exploits the connection between minimum-distortion perfect counterforensics and maximum-rate perfect steganography in order to provide the optimum solution to the first of these problems, in the case in which the forensic detector solely uses first-order statistics. The solution relies on Slepian's variant I permutation codes, which had previously been shown to implement maximum-rate perfect steganography when the host is memoryless (equivalently, when the steganographic detector only uses first-order statistics). Additionally, we demonstrate a blind counterforensic strategy made possible by permutation decoding, which may also find application in image processing.

Index Terms— Counterforensics, steganography, permutation coding, histogram-based forensics, histogram specification

1. INTRODUCTION

The field of counterforensics deals with techniques aimed at misleading digital forensic detection tests, whose goal is determining the authenticity of digital assets. Many algorithms have been proposed after the counterforensics concept was first formulated by Kirchner and Böhme [1]. The reader can find a good survey of relevant counterforensic algorithms for image forensics in [2]. However existing counterforensic methods have generally been of a heuristic nature, and thus neither blind (that is, they tend to be targeted to particular forensic detectors) nor minimum-distortion perfect (that is, they do not produce a forged signal which both evades detection and is as similar as possible to a target forgery). The exception are two algorithms recently put forward for the case in which the forensic detector only relies on first-order statistics: 1) Barni et al. [3] have given a blind counterforensic algorithm, which although not undetectable it conforms to a target fidelity constraint; and 2) Comesaña and Pérez-González [4] have essentially solved the problem of undetectable counterforensics with maximum fidelity, assuming full knowledge of the forensic detector (nonblind counterforensics).

In spite of these important advances, we firmly believe that the counterforensics problem warrants further examination. This is so even in the case of universal first-order detection (i.e. histogram-based), which is largely solved and on which we will focus here. A powerful reason for reexamining the problem is the observation by Böhme and Kirchner that counterforensics has strong links with data hiding (steganography and watermarking), and thus should not be studied in isolation but in connection with this field [2]. This is the main motivation behind this paper, in which we will follow a very specific link: the one between *minimum-distortion perfect counterforensics* and *maximum-rate perfect steganography*. We have chosen this nomenclature in order to unambiguously emphasise the parallels between the two settings. For the avoidance of doubt, by a minimum-distortion perfect counterforensic method we mean an algorithm that modifies a forgery in such a way that the modified signal is as close as possible to the forgery according to some distance, while simultaneously being accepted as legitimate by a forensic investigator performing optimum detection. On the other hand, by a maximum-rate perfect steganographic method we mean an algorithm that modifies a host to embed information with maximum embedding rate, and such that the modified signal is accepted as not suspicious by a warden performing optimum detection.

Both problems feature an agent (a forger and a steganographer, respectively) who modifies a signal with a view to avoiding detection of this modification by a second agent (a forensic investigator and a warden, respectively). The key link between *perfect* counterforensics and *perfect* steganography is that the modified signal is drawn in both cases from a formally identical pool: the set of all signals with a statistical structure such that the detector will not declare them illegitimate (counterforensics) or suspicious (steganography). The main difference lies in what the forger and the steganographer actually do with this set of perfect signals. The forger wishes to find a signal in the set which lies as close as possible to his forgery (*minimum-distortion* counterforensics); importantly, he needs to consider the whole ensemble of signals in order to always find an optimum one. Instead, the steganographer wishes to attach unique labels to all signals in the set, thus maximising the number of messages that can be conveyed (*maximum-rate* steganography), and also must be able to produce a signal in the set given its label, and vice versa[1].

In a recent paper [5] we showed that Slepian's variant I permutation codes [6] implement maximum-rate perfect steganography if the steganographic detector only examines first-order statistics (or if the host is memoryless). In the light of the discussion above, it follows that Slepian's variant I permutation codes must also be central to implementing minimum-distortion perfect counterforensics when the forensic detector only examines first-order statistics. In this paper we will describe, analyse and test this use of permutation coding.

Notation. Boldface lowercase Roman letters are column vectors. The special symbol $\mathbf{1}$ is an all-ones vector. Capital Greek letters denote matrices; the entry at row i and column j of matrix Π is $(\Pi)_{i,j}$. $(\cdot)^t$ denotes a vector or matrix transpose. The 2-norm of a vector $\mathbf{u}$ is $\|\mathbf{u}\| = \sqrt{\mathbf{u}^t\mathbf{u}}$. Calligraphic letters are sets. The indicator function is defined as $\mathbb{1}_{\{A\}} = 1$ if event A is true, and zero otherwise. Random variables are represented by capital letters.

A signal is denoted by a vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]^t \in$

This work has been financially supported by Science Foundation Ireland under grant 09/RFP/CMS2212.

[1] A distortion constraint must also be imposed if not all signals in the set are semantically equivalent to the host, but this does not alter our discussion.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

$\mathcal{V}^n$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_q\} \subset \mathbb{Z}$. We assume that $\mathbf{v} = [v_1, v_2, \ldots, v_q]^t$ gives the elements of $\mathcal{V}$ in increasing order, that is, $v_1 < v_2 < \cdots < v_q$. The histogram of $\mathbf{x}$ is a vector $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \ldots, h_q(\mathbf{x})]^t$ such that $h_k(\mathbf{x}) = \sum_{i=1}^n \mathbb{1}_{\{v_k = x_i\}} \geq 0$, and then $\mathbf{1}^t\mathbf{h}(\mathbf{x}) = n$. Let $\mathcal{S}_n$ be the symmetric group on $\{1, 2, \ldots, n\}$, which contains all permutations of the integers between 1 and n. We denote a permutation $\boldsymbol{\sigma} \in \mathcal{S}_n$ by means of a vector $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_n]^t$ where $\sigma_i \in \{1, 2, \ldots, n\}$ and $\sigma_i \neq \sigma_j$ for all $i \neq j$. This vector can be used in turn to define a permutation matrix $\Pi_{\boldsymbol{\sigma}}$ with entries $(\Pi_{\boldsymbol{\sigma}})_{i,j} = \mathbb{1}_{\{\sigma_i = j\}}$. Whenever we just write Π (without a $\boldsymbol{\sigma}$ subindex) we refer to a generic $n \times n$ permutation matrix. The rearrangement of $\mathbf{x}$ using $\boldsymbol{\sigma}$ is the vector $\mathbf{y} = \Pi_{\boldsymbol{\sigma}}\,\mathbf{x}$, for which $y_i = x_{\sigma_i}$ for $i = 1, 2, \ldots, n$. We will follow the convention that a *rearrangement* of $\mathbf{x}$ is a unique ordering of its elements —in general, different permutations can lead to the same rearrangement of a vector. A special case is the rearrangement of $\mathbf{x}$ in nondecreasing order, which we denote by $\overrightarrow{\mathbf{x}}$.

2. MINIMUM-DISTORTION PERFECT COUNTERFORENSICS

We assume a general post-processing counterforensic scheme [1]. The forger (attacker) post-processes a *forgery*, denoted by $\mathbf{z} \in \mathcal{V}^n$, to obtain a *post-processed forgery*, denoted by $\mathbf{y} \in \mathcal{V}^n$. The forensic detector is some function $\phi : \mathcal{V}^n \to \{0, 1\}$ such that $\phi(\mathbf{y}) = 1$ when $\mathbf{y}$ is classed as legitimate (authentic) and $\phi(\mathbf{y}) = 0$ when $\mathbf{y}$ is classed as illegitimate (forged). We assume that the forger knows that the forensic detector solely relies on the histogram of the signal to be tested, but no further knowledge of the detector is assumed in this section. The goal of the forger is to ensure that $\phi(\mathbf{y}) = 1$ while maximising the similarity between $\mathbf{z}$ and $\mathbf{y}$ according to some measure. We call a *decoy*, denoted by $\mathbf{x} \in \mathcal{V}^n$, any legitimate signal which the forger exploits to produce $\mathbf{y}$. Apart from requiring that $\phi(\mathbf{x}) = 1$, the only other strict condition that we are imposing on $\mathbf{x}$ at this point is that its length be the same as that of $\mathbf{z}$ (that is, n). The decoy choice will be discussed in Section 3, since it is not relevant to draw the parallels mentioned in Section 1.

The central observation is the following one: any sequence $\mathbf{y}$ with the same histogram as a given decoy $\mathbf{x}$ must be a rearrangement of it, i.e., $\mathbf{h}(\mathbf{y}) = \mathbf{h}(\mathbf{x})$ iff $\mathbf{y} = \Pi\mathbf{x}$. Since the detector only examines the histogram, then the forger knows that $\Pi\mathbf{x}$ will always be declared to be legitimate by the detector, that is, if $\phi(\mathbf{x}) = 1$ then $\phi(\Pi\mathbf{x}) = 1$. Thus, under the knowledge of the legitimacy of $\mathbf{x}$, the ensemble of signals among which the forger can choose a perfect post-processed forgery is the same as the set of Slepian's Variant I permutation codewords with base codeword $\mathbf{x}$ [6]. If one replaces "decoy" by "host" and "post-processed forgery" by "watermarked signal", the parallel with perfect steganography is evident (see [5]).

The fundamental difference with steganography is that in counterforensics the forger wants to find a rearrangement $\mathbf{y}$ of $\mathbf{x}$ that is as close as possible to $\mathbf{z}$. If the function $\delta : \mathcal{V}^n \times \mathcal{V}^n \to \mathbb{R}^{\geq 0}$ measures the distance between two signals, this entails finding a permutation $\boldsymbol{\sigma}_*$ that solves the following combinatorial optimisation:

$$\boldsymbol{\sigma}_* = \arg\min_{\boldsymbol{\sigma} \in \mathcal{S}_n} \delta(\mathbf{z}, \Pi_{\boldsymbol{\sigma}}\mathbf{x}). \tag{1}$$

Equivalently, the forger looks for a codeword which is as close as possible to $\mathbf{z}$ in the codebook formed by all possible signals with histogram $\mathbf{h}(\mathbf{x})$. This can also be seen as quantizing $\mathbf{z}$ using the codebook of all rearrangements of $\mathbf{x}$. Using (1), we may define the quantization of $\mathbf{z}$ using that codebook as $Q_{\mathbf{x}}(\mathbf{z}) \triangleq \Pi_{\boldsymbol{\sigma}_*}\mathbf{x}$, and then write $\mathbf{y} = Q_{\mathbf{x}}(\mathbf{z})$. This centroid is a minimum-distortion perfect post-processed forgery given the decoy $\mathbf{x}$. The setup just described, formally identical to source coding using permutation codes [7], is summarised in Figure 1.

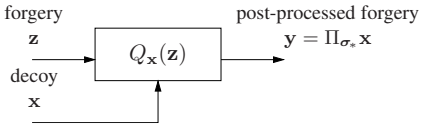

Fig. 1. Minimum-distortion perfect counterforensics using permutation decoding

In the remainder we will focus on the use of the Euclidean distance, $\delta(\mathbf{z}, \mathbf{y}) = \|\mathbf{z} - \mathbf{y}\|$. Minimising the Euclidean distance maximises the peak signal-to-noise ratio $\text{PSNR}(\mathbf{z}, \mathbf{y}) = n255^2/\|\mathbf{z} - \mathbf{y}\|^2$ (when $\mathcal{V} = \{0, 1, \cdots, 255\}$). The Euclidean distance is also convenient when the forensic test does not act upon $\mathbf{y}$, but upon an energy-preserving linear transformation of it, $\Upsilon\mathbf{y}$ (where Υ is an orthogonal matrix): in this case $\|\Upsilon(\mathbf{z} - \mathbf{y})\| = \|\mathbf{z} - \mathbf{y}\|$.

The solution to the problem of finding a rearrangement $\mathbf{y}$ of $\mathbf{x}$ closest to $\mathbf{z}$ in the Euclidean distance sense was obtained by Slepian in his original paper on permutation coding [6], motivated by maximum-likelihood decoding under a Gaussian channel. The solution is simple, perhaps surprisingly so when we consider that there are $\binom{n}{\mathbf{h}(\mathbf{x})} = n!/(h_1(\mathbf{x})! \cdots h_q(\mathbf{x})!)$ rearrangements of $\mathbf{x}$ —an amount that grows exponentially with n— and when we take into account the fact that, in general, the centroids of a permutation quantizer are not regularly laid out—unlike, for instance, those of lattice quantizers. We will give next a derivation of the solution more compact than the one given in [6]. First see that, since $\|\mathbf{z} - \Pi\mathbf{x}\|^2 = \|\mathbf{z}\|^2 + \|\mathbf{x}\|^2 - 2\,\mathbf{z}^t\Pi\mathbf{x}$ because all rearrangements of $\mathbf{x}$ have constant norm, then solving (1) is equivalent to maximising the bilinear form $\mathbf{z}^t\Pi_{\boldsymbol{\sigma}}\mathbf{x}$ over $\boldsymbol{\sigma} \in \mathcal{S}_n$. We then invoke the rearrangement inequality $\mathbf{z}^t\mathbf{x} \leq \overrightarrow{\mathbf{z}}^t\overrightarrow{\mathbf{x}}$ [8, chapter 10], which holds for any two n-vectors $\mathbf{z}$ and $\mathbf{x}$. This inequality also implies that

$$\mathbf{z}^t\Pi\mathbf{x} \leq \overrightarrow{\mathbf{z}}^t\overrightarrow{\mathbf{x}}. \tag{2}$$

One can now write $\overrightarrow{\mathbf{z}}^t\overrightarrow{\mathbf{x}} = \mathbf{z}^t\Pi_{\boldsymbol{\sigma}_z}^t\Pi_{\boldsymbol{\sigma}_x}\mathbf{x}$, where $\boldsymbol{\sigma}_z, \boldsymbol{\sigma}_x \in \mathcal{S}_n$ are any two permutations that sort $\mathbf{z}$ and $\mathbf{x}$, respectively, in nondecreasing order. Hence, identifying terms in (2), a permutation matrix associated to $\boldsymbol{\sigma}_*$ in (1) is $\Pi_{\boldsymbol{\sigma}_*} = \Pi_{\boldsymbol{\sigma}_z}^t\Pi_{\boldsymbol{\sigma}_x}$. In practice one does not need to manipulate two large matrices: an optimum post-processed forgery $\mathbf{y} = \Pi_{\boldsymbol{\sigma}_*}\mathbf{x}$ (not unique due to sorting ties) can be obtained by replacing the $h_q(\mathbf{x})$ largest elements of $\mathbf{z}$ by v_q, the next $h_{q-1}(\mathbf{x})$ largest elements of $\mathbf{z}$ by v_{q-1}, et cetera. The complexity of this operation is that of sorting two vectors, as evinced by (2), and the worst-case complexity of the best sorting algorithms is only $O(n \log n)$.

Additionally, the minimum distortion $\|\mathbf{z} - \mathbf{y}\|$ can be lower bounded using the geometry of permutation coding (see [9, Section II-C]). Since $\mathbf{z}$ cannot be closer to $\mathbf{y}$ than to the projections of $\mathbf{z}$ on the permutation sphere $\|\mathbf{u}\| = \|\mathbf{x}\|$, the covering sphere $\|\mathbf{u} - \overline{\mathbf{x}}\| = R_c$ (where $\overline{\mathbf{x}} \triangleq (\mathbf{x}^t\mathbf{1}/n)\mathbf{1}$ and $R_c^2 \triangleq \|\mathbf{x}\|^2 - (\mathbf{x}^t\mathbf{1})^2/n$), and the permutation plane $\mathbf{1}^t\mathbf{u} = \mathbf{1}^t\mathbf{x}$, then it follows that

$$\|\mathbf{z} - \mathbf{y}\| \geq \max\left\{\big|\|\mathbf{z} - \overline{\mathbf{x}}\| - R_c\big|, \big|\|\mathbf{z}\| - \|\mathbf{x}\|\big|, \left|\frac{\mathbf{1}^t}{\sqrt{n}}(\mathbf{z} - \mathbf{x})\right|\right\}.$$

This novel bound, although not always tight, may also find application in the use of permutation codes as source codes [7].

Finally, see from inequality (2) that the minimum distortion $\|\mathbf{z} - \mathbf{y}\|$ must be the same whether we use $\mathbf{x}$ or $\Pi\mathbf{x}$ as a decoy: the histogram $\mathbf{h}(\mathbf{x})$ is all that matters in order to find an optimum $\mathbf{y}$.

3. DECOY CHOICE

Up to this point "minimum distortion" has referred to the optimum distortion for a given decoy $\mathbf{x}$. The remaining question is finding a decoy that will globally minimise distortion over the space of all legitimate signals, that is to say,

$$\mathbf{x}^* = \arg\min_{\substack{\mathbf{x}\in\mathcal{V}^n \\ \phi(\mathbf{x})=1}} \min_{\boldsymbol{\sigma}\in\mathcal{S}_n} \delta(\mathbf{z}, \Pi_{\boldsymbol{\sigma}}\mathbf{x}).$$

The solution to this problem depends on the specific knowledge that the forger has about the forensic detector. Two decoy choice scenarios can be considered:

- **Scenario 1**: *The forger does not know the forensic detector* (blind counterforensics). The solution put forward by Barni et al. [3] for this scenario is to look in a database of legitimate signals for a decoy $\mathbf{x}$ whose histogram is close to that of the forgery $\mathbf{z}$. If the database is large then it is likely that a good match will be found. A procedure is given in [3] for obtaining $\mathbf{y}$ from $\mathbf{x}$ and $\mathbf{z}$, based on histogram and pixel remapping techniques related to transportation theory [10]; this method is able to enforce a target distortion, at the cost of an imperfect post-processed forgery $\mathbf{y}$. However it can always be replaced by the method in Section 2 to get a minimum-distortion perfect post-processed forgery, which is normally very faithful due the huge space of possibilities in the combinatorial optimisation. One issue with this approach occurs when the forensic investigator has memory and sees t different signals with the same histogram: as this is unlikely, he can suspect that, at least, $t-1$ of them are illegitimate. If the forger still wishes to enforce a distortion ν smaller than the minimum (and thus an imperfect post-processed forgery), he can do so by choosing $\mathbf{y}^\nu = \mathbf{z} - \nu(\mathbf{z}-\mathbf{y})/\|\mathbf{z}-\mathbf{y}\|$ instead of $\mathbf{y}$.
- **Scenario 2**: *The forger knows the forensic detector* (nonblind counterforensics). Comesaña and Pérez-González [4] have shown that in this scenario it is possible to find an ideal decoy $\mathbf{x}^*$ ($\mathbf{y}^\#$ in the notation of [4]) under mild assumptions about the detection function[2]. More precisely, they show that it is possible to find the histogram of an ideal decoy, which is all that matters as discussed at the end of Section 2. From the ideal decoy $\mathbf{x}^*$, these authors find an optimum solution $\mathbf{y}$ to minimum-distortion perfect counterforensics by relying on transportation theory. It must be remarked that this optimum is the same as the one derived in Section 2 by means of permutation decoding (cf. [4, expression (6)] with the upper bound in (2)).

3.1. Alternative decoy choice strategy for Scenario 1

Scenario 2 —which becomes plausible by invoking Kerckhoffs' principle [4]— is clearly the most desirable one. However, in many cases the forger can find himself in Scenario 1 and still mount a very effective decoy choice strategy without recourse to any database of legitimate signals. In effect, the forger always knows both the legitimate original signal (which we denote by $\mathbf{w}$) and the function $\varphi : \mathcal{V}^n \to \mathcal{V}^n$ used to produce the forgery $\mathbf{z}$ from it, $\mathbf{z} = \varphi(\mathbf{w})$. Therefore, he is aware of the type of artifacts that may appear in the histogram of the forgery by comparing it against $\mathbf{h}(\mathbf{w})$. Most importantly, as it will be seen in Section 4, he may also devise nearly ideal corrective measures to avoid them altogether. Some typical examples of this situation are forgeries produced by means of γ-correction, histogram stretching or (double) JPEG compression.

This opens the door to an alternative decoy choice strategy for Scenario 1, in which the forger fashions his own synthetic decoy from scratch in two steps: 1) fabrication of an artifact-free synthetic histogram $\mathbf{h}(\mathbf{x})$ for the post-processed forgery, based on the knowledge of $\mathbf{h}(\mathbf{w})$ and φ; and 2) generation of a synthetic decoy $\mathbf{x}$ with $\mathbf{h}(\mathbf{x})$ as its histogram. For the reasons discussed at the end of Section 2, any signal whose histogram is $\mathbf{h}(\mathbf{x})$ will do as a synthetic decoy, for instance

$$\mathbf{x} \triangleq \overrightarrow{\mathbf{x}} = [\underbrace{v_1, \cdots, v_1}_{h_1(\mathbf{x})}, \underbrace{v_2, \cdots, v_2}_{h_2(\mathbf{x})}, \cdots, \underbrace{v_q, \cdots, v_q}_{h_q(\mathbf{x})}]^t \qquad (3)$$

Although step 1) above is not novel (see for instance the counterforensic procedure by Stamm et al. in [12]) observe that a minimum-distortion post-processed forgery is only guaranteed through the use of permutation decoding. Although a post-processed forgery obtained from a synthetic histogram cannot be called "perfect", since the forger has no unequivocal guarantees about the legitimacy of the synthetic decoy, we will see in Section 4 that the forger often stands a good chance of producing a nearly ideal synthetic histogram, which, coupled with permutation decoding, generally improves by a large margin the fidelity results achievable in Scenario 1 when using a database of legitimate signals to find a decoy. To conclude, observe that the forger should not contemplate the use of $\mathbf{w}$ as a decoy, even though he knows it is legitimate: if he would do so, $\mathbf{y}$ would typically turn out to be closer to $\mathbf{w}$ than to $\mathbf{z}$, and then $\mathbf{y}$, although legitimate, could not be considered a proxy for the forgery anymore.

4. EMPIRICAL RESULTS

In this section we illustrate the approaches discussed in Sections 2 and 3.1 by means of a concrete practical case. We focus on forgeries of greyscale images whose samples are represented with 8 bits; therefore $q = 256$ and $\mathbf{v} = [0, 1, 2, \cdots, 255]^t$. We assume that the forgery $\mathbf{z}$ is the γ-corrected version of the original $\mathbf{w}$, that is, $z_i = \text{round}\,(255\,(w_i/255)^\gamma)$ for $i = 1, 2, \cdots, n$. It is well known that this operation introduces telltale traces in the histogram of $\mathbf{z}$ in the form of peaks and troughs. These artifacts can be exploited by a forensic detector, for instance through frequency analysis (see [13]).

We will address next the decoy choice approach outlined in Section 3.1. In order to fabricate an artifact-free synthetic histogram, the forger must examine the reason why artifacts appear in the first place. In the case at hand, artifacts are clearly due to the application of γ-correction to discrete intensity values, for which rounding is required. If the intensity values were continuous there would be no need for rounding, and then no artifacts would appear. Therefore the artifact-free synthetic histogram must be obtained through γ-correction of a continuous version of $\mathbf{h}(\mathbf{w})$.

In order to implement this strategy, the forger starts with the probability mass function (pmf) $\mathbf{p}(\mathbf{w}) = (1/n)\mathbf{h}(\mathbf{w})$ with support $\mathbf{v}$. Then he obtains a continuous counterpart of $\mathbf{p}(\mathbf{w})$ through interpolation, which yields the probability density function (pdf) $f_W(w)$. The application of γ-correction to the continuous random variable W does not require rounding, and hence it is just $X = 255\,(W/255)^\gamma$. The standard result for the transformation of continuous distributions yields the pdf of the transformed variable X, which is

$$f_X(x) = f_W\left(255\left(\frac{x}{255}\right)^{\frac{1}{\gamma}}\right)\left|\frac{1}{\gamma}\left(\frac{x}{255}\right)^{\frac{1}{\gamma}-1}\right|.$$

[2] In all likelihood, the same should be possible when the forger can only access the forensic detector as a black-box oracle (see [11]).

Original ($\mathbf{w}$) Forgery ($\mathbf{z}$) Post-processed forgery ($\mathbf{y}$)

Fig. 2. $\gamma = 2$, PSNR$(\mathbf{z}, \mathbf{y}) = 53.4$ dB (upper bound: 69.8 dB)

Original ($\mathbf{w}$) Forgery ($\mathbf{z}$) Post-processed forgery ($\mathbf{y}$)

Fig. 3. $\gamma = 1/2$, PSNR$(\mathbf{z}, \mathbf{y}) = 52.1$ dB (upper bound: 57.8 dB)

Next $f_X(x)$ is numerically integrated over $x \in [v_k - 1/2, v_k + 1/2)$ for $k = 2, 3, \cdots, q-1$, and over $x \in (-\infty, v_1 + 1/2)$ and $x \in [v_q - 1/2, +\infty)$, thus approximately obtaining the pmf $\mathbf{p}(\mathbf{x})$. The artifact-free synthetic histogram is $\mathbf{h}(\mathbf{x}) = \text{round}(n\,\mathbf{p}(\mathbf{x}))$. It is not guaranteed that the elements of $\mathbf{h}(\mathbf{x})$ add up to n, because of the final rounding and because of interpolation and numerical integration inaccuracies. If $\mathbf{1}^t\mathbf{h}(\mathbf{x}) < n$, a workaround is to increment by one $n - \mathbf{1}^t\mathbf{h}(\mathbf{x})$ elements of $\mathbf{h}(\mathbf{x})$ chosen at random; otherwise we should decrement by one $\mathbf{1}^t\mathbf{h}(\mathbf{x}) - n$ nonzero elements. For large n, this adjustment has negligible impact on the shape of the histogram. The final step is the generation of the synthetic decoy $\mathbf{x}$ as in (3), and of the post-processed forgery $\mathbf{y} = Q_{\mathbf{x}}(\mathbf{z})$ as discussed in Section 2.

The results of applying the procedure above to two γ-corrected images are shown in Figures 2 and 3. Although it might appear from a cursory visual inspection that $\mathbf{h}(\mathbf{y})$ does not faithfully follow the shape of $\mathbf{h}(\mathbf{z})$, notice that the way in which $\mathbf{h}(\mathbf{y})$ is produced from $\mathbf{h}(\mathbf{w})$ is nearly ideal with respect to γ-correction artifact removal. In fact, the forger should refrain from attempting a better "visual fit" by heuristically removing artifacts, such as, for instance, by smoothing peaks through filtering and by patching troughs through interpolation, as this would inevitably lead to much lower PSNR values. As discussed in Section 3.1, the forger cannot be completely certain about the undetectability of the synthetic decoy. However, the fact that the continuous approach is inherently free from γ-correction artifacts, as shown in the figures, suggests that it would be hard for a forensic detector to find evidence of γ-correction in the post-processed forgeries. A rigorous test should be undertaken to prove this, for example using the methods in [13]. Finally, a detector could also try to look for nonlinear resampling in the histogram domain.

In any case, the procedure just given is, in and of itself, of interest beyond counterforensics: it should also find application in image processing, since it enables repeated γ-corrections without accumulation of ill effects in the histogram. In fact one can see that permutation decoding is closely connected with exact histogram specification [14, 15]: it can be used as a procedure for finding the optimum version (in the Euclidean distance sense) of any arbitrary image with, exactly, some predefined target histogram.

5. CONCLUSIONS

We have given the solution to the problem of minimum-distortion perfect counterforensics when the forger only knows that the forensic detector uses first-order statistics and possesses a legitimate decoy. If the decoy can be optimally chosen, the solution found by means of permutation decoding is the same as the one found by Comesaña and Pérez-González [4] using transportation theory.

Nevertheless we must point out that the permutation coding view of the problem is more unifying and insightful, since it highlights the deeper theme of the connection between counterforensics and steganography: unlike transportation theory, permutation coding allows for optimum solutions in both scenarios. Furthermore permutation coding makes geometric reasoning possible. Apart from the analytic lower bounds on the minimum distortion given here, exploiting the geometry of the counterforensics setting can prove very useful in scenarios where distances are not preserved between the domain in which distortion is measured and the domain in which histograms are obtained (for instance, FSD counterforensics [16]).

The main lesson learned here is that future forensic detectors must take into account higher order statistics to be effective—like in steganography. However, both forger and steganographer have the upper hand as long as they stick to universal approaches (i.e. use of empirical rather than theoretical statistical models). As a concluding remark, observe that, despite their intrinsic connection, counterforensics and steganography are not dual problems in the sense that source coding and steganography are (see discussion in [9]), as they do not involve complementary use of the same pair of functions.

6. REFERENCES

[1] M. Kirchner and R. Böhme, "Tamper hiding: Defeating image forensics," in *Procs. of the 9th Int. Information Hiding Workshop*, 2007, pp. 326–341.

[2] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics*, Husrev Taha Sencar and Nasir Memon, Eds., pp. 327–366. Springer, 2013.

[3] M. Barni, M. Fontani, and B. Tondi, "A universal technique to hide traces of histogram-based image manipulations," in *14th ACM Int. Workshop on Multimedia Forensics and Security (MMSEC)*, Coventry, United Kingdom, September 2012.

[4] P. Comesaña and F. Pérez-González, "Optimal counterforensics for histogram-based forensics," in *38th IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, 2013, pp. 3048–3052.

[5] F. Balado and D. Haughton, "Permutation codes and steganography," in *38th IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 2954–2958.

[6] D. Slepian, "Permutation modulation," *Procs. of the IEEE*, vol. 53, no. 3, pp. 228–236, 1965.

[7] T. Berger, F. Jelinek, and J. Wolf, "Permutation codes for sources," *IEEE Trans. on Information Theory*, vol. 18, no. 1, pp. 160 –169, January 1972.

[8] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*, Cambridge at the University Press, 1934.

[9] F. Balado and D. Haughton, "Optimum perfect steganography of memoryless sources as a rate-distortion problem," in *5th IEEE Int. Workshop on Information Forensics and Security (WIFS)*, Guangzhou, China, November 2013.

[10] M. Barni and B. Tondi, "The source identification game: An information-theoretic perspective," *IEEE Trans. on Inf. Forensics and Security*, vol. 8, no. 3, pp. 450–463, March 2013.

[11] P. Comesaña, L. Pérez-Freire, and F. Pérez-González, "Blind Newton sensitivity attack," *IEE Procs. on Information Security*, vol. 153, no. 3, pp. 115–125, September 2006.

[12] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Anti-forensics of JPEG compression," in *IEEE Int. Conf. Acoustic, Speech, and Signal Processing (ICASSP)*, Dallas, USA, March 2010, pp. 1694–1697.

[13] M. C. Stamm and K. J. R. Liu, "Blind forensics of contrast enhancement in digital images," in *IEEE Int. Conf. Image Processing (ICIP)*, San Diego, USA, October 2008, pp. 3112–3115.

[14] D. Coltuc, P. Bolon, and J.M. Chassery, "Exact histogram specification," *IEEE Trans. Image Processing*, vol. 15, no. 5, pp. 1143–1152, May 2006.

[15] R. Chan, M. Nikolova, and Y.-W. Wen, "Exact histogram specification for digital images using a variational approach," *J. Math. Imaging Vision*, vol. 46, pp. 309–325, 2013.

[16] C. Pasquini and G. Boato, "JPEG compression anti-forensics based on first significant digit distribution," in *IEEE 15th Int. Workshop on Multimedia Signal Processing (MMSP)*, Pula, Italy, September 2013, pp. 500–505.