Feature Restoration and Distortion Metrics

Ventsislav K. Chonev and Andrew D. Ker

Oxford University Computing Laboratory, Parks Road, Oxford OX1 3QD, England.

ABSTRACT

Our work focuses on Feature Restoration (FR), a technique which may be used in conjunction with steganographic schemes to reduce the likelihood of detection by a steganalyzer. This is done by selectively modifying the stego image to reduce a given distortion metric to a chosen target feature vector. The technique is independent of the exact steganographic algorithm used and can be applied with respect to any set of steganalytic features and any distortion metric. The general FR problem is NP-complete and hence intractable, but randomized algorithms are able to achieve good approximations. However, the choice of distortion metric is crucial: our results demonstrate that, for a poorly chosen metric or target, reducing the distortion frequently leads to an *increased* likelihood of detection. This has implications for other distortion-reduction schemes.

Keywords: Steganography, Steganalysis, Feature Restoration, Mahalanobis Distance, Quadratic Programming, Numerical Stability, NP-Completeness

1. INTRODUCTION

When steganography is used to hide payload within a cover object, some statistics of the cover source are distorted. This presents the opportunity for a steganalyst to build a detector that can reliably determine whether an object contains a payload, given knowledge of the steganographic algorithm but not the embedding key. In turn, the steganographer may seek to reduce the statistical distortion caused by their embedding.

In digital images, most leading steganalysis algorithms are based on a vector of feature statistics chosen to be sensitive to steganographic embedding, but less so to the content of the images. Then machine learning techniques are used to train a classifier to distinguish between the features of innocent images and stego images. Examples of such detectors include Refs. 1–6. When facing such a detector, it is in the steganographer's interest to minimize the distortion that their embedding causes to features.

There are, broadly, three ways in which steganographers seek to reduce distortion to features. First, they may try to reduce the number of changes (to pixels, transform coefficients, or other image representation) by using coding tricks, such as conveying the message by syndromes of low covering-radius linear codes.[7,8] Certainly, if the absolute distortion to the image itself is kept low, the feature distortion and the risk of detection should also be low. Second, they may try to include an awareness of features in the embedding process itself, and when presented with a choice – for example, whether to increment or decrement a pixel when flipping a Least Significant Bit (LSB) – choosing the option which least distorts the feature vector. Such techniques, which are usually known as *distortion minimization* (DM), have now reached a high level of sophistication.[9–11]

The third option, and one which has been relatively little explored, is to embed a payload using a conventional technique, noting locations (pixels, transform coefficients, etc.) that are not used for payload. After the embedding is completed, the unused locations are modified to move the stego object's features towards a target vector that is believed to appear innocent. To our knowledge, the only literature on this idea is Ref. 12. We call these techniques *feature restoration* (FR). We emphasise the difference between DM and FR ideas: the former is part of the embedding process itself, the latter applied post-hoc to "spare" parts of the cover. In principle, FR could be applied after other distortion reduction strategies have already been applied. Both DM and FR require a metric for *distortion* of feature vectors, which determines what they choose to minimize, and selection of the right distortion metric will be an important theme in this paper.

Further author information: (Send correspondence to ADK):
A. D. Ker: E-mail: adk@comlab.ox.ac.uk, Telephone: +44 1865 283530
V. Chonev: Email: v.chonev@gmail.com

In such work it is assumed either that the steganographer knows the features which will be used by the opponent, or that they minimize or reduce distortion with respect to a sufficiently large and general feature set that it serves as a proxy for all such detectors. With the present state-of-art, it is not implausible that the steganographer might guess their opponent's behaviour, because the literature contains only a few leading examples of steganalysis feature sets.[3–5]

In this paper, our scope will be limited to the WAM feature set from Ref. 2. They comprise 27 features, which are the absolute moments of residuals from a wavelet-based denoising filter. For the purposes of this work, the detail of how the WAM features are calculated is not very relevant, so it will be omitted. We also limit ourselves to the type of embedding for which WAM was designed: spatial-domain LSB Matching steganography in never-compressed grayscale images.

Although the topic of this paper is feature restoration, we consider this only a background for an illustration of two messages: that intractable optimization, of which FR will turn out to be an example, can be approximated by randomized iterative algorithms, and that steganographers must take great care in their choice of distortion metric. The latter is an important lesson for DM too.

In Sect. 2 we will formalize the FR problem, and discuss three options for a feature distortion metric. True minimization of the FR problem is not tractable, and Sect. 3 describes heuristics for approximate optimization algorithms, benchmarking them by their ability to reduce feature distortion with respect to the various distance metrics. There is also an important discussion about the numerical stability of distortion computations. Sect. 4 then examines whether minimized distortion indicates low detectability, and demonstrates that choosing the right distortion metric is vital to achieving the aim of FR. With the wrong choice of metric, reduced feature distortion can lead to *increased* probability of detection. Finally, Sect. 5 draws conclusions.

1.1 Notation

Throughout the paper, we will use uppercase Roman letters to denote original (cover) and stego objects (O, S), and lowercase boldface letters to denote the corresponding vector of features ($\boldsymbol{o}$, $\boldsymbol{s}$). All other vectors will also be written lowercase boldface ($\boldsymbol{t}$, $\mathbf{1}$ a vector of all 1s), all matrices uppercase Greek (Δ, Σ, I the identity matrix), and all sets uppercase calligraphic ($\mathcal{C}$).

The set of all objects of the type under consideration (e.g. uncompressed grayscale digital images of a particular size, or colour JPEG images of a particular size and quality factor) will be denoted $\mathcal{O}$. We assume that the aim is to counter steganalysis by a particular set of features, and will write $\phi : \mathcal{O} \to \mathbb{R}^p$ for the map from objects to feature vectors. The dimensionality of the feature set, p, will be implicit throughout.

2. FEATURE RESTORATION

Feature restoration (FR) is applied after a conventional embedding procedure. Locations in the cover which have not been used for payload, and are therefore free to alter without damaging the content, are selectively modified by small amounts with the goal of moving the stego object's features towards a target, believed to appear innocent to a steganalyzer. Modifications on non-payload components should be constrained to be slight, lest they cause visible distortions. The technique has the advantage of being independent of the embedding algorithm.

The FR problem can be formalized as follows. Consider an abstract set of allowable *changes* to a stego object: some combinations of changes may not be possible (for example if one change is to increment a particular pixel and another to decrement it then they cannot both be applied, and one or the other might take a pixel beyond saturation), so suppose a set of all allowable combinations of changes $\mathcal{A}$. Given a cover object O, and a corresponding stego object S (the result of embedding payload by conventional means), let us write $S + c_i$ to mean that change c_i has been applied to the object S. Then the FR problem is

$$\text{minimize} \quad d\Big(\boldsymbol{t}, \phi\big(S + \textstyle\sum_{c\in\mathcal{C}} c\big)\Big) \quad \text{subject to} \quad \mathcal{C} \in \mathcal{A}, \tag{1}$$

where $d : \mathbb{R}^p \times \mathbb{R}^p \to [0, \infty)$ is some distance metric, and $\boldsymbol{t}$ a *target* feature, which might be the features of the original cover $\phi(O)$, or some canonically least-suspicious feature vector such as the mean of features from a large cover image corpus.

As we shall see, the problem (1) is generally intractable, so we first try to simplify it by imposing conditions on the metric $d(-, -)$ and the response of features to multiple changes, outlined in the next two subsections.

2.1 Distance Metrics

Whether performing FR or DM, one must not forget that *reduced distortion does not necessarily imply reduced risk of detection.* Everything hinges on the way one measures distortion, in this case the distance metric between feature vectors.

We will consider quadratic form distance metrics:

$$d(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u} - \boldsymbol{v})^T \Delta (\boldsymbol{u} - \boldsymbol{v}) \tag{2}$$

where Δ is some fixed nonnegative definite symmetric matrix. The simplest such metric is

$$\Delta = \mathrm{I}, \tag{3}$$

the *Euclidean distance.* It suffers, however, from sensitivity to the scale of the features: if one feature has much larger magnitude (or naturally larger variance) than another, its value will dominate the distance calculations and potentially detectable small-scale differences in the other feature will be ignored. It also ignores correlation between features.

An alternative is

$$\Delta = \hat{\Sigma}^{-1}, \text{ where } \hat{\Sigma} \text{ is an estimate of the covariance matrix for features of cover objects,} \tag{4}$$

the *Mahalanobis distance.*[13] This takes account of the expected magnitude of, and correlations between, features; it is equivalent to Euclidean distance between vectors whitened by the matrix $\hat{\Sigma}^{-\frac{1}{2}}$. A disadvantage of the Mahalanobis distance is its need for an estimated covariance matrix: such estimates can be unstable (numerical stability aside, consider that the matrix contains p^2 entries, which could be large compared with the number of available samples). An alternative, sometimes used in the machine learning literature (e.g. Ref. 14) is the *standardized Euclidean distance* or *diagonal distance*,

$$\Delta = \Lambda, \quad \begin{array}{l}\text{the diagonal matrix with } (1/\hat{\sigma}_1^2, \ldots, 1/\hat{\sigma}_p^2) \text{ on the diagonal,} \\ \text{where } \hat{\sigma}_i^2 \text{ is an estimate of the variance of feature } i \text{ in cover objects.}\end{array} \tag{5}$$

This can be estimated more stably than the full covariance matrix because only variances are required; it takes account of features having different magnitude or variance, but does not account for correlation between features and therefore may present a weakness.

We also have a choice about the *target vector* $\boldsymbol{t}$ for the restoration. An obvious choice is

$$\boldsymbol{t} = \phi(O), \tag{6}$$

the features of the cover before any embedding was performed. But if the cover itself is something of an outlier, the restoration might be suboptimal. An alternative, then, is

$$\boldsymbol{t} = \hat{\boldsymbol{\mu}}, \text{ where } \hat{\boldsymbol{\mu}} \text{ an estimate of the mean feature vector in cover objects.} \tag{7}$$

The aim is to move the features towards the "least suspicious" location.

We will consider each metric (3)–(5) and both targets (6) and (7) in all combinations. Where necessary, $\hat{\Sigma}$ and $\hat{\boldsymbol{\mu}}$ will be estimated from a corpus of cover images: we will discuss briefly whether the corpus is big enough to get a sufficiently stable estimate in Subsect. 3.1. In Sect. 4 we will see that some options for distortion metric work significantly better than others.

2.2 Additivity & Quadratic Programming

There are two ways in which we might attempt to reduce the complexity of the problem (1).

First, we seek to understand the effect of an individual change, computing

$$\boldsymbol{\delta_c} = \phi(R+c) - \phi(R),$$

where R represents a fixed object and c a possible change (e.g. incrementing one pixel). We wish to determine $\boldsymbol{\delta_c}$ as efficiently as possible, preferably without two calculations of ϕ. This is possible if we can trace the pixel-level effect of c through to the resulting change in features. Such an idea is used on the PEV-274 feature set[4] in Ref. 12 where it is called *differential feature computation.* The WAM features are highly nonlinear, but both the wavelet filter and the denoising operation use only local regions of pixels (though the moments have to be recomputed in their entirety), so we have been able to cache parts of the computation and perform a type of differential feature calculation for our purposes too: for details, see Ref. 15. This enables us to compute $\boldsymbol{\delta_c}$ much more quickly than a computation of ϕ.

Second, we try to reduce the very large search space in (1). In principle one should search all allowable *combinations* of changes, which is totally infeasible. We can simplify the problem if we believe that the effect, on a feature vector, of two separate changes equals the sum of the effect of each individually. When this happens we say that the changes have *additive* effect.

If all changes have additive effect,

$$\phi\big(R + \textstyle\sum_{c\in\mathcal{C}} c\big) = \phi(R) + \textstyle\sum_{c\in\mathcal{C}} \boldsymbol{\delta_c}, \quad \text{where } \boldsymbol{\delta_c} = \phi(R+c) - \phi(R),$$

for all $R \in \mathcal{O}$ and all sets of changes $\mathcal{C}$. We would *not* expect this to be exactly true for features like WAM, based on residuals of denoising filters, because of the local region used in the filter: when multiple changes occur in the same region, their effects may cancel or magnify. Neither is it exactly true for other leading feature sets such as PEV-274[4] or SPAM.[6] However, when changes are spatially well-separated, the interaction between their effects on features is expected to be small. For WAM features it turns out that the interaction remains small even for changes which are fairly close together and some empirical evidence, that *approximate* additivity holds almost always, can be found in Ref. 15. This will allow us to make an approximation to try to reduce the complexity of (1).

Consider, then, (1) under an assumption of exact additivity. Suppose that all possible changes, $\bigcup\mathcal{A}$, are enumerated $c_1, \ldots, c_n$. Then any set of changes can be specified by a binary vector $\boldsymbol{x} \in \{0,1\}^n$, with $x_i = 1$ indicating that change c_i is included. Write $\Gamma \in \mathbb{R}^{p\times n}$ for the matrix whose columns are $\boldsymbol{\delta_{c_1}}, \ldots, \boldsymbol{\delta_{c_n}}$. With convex quadratic distance as in (2), the objective function of the FR problem is

$$\begin{aligned} d\Big(\boldsymbol{t}, \phi\big(S + \textstyle\sum_{c\in\mathcal{C}} c\big)\Big) &= \big(\boldsymbol{s} + \textstyle\sum_{c\in\mathcal{C}} \boldsymbol{\delta_c} - \boldsymbol{t}\big)^T \Delta \big(\boldsymbol{s} + \textstyle\sum_{c\in\mathcal{C}} \boldsymbol{\delta_c} - \boldsymbol{t}\big) \\ &= (\boldsymbol{s}-\boldsymbol{t})^T\Delta(\boldsymbol{s}-\boldsymbol{t}) + \big(\textstyle\sum_{c\in\mathcal{C}} \boldsymbol{\delta_c}\big)^T \Delta \big(\textstyle\sum_{c\in\mathcal{C}} \boldsymbol{\delta_c}\big) + 2(\boldsymbol{s}-\boldsymbol{t})^T\Delta\big(\textstyle\sum_{c\in\mathcal{C}} \boldsymbol{\delta_c}\big) \\ &= (\boldsymbol{s}-\boldsymbol{t})^T\Delta(\boldsymbol{s}-\boldsymbol{t}) + \boldsymbol{x}^T\Gamma^T\Delta\Gamma\boldsymbol{x} + 2(\boldsymbol{s}-\boldsymbol{t})^T\Delta\Gamma\boldsymbol{x}. \end{aligned}$$

The first term is constant so, if additivity holds, the FR problem can be written as

$$\text{minimize} \quad \boldsymbol{x}^T\Gamma^T\Delta\Gamma\boldsymbol{x} + 2(\boldsymbol{s}-\boldsymbol{t})^T\Delta\Gamma\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{x} \in \mathcal{A}', \tag{8}$$

where $\mathcal{A}'$ is the set of vectors in $\{0,1\}^n$ corresponding to the permissible combinations of changes $\mathcal{A}$. The formulation (8) is a quadratic programming problem.

Unfortunately, even though this version of the FR problem is favourable (as it assumes exact additivity), it is still not tractable: in the Appendix, we prove that the problem is NP-complete. This means that we cannot expect to find an efficient algorithm for solving the minimization exactly; we must look for heuristics which find good solutions within reasonable time limits. Some of the heuristics will attempt to exploit approximate additivity, but others will perform heuristic minimization directly on the original problem (1).

3. HEURISTICS FOR FEATURE RESTORATION

We now propose and evaluate some heuristics for tackling the FR problem. In this section, we will only consider the ability of the algorithms to minimize the chosen distance to the chosen target; we will then consider how well this reduces *detectability* in Sect. 4.

At the heart of FR algorithms are two tools: a feature oracle and a distance calculator. The feature oracle is able to compute the feature function ϕ, and the effect of any change on any image. Conceptually, it should be thought of as a black box. However, since it will be heavily relied on, its workings should be as efficient as possible. As mentioned in section 2.2, techniques such as differential feature computation may be used to boost its time performance significantly. The distance calculator is a simpler tool which computes the quadratic form distance metric d, as defined in section 2.1, for any choice of Δ.

The FR heuristics maintain a *current image* R, with $R = S$, the stego image, initially. In the exposition, we use the following terminology. Firstly, applying a change c is taken to mean setting $R := R + c$. Secondly, if $\mathbf{t}$ is the choice of target, testing c is defined as computing $d(\phi(R + c), \boldsymbol{t})$ and comparing it to $d(\phi(R), \boldsymbol{t})$. If $d(\phi(R + c), \boldsymbol{t}) < d(\phi(R), \boldsymbol{t})$, the change is considered beneficial. Finally, if $d(\phi(R + c_1), \boldsymbol{t}) < d(\phi(R + c_2), \boldsymbol{t})$, then c_1 is considered more beneficial than c_2. The same applies to sets of changes.

Our heuristics for FR are briefly outlined below. They all treat the feature oracle and distance calculator as black boxes, making them applicable to various choices of steganalytic features and distortion metrics. For more detailed descriptions of the algorithms and the motivation behind their workings, see Ref. 15.

- **Slow Greedy Algorithm.** Test every change which increments or decrements a single unused pixel and apply only the most beneficial one. Repeat to find the next most beneficial change, and continue until no beneficial changes remain.

- **Fast Greedy Algorithm.** Test every change which increments or decrements a single unused pixel and, whenever a beneficial change is found, apply it immediately. (We understand this to be the method proposed in Ref. 12.)

- **Biased Greedy Algorithm.** A greedy algorithm as above, but which performs a local variance estimation in the spatial domain to prioritise noisy regions of the image: changes to these regions are examined first and are performed for as long as it is beneficial to do so.

- **Random Algorithm.** Pick a random selection of changes $\mathcal{C}$ and test whether it is beneficial. If so, apply it. Then repeat. We experimented with different options (see Ref. 15), and settled on a limit of 15 single-pixel changes in each random batch. We also included an adaptive behaviour which reduces the size of the batch by 25% if 100 consecutive iterations fail to find a beneficial $\mathcal{C}$. This can be seen as a very simple form of simulated annealing.[16]

- **Genetic Algorithm.** A greedy heuristic which borrows ideas from genetic algorithms. A population is maintained, consisting of pairs $(\mathcal{C}, x)$, where $\mathcal{C}$ is a set of changes and x the distance reduction entailed by applying $\mathcal{C}$. The label x is correct when the pair is introduced into the population, but is not updated when the current image R is changed. Thus, the labels x may become outdated, but the assumption of approximate additivity guarantees that the inaccuracy will be small. On each iteration, the change with the greatest distance reduction label is applied and removed from the population, and two new pairs $(\mathcal{C}, x)$, randomly chosen, are added to it. When the population reaches a certain size, its elements are combined (by taking the union of changes) in pairs, thereby halving its size.

- **Quadratic Programming Algorithm.** A set of admissible changes $\mathcal{C}$ is chosen at random. Under the assumption of exact additivity the FR subproblem, considering only changes in $\mathcal{C}$, is expressed as a Binary Integer Quadratic Problem following the exposition in section 2.2. The integrality constraints on the unknowns are relaxed[17] and replaced by $0 \leq x_i \leq 1$, allowing the resulting problem to be solved efficiently using known quadratic programming techniques.[18] The answer is rounded to the nearest integer, the appropriate changes are applied, and the algorithm is repeated with different choices of $\mathcal{C}$.

We will compare the efficiency of the algorithms by their ability to reduce the chosen distance metric. In order to abstract away from their implementation efficiencies, we will measure an algorithm's cost as the number of feature calculations it performs: as an increasing number of feature calculations is permitted, the best algorithms will make more progress towards minimizing the distortion.

Initial experiments allowed us to discard the Slow Greedy Algorithm at an early stage: it is so cautious, in only applying the very best change possible at each stage, that it requires far too many feature calculations to make sufficient progress.

3.1 Stability of Mahalanobis Distance

The Mahalanobis distance (4) requires an estimation of the inverse covariance matrix Σ^{-1}. In the course of our work, we discovered that the WAM features display multicollinearity, and hence the covariance matrix is ill-conditioned. Strong correlation is hardly surprising, given that the WAM features are absolute moments of the same vectors of residuals.

Unfortunately, the near-singularity of the covariance matrix raises stability issues which cannot be ignored, since computations can only be performed with a finite degree of accuracy. A small error, caused by estimating the covariance matrix from a small finite sample, might result in a large error in the estimation of the inverse if there is instability in the inversion algorithm, corrupting the distance measure and causing FR algorithms to apply the wrong changes. Alternatively, a small error in the computation of features themselves, caused by the limited precision of floating-point arithmetic, might result in large errors in the distance to target, and unduly influence the choice of whether to apply a particular change. In the worst case, the minimization heuristics might chase magnified rounding errors instead of true improvements in distortion.

This is an important consideration for researchers who work with multicollinear features (and almost all feature sets commonly in use are highly redundant in this way). We considered the issue in some detail in Ref. 19, where we analyzed the magnitudes of Mahalanobis distance errors that could be expected to arise from various imprecisions. The reader is referred to that technical report for the details: the conclusion is that, in our application, these covariance matrices are not *quite* sufficiently ill-conditioned for the errors in estimated Mahalanobis distance adversely to affect the heuristic optimizations.

3.2 Experimental Results

We will perform experiments on three sets of never-compressed grayscale images, the first two taken from Ref. 20.

1. **Set B**: 2000 RAW images taken with a Minolta DiMAGE A1 camera, denoised using the Minolta software with default options, then converted to grayscale and cropped to random 400 × 300 regions.

2. **Set C**: 2000 images supplied by the researchers at Binghamton University, never-compressed images taken with a mixture of 16 digital camera models (a subset of those used in Ref. 2) converted to grayscale and cropped to random 400 × 300 regions.

3. **BOSS**: 2000 images selected at random from the BOSSBase image set.[21] Never-compressed images which have been cropped and resampled to 512 × 512 pixels.

For each cover set $\mathcal{S}$, we estimated

$$\begin{aligned} \hat{\boldsymbol{\mu}} &= \frac{1}{|\mathcal{S}|}\sum_{O\in\mathcal{S}}\phi(O) \\ \hat{\Sigma} &= \frac{1}{|\mathcal{S}|-1}\sum_{O\in\mathcal{S}}(\phi(O)-\hat{\boldsymbol{\mu}})^T(\phi(O)-\hat{\boldsymbol{\mu}}) \end{aligned}$$

in order to calculate the distance measures (3)–(5).

We considered payload sizes between 0.5 bits per pixel (bpp) and 0.99 bpp. For each size, payloads were embedded using LSB Matching, and then each FR heuristic performed with a limit of 25000 feature calculations

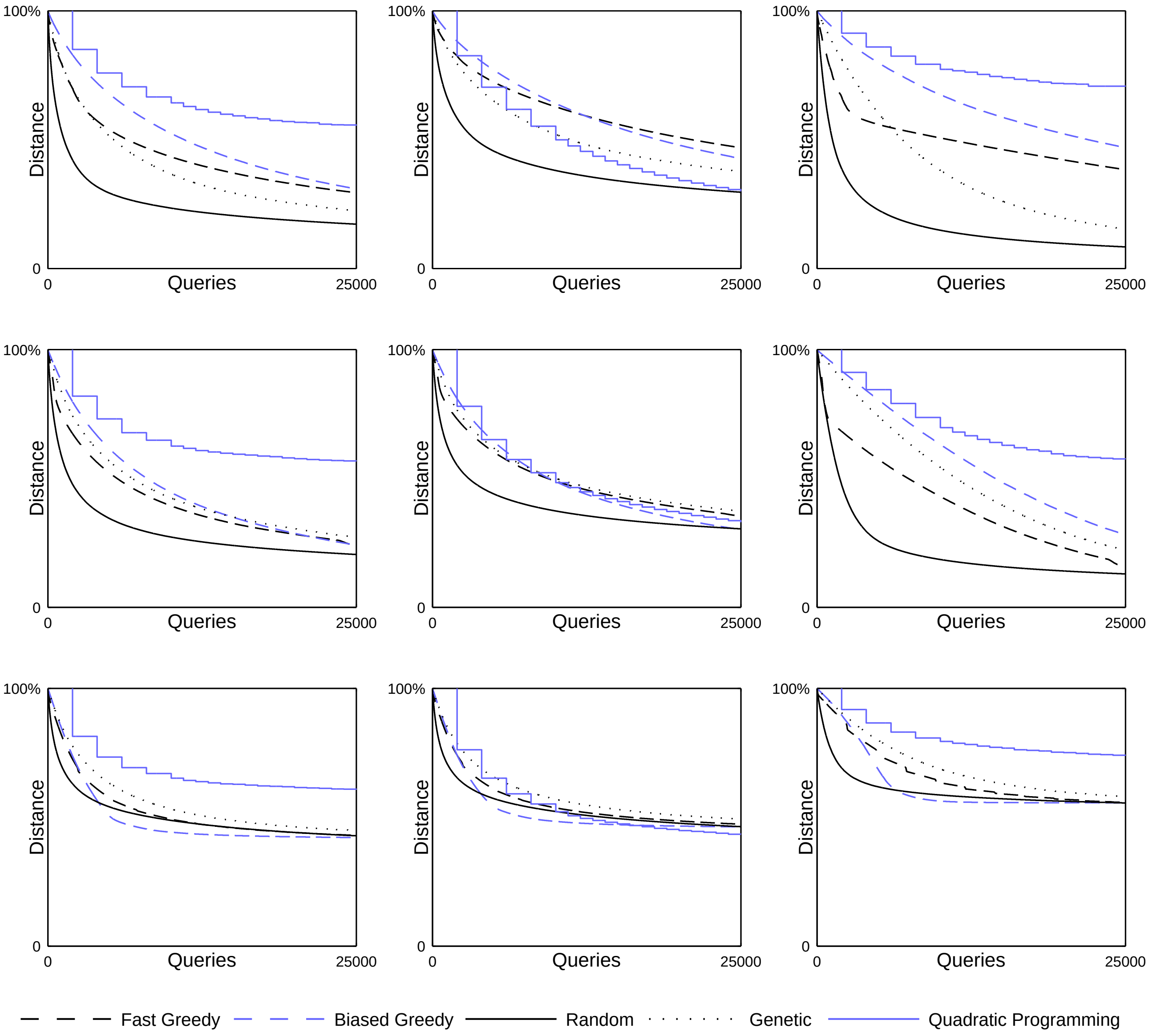

Figure 1. Comparison of FR heuristics, for various payloads and distortion metrics. From top to bottom: payload sizes 0.5 bpp, 0.9 bpp, 0.99 bpp. From left to right: Mahalanobis distance to original cover features, Mahalanobis distance to mean cover features, Euclidean distance to original cover features. Each chart displays the relative distance of the restored stego image compared with the unrestored stego image, averaged over all images in Set C, as the number of feature calculations ("queries") grows to a maximum of 25000.

per image, keeping track of the distance to the target at each stage as a fraction of the stego image's initial distance. This allowed us to compare the distance reduction achieved by each algorithm as the number of permitted feature calculations increases. We then graph this proportionate distance to target, averaging over all images, as a function of the number of feature calculations.

With images of this size, and running on a single core of a machine with a 2.67GHz Xeon processor, it takes about ten minutes to perform feature restoration limited to 25000 calculations of the WAM features. In order to test multiple image sets with 2000 images each, four different payload sizes, three choices of distortion metric and two target vectors, and five different minimization heuristics, we distributed the computation amongst a cluster with 50 cores. The experiments reported in this paper took a total of 6.4 core years to perform.

We display results only for Set C (the others are similar). The cases of Mahalanobis distance (4) and target vector either the cover features $\phi(O)$ or estimated mean cover vector $\hat{\boldsymbol{\mu}}$, and also Euclidean distance (3) and target $\phi(O)$, and three payload sizes, are shown in Fig. 1. Despite a lack of sophistication, the Fast Greedy Algorithm does not perform too badly; conversely, the complicated Quadratic Programming Algorithm (whose jumps in distance reduction are a consequence of the large number of feature computations required to set up each FR subproblem) does not perform well. In almost every case (including cases not displayed), the Random Algorithm most quickly reduces the distance, though for very large numbers of queries all algorithms except Quadratic Programming appear to converge to roughly the same performance: perhaps this is the true objective minimum of (1). The distortion metric and target feature affect the results, but generally not the relative performance of the different heuristics.

The results suggest that for the purposes of large distance reduction in few oracle queries, the Random Algorithm provides the best option (for very large payloads, where there are few pixels left to change, the Biased Greedy Algorithm may be used). It is not uncommon for randomized algorithms to provide good approximate solutions to intractable problems.[22] Our investigation into heuristics for FR also suggests the following rules of thumb:

1. **Greediness.** When you spot a beneficial change, perform it, instead of being cautious and looking for even better ones.
2. **Conserving feature calculation.** Try to perform many changes before recomputing features, possibly exploiting additivity.

Finally, note that the relative distance, after restoration, is much the same whether the payload is 0.5 bpp or 0.9 bpp: although the former leaves many more pixels available to make feature-restoring changes, the larger search space negates any advantage. Only for very large payloads such as 0.99 bpp does the restricted search space impede the restoration.

4. FEATURE RESTORATION AND DETECTABILITY

Although the Random Algorithm might be successful at restoring the features of a stego image towards a target, this does not necessarily mean that it is successful at evading detection. Everything hinges on the choice of distortion metric: the distance we use to compare features, and the target for the restoration. It is even conceivable that the process of feature restoration makes stego images even more detectable. In this section, we take stego images which have been subject to FR and test them against steganalysis detectors.

The original presentation of WAM steganalysis[2] applied a Fisher Linear Discriminant classifier to features; we also tested a simple linear SVM. Such classifiers need to be trained, so the images of the previous section were divided into equal-sized training and testing sets (the same images were allocated to the training or testing set each time, including when there are different-sized payloads, to avoid additional perturbations in the accuracy). The question then arises: what is the proper training set? Do we train on cover and stego images, or cover and *restored* stego images?

This goes to the heart of the steganalysis problem, asking whether we should take the role of a detector who knows that the embedder is attempting FR (and indeed what sort of FR is being used) or whether the detector only suspects plain embedding, in this case LSB Matching. We adopt the usual computer security convention, and assume that the detector knows the entire system used by the embedder, including that feature restoration is taking place, and therefore it is appropriate to train on cover and restored stego images before testing on a (different) set of cover and restored stego images.

We then benchmark the accuracy of the trained classifier on the test images: the proportion of images correctly classified as cover or (restored) stego images. Classifier sensitivity can be adjusted (by changing the objective functions in the FLD or SVM) to trade false positive detection for false negative detection, so we

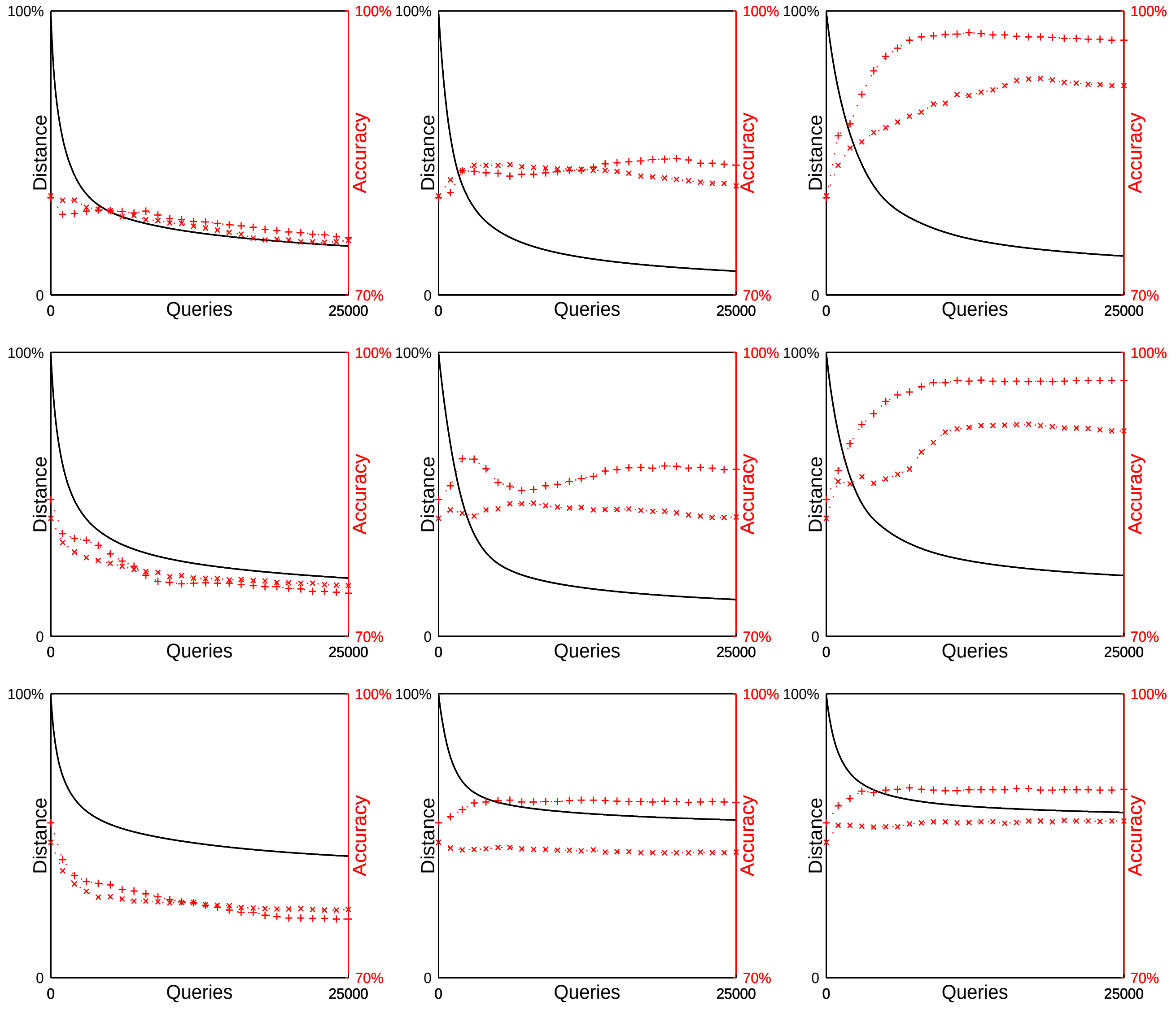

Figure 2. Detectability of Set C stego images after FR by the Random Algorithm, for various payloads and distortion metrics. From top to bottom: payload sizes 0.5 bpp, 0.9 bpp, 0.99 bpp. From left to right: Mahalanobis distance to original cover features, Euclidean distance to original cover features, Euclidean distance to mean cover features. Each chart displays both the relative distance of the restored stego image compared with the unrestored stego image (left axis), and the detection accuracy of the FLD (+) and SVM (×) classifiers based on WAM features (right axis), as the number of feature calculations during restoration grows to a maximum 25000.

report the maximum accuracy over all possible thresholds. This is equivalent to the benchmark commonly used in steganalysis literature,

$$\max(1 - P_E) = 1 - \tfrac{1}{2}\min(P_{FP} + P_{FN})$$

where P_{FP} and P_{FN} indicate the false positive and false negative error probabilities.

Some of the results are displayed in Fig. 2, for payloads of 0.5 bpp, 0.9 bpp, and 0.99 bpp in Set C, and three distortion metrics: Mahalanobis distance to original, Euclidean distance to original, and Euclidean distance to a mean cover feature vector. We limit ourselves to the Random Algorithm, since it has been shown to be most effective at restoring features to their target. We also show the final accuracy, after a limit of 25000 computations

Table 1. Comparison of detector (WAM features with Fisher Linear Discriminant) accuracy, before and after feature restoration limited to 25000 feature calculations. Three images sets, four payload sizes, and six combinations of distortion metric and restoration target. Accuracy displayed to 3 sig. fig.

Covers	Payload	WAM/FLD accuracy on stego images	Accuracy after restoration using					
			$\Delta{=}\hat{\Sigma}^{-1}$ $\boldsymbol{t} = \phi(O)$	$\Delta{=}\hat{\Sigma}^{-1}$ $\boldsymbol{t} = \hat{\boldsymbol{\mu}}$	$\Delta = \mathrm{I}$ $\boldsymbol{t} = \phi(O)$	$\Delta = \mathrm{I}$ $\boldsymbol{t} = \hat{\boldsymbol{\mu}}$	$\Delta = \Lambda$ $\boldsymbol{t} = \phi(O)$	$\Delta = \Lambda$ $\boldsymbol{t} = \hat{\boldsymbol{\mu}}$
Set B	0.5 bpp	72.0%	52.3%	62.9%	63.4%	93.2%	59.1%	93.6%
Set B	0.7 bpp	76.8%	54.0%	61.7%	66.5%	93.5%	63.0%	94.0%
Set B	0.9 bpp	80.6%	57.1%	66.0%	69.5%	92.8%	68.8%	93.2%
Set B	0.99 bpp	81.6%	75.2%	79.8%	78.2%	82.0%	77.7%	82.0%
Set C	0.5 bpp	80.2%	76.0%	77.0%	83.7%	96.9%	81.3%	96.1%
Set C	0.7 bpp	81.5%	75.2%	77.5%	84.8%	97.1%	85.8%	97.0%
Set C	0.9 bpp	84.5%	74.6%	77.9%	87.6%	97.0%	88.6%	96.7%
Set C	0.99 bpp	86.4%	76.2%	75.9%	88.5%	89.9%	89.2%	89.5%
BOSS	0.5 bpp	65.9%	55.2%	57.8%	56.3%	90.7%	53.9%	84.2%
BOSS	0.7 bpp	66.5%	53.2%	58.6%	59.5%	91.0%	56.7%	85.1%
BOSS	0.9 bpp	67.2%	54.3%	59.7%	65.0%	90.1%	63.1%	83.2%
BOSS	0.99 bpp	68.0%	61.0%	65.5%	65.3%	68.6%	66.2%	69.3%

of features, in Tab. 1; only the FLD classifier accuracies are displayed, but all three cover sets, four payload sizes, and all six combinations of distance measure with restoration target can be included.

These results show that feature restoration *can* be successful in reducing detectability: with Set C images, for example, the WAM/FLD detector accuracy drops from 84.5% on 0.9 bpp stego images to 74.6% on the same images after feature restoration, if the Mahalanobis distance is used and the target vector is that of the corresponding cover. Unsurprisingly, there is less scope for reducing detectability in images with 0.99 bpp payload, because only 1% of pixels are available to change during restoration, but even then the accuracy of the detectors is significantly reduced. Payloads of 0.99 bpp, subject to feature restoration, become approximately as detectable as payloads of 0.5 bpp or less, which have not been subject to restoration.

Perhaps contrary to intuition, restoration of images with only 0.5 bpp of payload seems marginally more difficult than with 0.9 bpp, with the detector accuracy usually not reducing by quite as much in the case of the smaller payload. We attribute this to the relatively larger search space for the problem (1) when a large proportion of the image is unused for embedding, but the phenomenon bears further investigation.

The charts of Fig. 2 also show that most of the reduction in detector accuracy occurs within a few hundred or thousand feature calculations, especially for images with larger payloads. So it is not necessary to expend nearly as many as the 25000 queries we performed. Nonetheless, the computation time of the feature restoration may become significant in larger images (where there is a larger space of potential changes to search), or for features that are slow to compute.

The results also illustrate that the distortion metric is quite crucial to success in evading detection. In the charts of Fig. 2, accuracy is observed to decrease (regardless of whether an FLD or SVM classifier is used) as more work is done in restoration with respect to Mahalanobis distance to original features. Accuracy stays roughly the same when restoration uses Euclidean distance to original features, and *increases* when restoration uses Euclidean distance to the mean cover feature. In some cases the increase is dramatic. It is clear, from Tab. 1, that it is incorrect to restore to a target which is the mean feature vector from a corpus of covers. Consider embedding 0.9 bpp payload in Set C, for example: WAM detector accuracy *increases* from 84.5% to 97.0% on restored images, even though the restoration was "successful" in reducing the Euclidean distance to this target. In fact, we attribute the results to *overly-successful* restoration: the restored stego objects all cluster around the same target point, which allows classifiers to detect them easily. This is born out by the results for 0.99 bpp payloads, where the same effect is not as marked because the restoration has few pixels to work with and so is unable to make restored stego objects as predictable.

We conclude that the choice of distortion metric is of key importance. One must not, at least for WAM features, disregard the strong correlations between features when attempting the restoration. We also observe that performance after restoration to standardized Euclidean distance (5) is not much better than with plain Euclidean distance and certainly does not match that of the Mahalanobis distance: although standardized Euclidean distance does not present the same numerical instability as Mahalanobis distance, disregarding correlations between features results in a severe penalty.

5. CONCLUSIONS

We have demonstrated that, under certain conditions, feature restoration can be added to conventional embedding to reduce the likelihood of detection. This is despite it being an NP-complete optimization problem. Considering the WAM features for detection of LSB Matching steganography, we showed that detector accuracy can be substantially reduced when tested on a number of different cover image sets. The restoration seems to work best for large payloads, around 0.9 bpp, and succeeds substantially even for payloads as high as 0.99 bpp, when only 1% of the cover is available for alteration during restoration. This is complementary to the sorts of coding tricks[7,8] which reduce the number of embedding changes, since those tricks are barely effective for payloads near the maximum.

Nonetheless, we advise caution in use of feature restoration: an important limitation, which fell outside the scope of this paper, is that we assume knowledge of the opponent's steganalysis feature set. Were the opponent to use some alternative set of features, the restoration process could introduce additional distortion in the other feature set and enable better detection.

In some respects, distortion minimization appears preferable to feature restoration: if properly implemented, it can cope with very large feature sets (see Ref. 23 for an extreme example) and it should be less prone to overtraining because it generally does not introduce *additional* distortion to the stego object. On the other hand, one could apply feature reduction as an add-on to an embedding which already includes distortion minimization.

In examining the feature restoration problem we have illustrated some lessons which are applicable more widely. First, that intractable optimization problems can often be attacked using randomized algorithms, and that greedy algorithms which test lots of potential changes at once are appropriate for this situation. Second, and more important, that the choice of distortion metric is crucial: if one fails to take account of the correlation between features then the restoration can be futile or detrimental. The Mahalanobis distance is the simplest metric that deals with correlation. It effectively whitens the cover feature distribution but requires some care to avoid instability; therefore, other distances could yet prove superior.

We note that a recent distortion-minimization steganography scheme, HUGO,[11] uses a distortion metric related to standardized Euclidean distance, which cannot account for dependence between features. This may prove to be a weakness and should be investigated further.

ACKNOWLEDGMENTS

Some of this work appeared in the first author's BA dissertation,[15] and was completed during a summer project funded by the EPSRC Vacation Bursary scheme. The second author is a Royal Society Research Fellow.

REFERENCES

[1] Farid, H. and Lyu, S., "Detecting hidden messages using higher-order statistics and support vector machines," in [*Proc. 5th Information Hiding Workshop*], *Springer LNCS* **2578**, 340–354 (2002).

[2] Goljan, M., Fridrich, J., and Holotyak, T., "New blind steganalysis and its implications," in [*Security, Steganography and Watermarking of Multimedia Contents VIII*], *Proc. SPIE* **6072**, 0101–0113 (2006).

[3] Shi, Y., Chen, C., and Chen, W., "A Markov process based approach to effective attacking JPEG steganography," in [*Proc. 8th Information Hiding Workshop*], *Springer LNCS* **4437**, 249–264 (2006).

[4] Pevný, T. and Fridrich, J., "Merging Markov and DCT features for multi-class JPEG steganalysis," in [*Security, Steganography and Watermarking of Multimedia Contents IX*], *Proc. SPIE* **6505**, 0301–0314 (2007).

[5] Kodovsky, J., Pevný, T., and Fridrich, J., "Modern steganalysis can detect YASS," in [*Media Forensics and Security XII*], *Proc. SPIE* **7541**, 0201–0211 (2010).

[6] Pevný, T., Bas, P., and Fridrich, J., "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security* **5**(2), 215–224 (2010).

[7] Crandall, R., "Some notes on steganography," (1998). Steganography Mailing List, available from `http://os.inf.tu-dresden.de/~westfeld/crandall.pdf`.

[8] Fridrich, J., Lisoněk, P., and Soukal, D., "On steganographic embedding efficiency," in [*Proc. 8th Information Hiding Workshop*], *Springer LNCS* **4437**, 282–296 (2006).

[9] Kim, Y., Duric, Z., and Richards, D., "Modified matrix encoding technique for minimal distortion steganography," in [*Proc. 8th Information Hiding Workshop*], *Springer LNCS* **4437**, 314–327 (2006).

[10] Filler, T., Judas, J., and Fridrich, J., "Minimizing embedding impact in steganography using trellis-coded quantization," in [*Media Forensics and Security XII*], *Proc. SPIE* **7541**, 0501–0514 (2010).

[11] Pevný, T., Filler, T., and Bas, P., "Using high-dimensional image models to perform highly undetectable steganography," in [*Proc. 12th Information Hiding Workshop*], *Springer LNCS* **6387**, 161–177 (2010).

[12] Kodovský, J. and Fridrich, J., "On completeness of feature spaces in blind steganalysis," in [*Proc. 10th ACM Workshop on Multimedia and Security*], 123–132 (2008).

[13] Mahalanobis, P., "On the generalised distance in statistics," in [*Proceedings National Institute of Science, India*], **2**(1), 49–55 (1936).

[14] Madzarov, G. and Gjorgjevikj, D., "Evaluation of distance measures for multi-class classification in binary SVM decision tree," in [*Artificial Intelligence and Soft Computing*], *Springer LNCS* **6113**, 437–444 (2010).

[15] Chonev, V., "Improved steganography via feature restoration," Oxford University Computing Laboratory Research Report CS-RR-10-12 (2010).

[16] Kirkpatrick, S., Gelatt, C., and Vecchi, M., "Optimization by simulated annealing," *Science* **220**(4598), 671–680 (1983).

[17] Wolsey, L., [*Integer Programming*], Wiley (1998).

[18] Coleman, T. and Li, Y., "A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables," *SIAM Journal on Optimization* **6**(4), 1040–1058 (1996).

[19] Ker, A., "Stability of the Mahalanobis Distance: A Technical Note," Oxford University Computing Laboratory Research Report CS-RR-10-20 (2010).

[20] Ker, A. and Lubenko, I., "Feature reduction and payload location with WAM steganalysis," in [*Media Forensics and Security XI*], *Proc. SPIE* **7254**, 0801–0811 (2009).

[21] "`BOSSBase`: Database from the BOSS contest." `http://boss.gipsa-lab.grenoble-inp.fr`, accessed 22 June, 2010.

[22] Welsh, D., "Randomised algorithms," *Discrete Applied Mathematics* **5**(1), 133–145 (1983).

[23] Filler, T. and Fridrich, J., "Steganography using Gibbs random fields," in [*Proc. 10th ACM Workshop on Multimedia and Security*], 199–212 (2010).

[24] Garey, M. and Johnson, D., [*Computers and Intractability: A Guide to the Theory of NP-Completeness*], W. H. Freeman & Co. (1979).

APPENDIX A. FEATURE RESTORATION IS NP-COMPLETE

"P" represents the class of decision problems (problems with yes/no answers) that can be solved in time polynomial in the size of their input. "NP" is the class of decision problems that have a "certificate" – a purported answer and succinct evidence that it is correct – which can be *verified* in time polynomial in the size of their input. NP contains all of P as well as many problems that are widely believed to be difficult, for example the travelling salesman problem (minimal Hamiltonian cycle) and the satisfiability problem from propositional logic. A decision problem is called NP-hard if every other problem in the class NP can be reduced to it with at most polynomial time overhead, and NP-complete if it is both in NP and NP-hard. This means that NP-complete problems are the "hardest" in the class NP: travelling salesman and satisfiability are examples of NP-complete problems. The same definitions can apply to optimization problems, which are transformed into decision problems by asking the question: for given k, does there exist a solution with optimal value at least as good as k? For a general introduction to complexity, and NP-completeness in particular, see Ref. 24.

If a problem has been proved NP-complete, it means that it almost certainly has no efficient solution in general: if it had a polynomial time solution, so would many famously-difficult problems. Let us take an easy version of the FR problem, where all possible pixel changes lead to additive feature changes, and there are no constraints on which pixel changes can be combined (unless there are very few possible combinations of changes, removing the constraints always makes the quadratic programming problem easier). If the distance metric is a positive definite quadratic form, recall that the FR problem can be described as minimization of $\boldsymbol{x}^T\Theta\boldsymbol{x} + \boldsymbol{k}^T\boldsymbol{x}$, where $\Theta \in \mathbb{R}^{n\times n}$ (symmetric non-negative definite) and $\boldsymbol{k} \in \mathbb{R}^n$ are determined by the distance metric, result of the possible changes, and the difference between stego and target features, as in (8). The minimization is over binary vectors $\boldsymbol{x} \in \{0,1\}^n$, which tells us what changes are selected.

We now show that, even though this is a favourable formalization of the FR problem, it is NP-complete.

Theorem 1. *The problem*

$$\text{minimize} \quad \boldsymbol{x}^T\Theta\boldsymbol{x} + \boldsymbol{k}^T\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{x} \in \{0,1\}^n, \tag{9}$$

for arbitrary nonnegative definite $\Theta \in \mathbb{R}^{n\times n}$ *and* $\boldsymbol{k} \in \mathbb{R}^n$*, is NP-complete.*

Proof. It is easy to see why a certificate to answer the question $\exists \boldsymbol{x} \in \{0,1\}^n.\, \boldsymbol{x}^T\Theta\boldsymbol{x} + \boldsymbol{k}^T\boldsymbol{x} \leq k$ can be checked in polynomial time: the certificate is just $\boldsymbol{x}$, and we can perform the multiplication to verify whether $\boldsymbol{x}^T\Theta\boldsymbol{x} + \boldsymbol{k}^T\boldsymbol{x} \leq k$ in cubic time or better. The challenge is to show that the problem is NP-hard.

Take any formula of propositional logic, of n free variables $v_1, \ldots, v_n$, in conjunctive normal form with exactly two literals per clause and m clauses:

$$\phi(v_1, \ldots, v_n) = (v'_{i_1} \vee v'_{j_1}) \wedge (v'_{i_2} \vee v'_{j_2}) \wedge \cdots \wedge (v'_{i_m} \vee v'_{j_m}) \tag{10}$$

where each v'_k is either v_k or $\neg v_k$, and $\{i_1, \ldots, i_m, j_1, \ldots, j_m\} \subseteq \{1, \ldots, n\}$. Convert ϕ to a function $F(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}$ by the following procedure: conjunction maps to addition, and

$$\begin{array}{rll} v_k \vee v_l & \text{maps to} & x_k + x_l - x_k x_l, \\ v_k \vee \neg v_l & \text{maps to} & 1 - x_l + x_k x_l, \\ \neg v_k \vee \neg v_l & \text{maps to} & 1 - x_k x_l. \end{array}$$

By construction, there exists $\boldsymbol{x} \in \{0,1\}^n$ with $F(\boldsymbol{x}) = k$ if and only if there is a truth assignment to $\{v_1, \ldots, v_n\}$ such that exactly k clauses of (10) are true. This is relevant because the maximum number of clauses which can simultaneously be satisfied by a conjunctive normal form, with exactly two literals per clause, is known as MAX-2-SAT and it is an NP-hard problem [24, LO5].

Note that F is a quadratic function, so can be written in the form

$$F(\boldsymbol{x}) = \boldsymbol{x}^T\Gamma\boldsymbol{x} + \boldsymbol{d}^T\boldsymbol{x} + e,$$

for some constant e. Γ will normally be indefinite, but we can regularise it as follows. Define the function

$$G(\boldsymbol{x}) = e - F(\boldsymbol{x}) + m\sum_{i=1}^{n}(x_i^2 - x_i) = \boldsymbol{x}^T(m\mathrm{I} - \Gamma)\boldsymbol{x} + (m\mathbf{1} - \boldsymbol{d})^T\boldsymbol{x}.$$

We now argue that $m\mathrm{I} - \Gamma$ is nonnegative definite. Let λ be any eigenvalue of Γ; it is known that $\lambda \leq \|\Gamma\|_1$, the 1-norm of Γ. But $\|\Gamma\|_1 \leq \sum_{i,j} |\Gamma_{ij}| \leq m$, by the construction of F. Hence all the eigenvalues of $m\mathrm{I} - \Gamma$ are nonnegative so G is in the form of (9). Furthermore, for all $x \in \{0, 1\}$, $x^2 - x = 0$, so

$$\min_{\boldsymbol{x} \in \{0,\,1\}^n} G(\boldsymbol{x}) = e - \max_{\boldsymbol{x} \in \{0,\,1\}^n} F(\boldsymbol{x}).$$

This means that any solver for (9) can solve MAX-2-SAT, and hence every other NP problem.

The reduction has at worst a polynomial time overhead, because construction of F is linear and guarantees $e \leq m$. That completes the proof. $\square$