MPSteg-color: data hiding through redundant basis decomposition

Giacomo Cancelli and Mauro Barni, *Senior Member, IEEE*

***Abstract*—The possibility of using redundant basis expansion to securely hide a message within a cover color image is explored by improving previous attempts in this sense in terms of security and payload. The stability and computational complexity problems of previous works are solved by introducing new selection and update rules working entirely in the integer domain, and by fully exploiting the availability of three color bands in such a way that all the available atoms in the three color bands are used to convey the stego-message. Image decomposition is randomized in several ways thus improving the stego-message undetectability, and making the hidden message undetectable by targeted steganalyzers explicitly developed to exploit the weaknesses of the MPSteg algorithm. The security of the new scheme is also evaluated by testing it against blind steganalyzers and compared to that of ± 1 embedding algorithm applied in the pixel domain.**

***Index Terms*—Steganography, Matching Pursuit, High Redundant Basis, Targeted steganalysis in the MP domain.**

EDICS Category: WAT-STEG

I. Introduction

The ultimate goal of steganography is that of hiding a message within an innocuous signal in such a way that the very presence of the hidden message remains secret. The opposite effort of determining the presence of a hidden message within a cover signal is carried out by steganalyzers. Common steganalyzers rely on a statistical analysis to understand whether a given signal contains hidden data or not, however, this analysis disregards the semantic content of the cover signal. For the above reason it may be argued that, from a steganographic point of view, it is preferable to embed the steganographic message at the highest possible semantic level, e.g. by modifying structural elements of the host signal like lines, edges or flat areas in the case of still images.

Following a similar need arising from image compression applications [1], a new class of image representation methods has been recently developed that rely on redundant bases decomposition. In practice a dictionary with a large number of elementary signals (called atoms) is built, trying to ensure that, for each image (or image block), a subset of few atoms exists that permits to represent the image efficiently. The main problems with redundant basis decomposition of images are the construction of the dictionary and, more importantly, the definition of an efficient procedure to select the best subset of atoms for each image. The most common approach to solve the latter problem, consists to resort to Matching Pursuit (MP) techniques, that use a greedy algorithm to select a subset of atoms capable of representing the to-be-decomposed image efficiently [2].

Previous attempts to exploit redundant basis expansion to design a secure data hiding scheme are reported in [2] and [3]. The scheme described in [2] suffers from several problems including computational complexity, and instability of image decomposition. To explain the reason for these difficulties, let us recall that when an image (or any other signal) has to be represented by using the elements of a redundant basis, several decompositions are possible. MP algorithms work by selecting an element of the basis at a time in a greedy fashion, with no guarantee of global optimality. In a steganographic scenario, the embedder usually first decompose the image by using an MP algorithm, then modify the decomposition coefficients to insert the stego-message and then go back into the pixel domain. Due to the presence of the stego-message, when the decoder applies again the MP algorithm it may select from the redundant basis a different set of elements - and in a different order - hence making it impossible for the decoder to correctly extract the hidden message. Even though the subset of elements of the basis (and their order) is fixed, a change to one coefficient of the decomposition usually results in a variation of all the coefficients of the decomposition when the MP is applied to the modified image. The scheme described in [3] solves the above problems, however, the solution proposed therein is still not satisfactory since it presents a number of security problems that are not adequately addressed. Among them the most relevant is the possible lack of security against targeted steganalyzers that exploit the knowledge of the embedding algorithm and domain. In addition, as it will be explained later, the availability of three color bands to embed the stego-message is not fully exploited, thus reducing its payload.

In this paper, we revisit the systems described in [2] and [3], by tackling the problems outlined above to develop a new MP algorithm (still referred to as MPSteg-color for sake of simplicity) that permits to take full advantage of the characteristics of the MP embedding domain without sacrificing security. The main features of the new scheme (some of which are inherited from the algorithm proposed in [3]) can be summarized as follows: i) a particular MP decomposition strategy and a suitably tailored data hiding rule that permits to solve the instability problems of MP decomposition are used; ii) several sources of randomization are included in the embedding algorithm to prevent targeted steganalyzers from

Manuscript received ?? ??, ????; revised ?? ??, ????.

The authors are with Università di Siena, Italy (e-mail: giacomo.cancelli@unisi.it; barni@dii.unisi.it).

detecting the presence of the hidden message; iii) a refinement decomposition step is applied to the three color bands to increase the stego-message payload.

As a further contribution with respect to previous works, the performance of the new scheme are thoroughly evaluated by testing the detectability of the hidden message against three classes of steganalyzers, namely: targeted steganalyzers explicitly designed to exploit the weaknesses of the MPSteg color algorithm, steganalyzers developed to detect messages embedding by applying the ± 1 embedding algorithm directly in the pixel domain and general purpose steganalyzers.

The rest of the paper is organized as follows. In Section II a brief introduction to MP image decomposition is given. In Section III the requirements set by the steganographic application scenario onto the MP algorithm are discussed. The new MPSteg-color algorithm is presented in Section IV. Section V reports the results of the experiments that we carried out to validate the proposed technique. The paper ends in Section VI with some conclusions and hints for future research.

II. Introduction to MP image decomposition

Given a vector space V, a high redundant basis is a set of elements of V whose number greatly exceeds the dimension of V. The main idea behind the use of redundant basis for signal representation is that for any given signal it is likely that we can find a small subset of elements within the basis which are enough to represent the signal up to a certain accuracy level. Indeed, the more elements are contained in the basis the more likely the representing set will be small. Of course, since the number of signals in the basis exceeds the size of the space the host signal belongs to, the elements of the basis will no longer be orthogonal as in standard signal decomposition. At the same time, the availability of many degrees of freedom in the design of the redundant basis permits to include signals with specific semantic meaning.

In the following, the elements of the redundant basis will be called atoms, and the redundant basis the dictionary. The dictionary is usually indicated as $\mathcal{D}$:

$$\mathcal{D} = \{g_k\}_{k \in 1,\dots,N}, \tag{1}$$

where g_k is the k-th atom. If $\mathcal{I}$ is a generic signal (hereafter an image), we can describe it as the sum of a subset of elements of $\mathcal{D}$:

$$\mathcal{I} = \sum_{k=1}^{N} c_k g_k, \tag{2}$$

where c_k is the specific weight of the k-th atom, and where as many c_k as possible are zero. There are no particular requirements concerning the dictionary: in fact, the main advantage of this approach is the complete freedom in designing $\mathcal{D}$ which can then be efficiently tailored to closely match signal structures. Due to the non-orthogonality of the atoms, the decomposition in equation (2) is not unique, hence one could ask which is the best possible way of decomposing $\mathcal{I}$. Several meanings can be given to the term *best decomposition*. In compression applications, for instance, it is necessary that a suitable approximation in terms of human perceptible distortion of the image $\mathcal{I}$ is obtained. In this case, it is convenient to restate the decomposition problem as follows. Let $\gamma = \{\gamma_1, \gamma_2 \dots \gamma_N\}$ be a decomposition path, with γ_k indicating the index of the k-th atom of the decomposition. Let also define the residual signal $\mathcal{R}^n$ as the difference between the original image $\mathcal{I}$ and the approximation obtained by considering only n atoms of the dictionary. We have:

$$\mathcal{I}^n = \sum_{k=1}^{n} c_k g_{\gamma_k}, \tag{3}$$

$$\mathcal{R}^n = \mathcal{I} - \mathcal{I}^n, \tag{4}$$

where γ_k ties the atom identifier to the k-th position of the decomposition sum.

Given the above definitions, the best approximation problem can be restated as follows:

$$\min_{\gamma, c_k : \|\mathcal{R}^n\|^2 \leq \varepsilon} n \tag{5}$$

where ε is suitable approximation error. Unfortunately, the above minimization is an NP-hard problem, due to the non-orthogonality of the dictionary [4]. Matching Pursuit is a greedy method that, by looking for a suboptimal solution, permits to overtake the above NP problem with a polynomial complexity algorithm [4], by looking for a step by step minimization of the current residual $\mathcal{R}^k$. While MP finds the best solution at each step, it generally does not find the global optimum.

In the following, we will find convenient to rephrase MP as a two-step algorithm. The first step is defined through a selection function that, given the residual $\mathcal{R}^{k-1}$, selects the appropriate element of $\mathcal{D}$ and its weight:

$$[c_k, g_{\gamma_k}] = \mathcal{S}(\mathcal{R}^{k-1}, \mathcal{D}), \tag{6}$$

where $\mathcal{S}(\cdot)$ is a particular selection operator. At the second step, the residual is updated

$$\mathcal{R}^k = \mathcal{U}(\mathcal{R}^{k-1}, c_k, g_{\gamma_k}). \tag{7}$$

As it can be seen, for a complete definition of the MP framework several specifications must be given including the definition of the dictionary, the selection and the update rules. To do so, we must first investigate the requirements set by the particular framework in which we will apply the MP algorithm, i.e. image steganography.

III. Embedding a message in the MP domain

Given the representation formula

$$\mathcal{I} = \sum_{k=1}^{n} c_k \cdot g_{\gamma_k} + \mathcal{R}^n, \tag{8}$$

there are different ways of embedding a message within $\mathcal{I}$. In [5], for instance, the stego message is hidden in the particular decomposition path used to represent the image, whereas in [2] and [3], the message is hidden by modifying the decomposition coefficients c_k. In this paper, we adopt the latter approach, due to the difficulties of applying the former strategy in a blind detection framework (indeed the scheme

described in [5] requires non-blind detection). However, this strategy requires several problems to be addressed.

First of all, it is necessary that the transition from the pixel domain to the MP domain and then back to the pixel domain does not introduce approximation errors that could prevent the correct decoding of the stego-message. The easiest way of achieving this result consists in requiring that all the operations are performed in integer arithmetic with no need to quantize the stego image when the transformation from the MP to the pixel domain is performed.

The second requirement stems from the very goal of all our work, that is to embed the stego-message at as high semantic level as possible, hence the dictionary are as semantically meaningful as possible.

Third and the most fundamental requirement, regards the stability of the MP decomposition. As described in the introduction, MP instability has two different facets:

- Decomposition path instability: this source of instability is due to the fact that the insertion of the message may change the order in which the atoms are chosen by the MP algorithm. As a matter of fact, if this is the case, the decoder will fail to read the hidden message correctly (note that in image compression, where the image is reconstructed from a list of weighed atoms, the fact that a successive decomposition generates a different list of atoms is not a problem).
- Coefficient instability: the second source of instability derives from the non-orthogonality of the dictionary: if we modify one single coefficient c_{k^*}, reconstruct the modified image and apply the MP algorithm again, even if we do not change the order in which the atoms are selected, it may well be the case that all the coefficients will have different values. Even worse, there is no guarantee that the coefficient of the k^*-th atom will be equal to the value we set it to. It is easy to show that this is the case, for example, if the selection and update rules are based on the classical projection operator.

As the last observation, we note that, even though MP decreases the decomposition problem to polynomial complexity, the computational burden may still be prohibitive, especially if MP is applied to large image blocks. For this reason we decided to apply MP to small non-overlapping blocks rather than to consider the whole image. Note, however, that in principle, the subsequent discussion can be indifferently applied to the whole host image or to subparts of it.

In the next two sections we describe how the above constraints are satisfied by MPSteg color. We first describe the dictionary, then we introduce new selection and update rules explicitly designed to avoid coefficient instability[1].

A. Dictionary

There are several ways of building the dictionary. Discrete- or real-valued atoms can be used and atoms can be generated manually or by means of a generating function. In classical MP techniques, applied to still images [1], the dictionary is built by starting from a small set of generating functions that generate real-valued atoms. A problem with real-valued atoms is that when the modified coefficients are used to reconstruct the image in the pixel domain, non-integer values may be produced, thus resulting in a quantization error when the grey levels are expressed in the standard 8-bit format. This is a problem in steganographic applications where the hidden message is so weak that the quantization error may prevent its correct decoding. For this reason, and to prevent instability problems (see Theorem 1), we decided to work with binary-valued atoms for which only the 0 and 1 values are allowed.

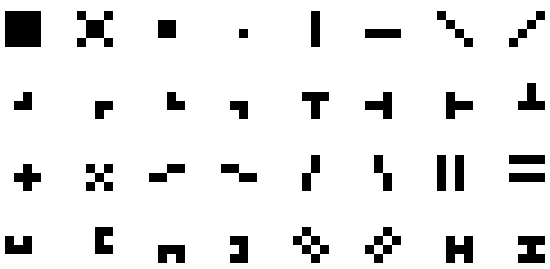


Fig. 1. A subset of the atoms the dictionary consists of.

The most important property of the dictionary is that it should be able to describe each type of image with a linear combination of few atoms. To simplify the construction of the dictionary and to keep the computational burden of the MP decomposition low, we decided to work on a block by block basis, applying the MP algorithm to 4×4 blocks. At this level, each block may be seen as the composition of few fundamental geometric structures like flat regions, lines, edges and corners. Specifically, we designed the dictionary by considering elements which describe uniform areas, contours, lines, edges, C-junctions, H-junctions, L-junctions, T-junctions and X-junctions. In Figure 1 the basic (non-shifted) atoms forming the dictionary are shown. The complete dictionary is built by considering the atoms reported in Figure 1 and their cropped 4×4 version when the center of the zero-padding atom - at coordinate (2,2) - is shifted around the 4×4 crop window. The whole dictionary is formed by 324 distinct atoms.

B. MP selection and update rules

In order to avoid that quantization errors prevent the correct decoding of the hidden message, let us observe that the stego-message will be embedded in the MP domain by modifying the coefficients c_k in equation (3), however, after embedding, the modified image must be brought back into the pixel domain. If we want to avoid the introduction of quantization errors it is necessary that the reconstructed image belongs to the *Image class*. The *Image class* is defined by the following property:

Property 1: Let $\mathcal{I}$ be a generic gray image[2] in the pixel domain and let $n \times m$ be its size. Let $\mathcal{I}(x, y)$ be the value of the image $\mathcal{I}$ at x-row and y-column. We say that $\mathcal{I}$ belongs

[1]The solutions reported below are inherited from the system described in [3]. We included their description in the current paper for the sake of clarity.

[2]It is possible to extend this definition to RGB images by considering each color band as a gray image.

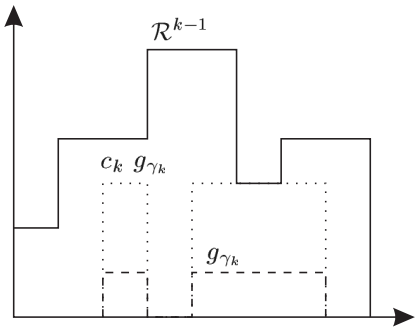


Fig. 2. The Selection Rule.

to the *Image class* if:

$$\forall x \in 1,\dots,n, \forall y \in 1,\dots,m$$
$$0 \leq \mathcal{I}(x,y) \leq 255 \quad \text{and} \quad \mathcal{I}(x,y) \in \mathbb{N},$$

the value 255 is used by considering an 8 bit color depth for each color band.

The necessity of ensuring that at each step the approximated image and the residual belong to the *Image class* already suggested us to consider binary-valued atoms, now we also impose that atom coefficients take non-negative integer values. In this way, we ensure that the reconstructed image belongs to the *Image class*[3]

Coefficient instability is more difficult to deal with, especially when coupled with the requirement that the decomposition path includes atoms matching the structural content of the image. MPSteg-color achieves the above result by defining the selection rule as follows. At each decomposition step k let

$$\mathcal{S}(\mathcal{R}^{k-1}, \mathcal{D}) = [c_k^*, g_{\gamma_k^*}] \tag{9}$$

with

$$\gamma_k^* = \arg \min_{\gamma_k \in \{1,2,\dots,|\mathcal{D}|\}} \sum_{i,j} \|\mathcal{R}_{\gamma_k}^k(i,j)\|^2 \tag{10}$$

and

$$\mathcal{R}_{\gamma_k}^k = \mathcal{R}^{k-1} - c_k^* g_{\gamma_k}, \tag{11}$$

where the notation $\mathcal{R}_{\gamma_k}^k(i,j)$ makes explicit the dependence of the residual at the k-th step on the selected atom, and where c_k^* is computed as follows:

$$c_k^* = \max\{c \geq 0 : \mathcal{R}^{k-1} - c g_{\gamma_k} \geq 0 \quad \text{for every pixel}\}. \tag{12}$$

An illustration of the behavior of the selection rule is given in Figure 2, where the choice of c_k is shown in the one-dimensional case. By starting from the residual R^{k-1} (solid line) and the selected atom g_{γ_k} (dashed), the weight c_k is calculated as the maximum integer for which $c_k g_{\gamma_k}$ is lower than or equal to R^{k-1} (the dotted line in the figure). Note that given that the atoms take only 0 or 1 values, at each step the inclusion of a new term in the MP decomposition permits to set to zero at least one pixel of the residual. Note also that the partial residual $\mathcal{R}^k$ continues to stay in the *Image class*.

[3]Actually we must also ensure that no underflow or overflow errors occur. We will consider this problem later on in section IV.

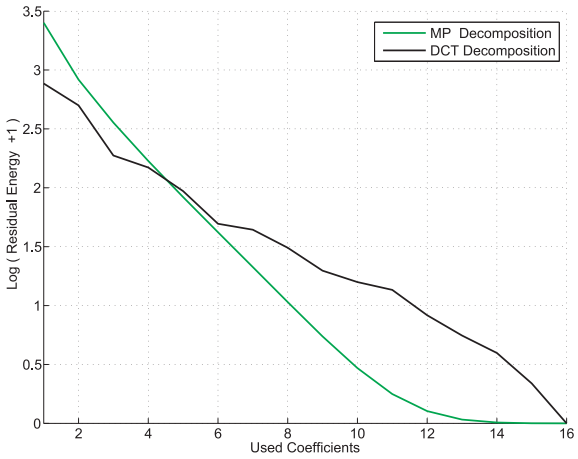


Fig. 3. Comparison between the compaction property of the DCT and MP domains.

We must now determine whether the selection rule described above is able to avoid the instability of MP coefficients. This is indeed the case, if we assume that the decomposition path is fixed and that only non-zero coefficients are selected for embedding, as it is shown by the following theorem.

Theorem 1: Let $\mathcal{I} = \mathcal{R}^0$ be an image and let $\vec{g}_\gamma = (g_{\gamma_1}, \dots, g_{\gamma_n})$ be a decomposition path. We suppose that the atoms are binary valued, i.e. they take only values 0 or 1. Assume that the MP decomposition coefficients are computed iteratively by means of the following operations:

$$c_k = \max\{c \geq 0 : \mathcal{R}^{k-1} - c g_{\gamma_k} \geq 0 \quad \text{for every pixel}\} \tag{13}$$
$$\mathcal{R}^k = \mathcal{R}^{k-1} - c_k g_{\gamma_k}, \tag{14}$$

and let $\vec{c} = (c_1, c_2, \dots, c_n)$ be the coefficient vector built after n iterations. Let c_k be an element of $\vec{c}$ with $c_k \neq 0$, and let $\vec{c}\,'$ be a modified version of $\vec{c}$ where c_k has been replaced by c_k'. If we apply the MP decomposition to the modified image

$$\mathcal{I}' = \sum_{i=1, i\neq k}^{n} c_i \cdot g_{\gamma_i} + c_k' g_{\gamma_k} + \mathcal{R}^n \tag{15}$$

by using the decomposition path $\vec{g}_\gamma$, we re-obtain exactly the same vector $\vec{c}\,'$ and the same residual $\mathcal{R}^n$.

The proof of Theorem 1 is given in Appendix A. Theorem 1 can be applied recursively to deal with the case in which more than one coefficient in $\vec{c}$ is changed. In the next section we show how the stability result stated in Theorem 1 was used to build the MPSteg-color algorithm.

C. A closer look at the new MP domain

One may wonder whether the particular dictionary, selection and update rules we used, which are the result of the requirements set in the previous section, maintain the compaction properties of high-redundant basis. This is indeed the case as it is witnessed by Figure 3 and exemplified in Figure 4. Specifically, in Figure 3 the reconstruction error is plotted (in log scale) as a function of the number of basis elements considered for the reconstruction (the results have

been obtained by averaging the plots relative to 25 images), as it can be seen when very few coefficients are used the DCT decomposition performs better. This is due to the decision we made to design the update rule in such a way that the residual image is always positive (while the DCT coefficients are chosen in such a way to minimize the error energy). However, when the number of basis elements increases the MP capacity of fully describing the image with a lower number of elements is evident. Indeed in the DCT case all the 16 coefficients of the orthogonal basis are needed to bring the reconstruction error to zero, while in the MP case only 9.63 atoms are needed (on the average).

From a different perspective, the higher semantic level MP operates at is exemplified in Figure 4. The original image (Figure 4(a)) is first decomposed by applying a 4×4 DCT and reconstructed by using only the DC and the first AC coefficient, yielding the result depicted in Figure 4(b). The same approach is applied in Figure 4(c) where the image is generated by using only the first 2 atoms of the MP decomposition. Though the reconstruction error is larger in the MP case (in accordance with the plot of Figure 3), the perceived quality of the image obtained through MP decomposition is better than that obtained with DCT, since the selected atoms permit to better represent the geometric structures contained in the image.

IV. MPSteg-color

In this section we give a detailed description of the MPSteg-color algorithm. We first review the main structure of the algorithm, then we describe how we modified such an algorithm to achieve security against targeted steganalyzers and to increase the stego-message payload.

Theorem 1 ensures that by using the selection rule described in equations (9) through (12), it is possible to correctly write and read a message hidden in the MP coefficients if the decomposition path $\vec{g}_\gamma$ is known. In order to cope with decomposition path instability, we exploit the availability of three color bands. To explain how, let us introduce the following notation:

$$\mathcal{I} = \begin{pmatrix} \mathcal{I}_r \\ \mathcal{I}_g \\ \mathcal{I}_b \end{pmatrix}$$

where $\mathcal{I}_r$, $\mathcal{I}_g$ and $\mathcal{I}_b$ are the RGB bands of a traditional color image.

MPSteg-color works on a non-overlapping, 4×4 block-wise partition of the original image, however, for simplicity we continue to refer to image decomposition instead of block decomposition, the use of blocks, in fact, is only an implementation detail, not a conceptual strategy.

The main idea behind MPSteg-color is to use the correlation of the three color bands to stabilize the decomposition path. Specifically the decomposition path is calculated on a color band and then used to decompose the other two (the validity of such an argument will be tested in Section V-B1) bands. Due to the high correlation between color bands, we argue that the structural elements found in a band will also be present in the other two. Suppose, for instance, that the decomposition path is computed on the $\mathcal{I}_r$ band, we decompose the original image as follows

$$\mathcal{I} = \begin{pmatrix} \sum_{k=1}^{n} c_{r,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}_r^n \\ \sum_{k=1}^{n} c_{g,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}_g^n \\ \sum_{k=1}^{n} c_{b,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}_b^n \end{pmatrix} \tag{16}$$

where $g_{\gamma_{r,k}}$ are the atoms selected on the red band, $c_{r,k}$,$c_{g,k}$ and $c_{b,k}$ are the atom weights of each band and $\mathcal{R}_r^n$,$\mathcal{R}_g^n$ and $\mathcal{R}_b^n$ are the partial residuals. By using eq. (16) we do not obtain the optimum decomposition of $\mathcal{I}$ for the green and blue bands, but this decomposition has a good property: if the red band is not modified then the decoder may apply the selection function $\mathcal{S}(\cdot)$ to the red band and use it to retrieve the decomposition path used by the embedder to hide the message in the other two bands.

By assuming, for instance, that the decomposition path is computed on the red band, then MPSteg-color can embed the stego-message by operating on the vector with the decomposition weights of the green and blue bands, i.e. the vector

$$\vec{c}_{gb} = (c_{g,1}, c_{b,1}, \ldots, c_{g,n}, c_{b,n}). \tag{17}$$

According to Theorem 1, we know that the stego message can be correctly embedded by changing the coefficients of the MP decomposition vector $\vec{c}_{gb}$, however, for this result to hold it is necessary that only non-zero coefficients are modified. In fact, given that the decomposition path is computed on one band and the message embedded in the other two, it may be the case that the coefficients of some atoms of the decomposition path are zero, i.e. the vector $\vec{c}_{gb}$ may contain some null coefficients. This issue will be considered in the next subsection, where the embedding rule used by MPSteg-color is described.

A. Embedding Rule

We now describe the embedding rule used to embed the stego-message within $\vec{c}_{gb}$. Given that the coefficients of $\vec{c}_{gb}$ are non-negative integers, we can apply any method that is usually applied to embed a message in the pixel domain. However, we must consider that the embedder cannot modify zero coefficients (due to Theorem 1 assumptions), but in principle it could set to zero some non-zero coefficients. If this is the case a de-synchronization would be introduced between the embedder and the decoder since the decoder will not know which coefficients have been used to convey the stego-message. In the steganographic literature this is known as the channel selection problem, for which an elegant solution exists, namely the Wet Paper Code strategy introduced by Fridrich *et al.* in [6]. However, the aim of this work is to analyze the capability of the MP domain as a cover domain, hence will not consider any procedure to redirect the embedding changes of the basic MPSteg algorithm[4]. In

[4]Similarly we will not consider matrix embedding [7] as well, since it can be used to boos the performance of any steganographic scheme, regardless of the embedding domain.

(a) Original (b) 2 DCT coefficients (c) 2 atoms

Fig. 4. Compaction capability: original gray-scale image (a), reconstructed image by using the first 2 DCT coefficients in a zig-zag ordering for each 4×4 block (b) and by using 2 atoms for each 4×4 block (c).

fact, the same procedures could be applied to pixel domain methods, and are not related to the particular domain in which the message is embedded.

For this reason, we adopted the standard ± 1 technique to embed the message in the non-null weights. The ± 1 embedding strategy is described by the following rule:

$$p_s = \begin{cases} p_c + 1, & \text{if } b \neq \text{LSB}(p_c) \text{ and } \big(\kappa > 0 \text{ or } p_c = 0\big) \\ p_c - 1, & \text{if } b \neq \text{LSB}(p_c) \text{ and } \big(\kappa < 0 \text{ or } p_c = 255\big) \\ p_c, & \text{if } b = \text{LSB}(p_c) \end{cases} \tag{18}$$

where p_s (resp. p_c) denotes a pixel value in the stego image (resp. cover image), b is the message bit to be hidden, and κ is an i.i.d. random variable uniformly distributed in $\{-1, +1\}$[5]. In order to avoid the channel selection problem, we add 2 to all the coefficients for which equation (18) yields a null value. By indicating with $\vec{c}^{\,w}_{gb} = (c^w_{g,1}, c^w_{b,1}, \ldots, c^w_{g,n}, c^w_{b,n})$ the marked coefficient vector, then we build the stego image $\mathcal{I}^s$:

$$\mathcal{I}^s = \begin{pmatrix} \sum_{k=1}^{n} c_{r,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}^n_r \\ \sum_{k=1}^{n} c^s_{g,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}^n_g \\ \sum_{k=1}^{n} c^s_{b,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}^n_b \end{pmatrix}. \tag{19}$$

While the application of equation (18) to MP coefficients guarantees that the modified coefficients lie in the [0,255] interval, it is possible that some pixels of the reconstructed image exceed the 255 limit. If this happens, the coefficients larger than 2 are decreased by 2 until the overflow error disappears. In this way the embedding distortion is slightly augmented, however, such an effect is completely negligible since overflow errors are extremely rare.

[5]Note that this strategy may affect bit-planes other than the LSB plane. For example, if the secret bit is a "0", and the original 8-bit pixel value is 01111111, then incrementing this value results in 10000000.

B. Improving security

In [3] the security of the above scheme against general purpose steganalyzers has been shown, however, security against targeted steganalysis may be a problem. First of all, if the dictionary is assumed to be known a steganalyzer may look for specific artifacts introduced by MPSteg-color directly in the MP domain. Secondarily, even if the dictionary is kept secret, the particular nature of atoms and the application of the MP algorithm at a block level, may introduce blocking artifacts that could be used by a targeted steganalyzer to detect the presence of a stego-message. As it will be shown in section V-C this is indeed the case, hence some countermeasures need to be taken.

First of all we decided to not use the first decomposition coefficient as support of the secret message. Usually such a coefficient is able to describe most of the image energy compared to the remaining atoms. For this reason, any modification to the first atom is likely to introduce significant blocking artifacts, hence we decide to keep such an atom unchanged.

The second and more important countermeasure we took, is randomization of the embedding process. Randomization is applied at two different levels. At the first level randomization affects the image decomposition into blocks. By following an approach similar to that proposed by Solanki *et al.* in [8] the image is partitioned into disjoint and contiguous windows with size 5×5 or 6×6, and MP decomposition is applied to 4×4 blocks randomly chosen within the larger 5×5 (or 6×6) windows[6]. By doing so we reduce and randomize the blocking artifacts introduced by MPSteg-color that will be more difficult to detect. In addition, even by knowing the MP dictionary, the MP domain used by a possible adversary will be spatially desynchronized with respect to the one used by the embedder, thus making steganalysis in the MP domain more difficult. Of course a compromise between payload and security must be found here, given that the larger the window size the better the security at the expense of payload (given that the number

[6]Randomization is achieved by changing the offset of the 4x4 window within the larger 5x5 or 6x6 window.

of pixels not touched by MPSteg-color will increase).

The second randomization level regards the choice of the reference color band that is used to calculate the MP decomposition path. Specifically, a secret key is used as a seed for a random number generator that decides on a block by block basis which color band is used to calculate the decomposition path. The MP decomposition is applied to the chosen band, while the secret message is embedded within the other bands.

As it will be seen in section V, through randomization, especially block position randomization, it is possible to resist to security attacks brought by targeted steganalysis.

C. Increasing the payload

An undesirable effect of block position randomization is that the payload is (slightly) decreased, all the more that the capacity[7] of MP domain is intrinsically lower than that of the spatial domain (see [2], [3]). A possible way to improve (slightly) the payload of messages hidden by MPSteg-color stems from the observation that though the color bands are highly correlated, the decomposition path calculated on one of them in general is not able to lead to a zero residual on the other two bands. For some of the atoms selected in the reference band, in fact, a null coefficient is obtained in the other bands, thus diminishing the number of coefficients available for embedding. For this reason, after that the decomposition path computed on the reference band is applied to the other two bands, the residual of one of the these two bands is further decomposed to provide an additional list of atoms that are used on the remaining band to provide additional coefficients to embed some more bits. In the rest of the paper we will refer to this second decomposition step as the *decomposition refinement step*. The actual payload increase obtained thanks to the decomposition refinement step will be evaluated experimentally in section V-B.

V. Experimental Results

In this section we report experimental results that demonstrate the security of the new version of MPSteg-color and validate the main assumptions behind it. First of all in Section V-A the image database used for the experiments is described. Afterwards, in Section V-B we take a closer look at the MP domain to support the hypothesis that the decomposition path calculated in one color band can be used with little loss for the other bands. We also evaluate the gain in terms of payload that is brought by the decomposition refinement step.

After that, in Section V-C, we carefully analyze the security of the proposed technique, with particular attention to the effectiveness of partition randomization as a countermeasure to targeted steganalysis. For this reason the undetectability of the stego-message is tested first again two targeted steganalyzers explicitly developed to detect MPSteg-color messages, then against general purpose steganalyzers.

[7]We are using the term capacity in a loose sense, without any reference to the corresponding information theoretic concept.

A. Image Database

For the experimental validation we used a database of 2564 raw color images of 512×512 size.

Images are the cropped version of the original ones which are taken in a RAW format from several kinds of common cameras. The images in the database show a wide range of scenarios including countryside, houses, people, faces, man-made objects, etc.

B. Effectiveness of the proposed MP decomposition

We first validate the conjecture that, due to the correlation between RGB color bands, computing the decomposition path on one band and using it on the other two does not impair the capability of the MP algorithm to extract the most important features of image blocks. Moreover we give a measure of the payload allowed by the MP domain and the payload gain allowed by the decomposition refinement step. On one side this is a good result showing a high degree of correlation, on the other side it shows that the decomposition path calculated on one band is capable of fully describe the content of the other bands, thus justifying the resort to a decomposition refinement step.

1) Interband correlation of decomposition path: In the proposed technique we argue that the three bands are highly correlated and for each block we randomly select one color band to build a decomposition path that will be used to decompose the other two bands. To experimentally validate the above conjecture, we decomposed a random color band until a null residual is obtained, then with the same decomposition path we decomposed one of the remaining bands. After this second decomposition, we usually obtain a non-null residual that will be null only if the decomposition path calculated on the first band fits the content of the second band. At this point we applied a matching pursuit decomposition to the non-null residual and we measured its length. By averaging the results obtained on all the images of the test database, we found that about 3.7 additional atoms are needed to decompose the second and the third band residuals that is about 40,80dB (while about 9.63 atoms were necessary for the reference band).

2) Effectiveness of the decomposition refinement step: The goal of the decomposition refinement step is to further decompose the residuals of the two remaining bands after that the decomposition path computed on the reference band is applied to them. In this way some extra non-zero coefficients are obtained thus contributing to increase the payload of MPSteg-color. Specifically, we found that the number of available coefficients for embedding is increased by 12.29% on average. In terms of payload this means that if we embed one bit per non-null coefficient then we are able to increase the size of the secret message by a 12.29% factor.

C. Security analysis

The most important requirement for any steganographic technique is undetectability. In this section we report the results that we obtained by applying four state-of-the-art

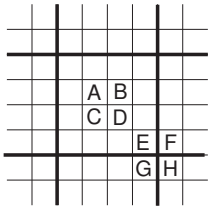


Fig. 5. For each block the numbers $Z' = |A + D - B - C|$ and $Z'' = |E + H - F - G|$ are computed.

steganalyzers to detect ± 1-Steg applied in the MP domain and the pixel domains. Before doing that, however, we test the effectiveness of block partition randomization to combat targeted steganalyzers. In the following, we briefly describe the steganalyzers we used by grouping them into two main sets.

The first set comprises target steganalyzers. It will be used to show the weakness of the previous versions of MPSteg-color, noticeably the one proposed in [3]. The second set of steganalyzers is composed by some of the most popular steganalyzers proposed until now.

All the steganalyzers are used as feature extractors, however, we decide to always use a simple linear classifier, namely the Fisher Linear Discriminant (FLD) that is described in Appendix B, to compare the goodness of each tool even though in the original version some of them are associated with an SVM classifier. We chose to compare all the steganographic algorithms by using a FLD classifier in order to highlight the capability of the various types of features to detect the presence of a hidden MPSteg message.

1) Targeted steganalyzers: The first targeted steganalyzer we used is built on the simple blocking artifacts detector (BD) described in [9]. This technique was originally developed for detecting JPEG block artifacts, however, we adapted it to detect the artifacts introduced by MPSteg-color and use them as a feature to detect the presence of the hidden message. The algorithm is very simple: we split the image into blocks whose size should be matched to that used by the MP algorithm. Regardless of the block partition strategy the steganalyzer assumes that blocks are located on a grid aligned with the top-left corner of the image. For each block we calculate Z and Z' as follows:

$$\begin{aligned} Z' &= |A + D - B - C| \\ Z'' &= |E + H - F - G| \end{aligned}$$

where A, B, C, D, E , F, G and H are taken as shown in Figure 5 in the case of 4×4 blocks, the extension to larger blocks being trivial. Next the normalized histograms vectors $h'(n)$ and $h''(n)$ are computed respectively for Z' and Z'' and the following feature is calculated:

$$\mathbf{f}_{BD} = \sum_{n=0}^{255} |h'(n) - h''(n)|.$$

The above procedure is repeated for the three color bands producing a three-dimensional feature vector that is given as input to the FLD classifier.

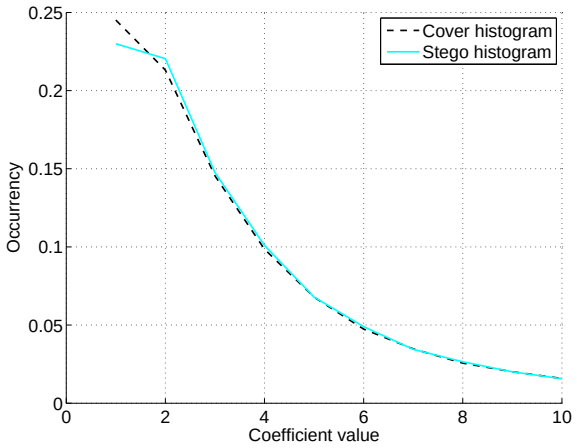


Fig. 6. Comparison between coefficients histogram of a cover image (dashed line) and a stego MPSteg-color image (solid line).

The second steganalyzer we developed relies on the knowledge of the histogram of MP coefficients. For this to be possible, we assume that the steganalyzer knows the MP dictionary but it does not know the reference band that is used to calculate the decomposition path (hence a random band is used as a reference by the steganalyzer). Figure 6 shows a typical histogram of a cover image and a stego MPSteg-color image. Due the embedding asymmetry applied to coefficients having value equal to 1 - that are either left unchanged or incremented by one - a flat step appears in the leftmost part of the histogram, while this effect does not appear in the cover image. By considering this effect, we propose to use the following feature:

$$\mathbf{f}_{MPHA} = h(2) - \frac{h(1) + h(3)}{2} \tag{20}$$

where h is the histogram function. In the sequel we will refer to this technique as MPHA.

2) State-of-art steganalyzers: The first steganalyzer of the second group is a rather new technique based on the artifacts introduced by ± 1-Steg in the image histogram [10]. It is possible to theoretically prove that a stego histogram is smoother than a cover one. By relying on this assumption, Zhang *et al.* extract a feature that estimates the histogram smoothness by checking peaks and valley heights.

The second algorithm we used in this set is called WAM steganalyzer [7]. It works in the wavelet domain and the extracted features are central moments that are calculated in the three detail bands of first order wavelet decomposition. This steganalyzer is a blind steganalyzer because it is not explicitly developed to detect any particular kind of messages.

The third steganalyzer we used was introduced by Ker in [11]. It builds on some considerations made in [12] about artifacts generated in the histogram domain by ± 1-Steg. In particular we used the concatenated features from the histogram analysis and the adjacency matrix analysis. We will refer to this steganalyzer by 2D-HCFC.

The fourth one is a steganalyzer that works with color images in the wavelet domain. It extracts wavelet features from the first three detail levels of decomposition by using common

statistics analysis as mean, variance, skewness and kurtosis. Moreover it extracts features from the linear prediction error between each decomposition detail band, however the whole analysis produced a 72-feature vector for each color image. This steganalyzer was proposed by Lyu and Farid in [13].

From the initial gray scale steganalyzers we implemented a color version by joining the 3 RGB band feature vectors in a unique vector with triple components. In this way we worked with three features for Zhang and 2D-HCFC steganalyzers and 81 features for WAM steganalyzer.

3) Steganalysis Results: For our experiments we embedded in each image a random message by using a secret unique key.

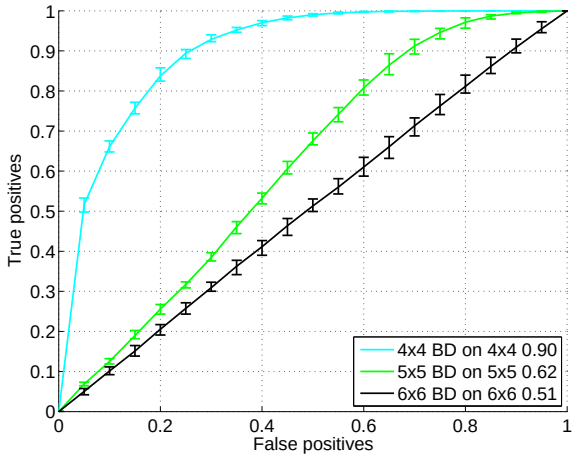
For MPSteg-color we used three window sizes in the experimental tests: 4×4, 5×5 and 6×6. The comparison between different methods was always made by using the maximum payload allowed by the techniques involved in the comparison, for instance when comparing MPSteg-color versions with different window sizes the payload imposed by the largest window is used [8].

The cover and stego images produced as described above were used to build a training and a test set, both containing 50% cover and 50% stego images. The size of the training set was equal to 20% of the 2564 images, the remaining 80% forming the test set. The training and the test sets were built randomly, however, to avoid any dependence of the results upon the specific training and test sets, the experiments were repeated 20 times, each time with a different training and test set. In this way we obtained 20 ROC curves that were vertically averaged to produce the final plots shown in the following. In the plots the minimum and maximum bound of the beam of ROC curves is shown. A brief introduction to the above experimental procedure (usually referred to as k-fold cross validation) can be found in [14].

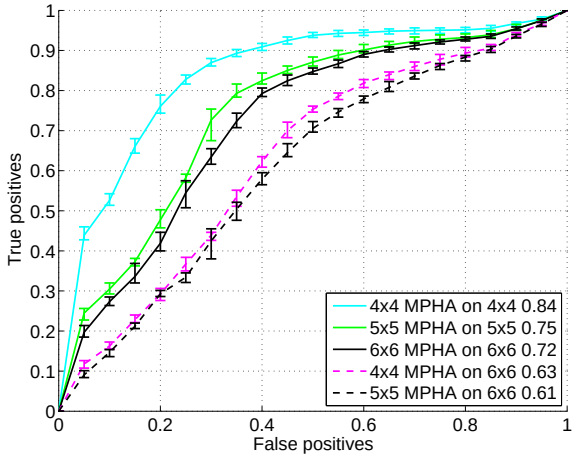
Figure 7 shows the performance of the two target steganalyzers described in Section V-C1. We considered several scenarios: in a first optimistic (for the steganalyzer) situation the steganalyzer knows the window size used by the embedder, though it does not know the particular randomization key used. In this case the steganalyzer simply picks a random 4x4 block out of the larger window, with a probability of guessing the right position equal to $1/4$ for 5×5 windows and $1/9$ for 6×6 windows. In the second case the steganalyzer is tuned on 4×4 blocks, but the actual partitioning window used by the embedder is larger.

The results shown in the Figure 7 indicate that both BD and MPHA steganalyzers performs best when no block randomization is used (MPSteg-color window size = 4×4) and the steganalyzers are tuned on the a 4×4 window size. At the same time, the security of MPSteg-color increases dramatically when block position is randomized, even when the steganalyzers are tuned to the exact window sizes. This is especially true for the BD steganalyzer whose performance are bad already for a 5×5 randomization window and are very close to a random guess for the 6×6 case. Since the MPHA performs slightly better, we also investigated its performance

[8]The payload is expressed in bit per pixel, by considering $512 \times 512 \times 3$ thus the number of the pixels.



(a) BD - MPSteg-color payload: 0.3159 bpp



(b) MPHA - MPSteg-color payload: 0.3159 bpp

Fig. 7. Comparison between BD (a) and MPHA (b) steganalyzers: the first window size reported in the figure indicates the window size used by the steganalyzer, the second one the size of the blocks used by the embedder to partition the image. The payload is 0.3159 bpp.

when it does not adopt the correct window size (a case that is closer to reality since in general the steganalyzer may not know the size of the randomization window used by the embedder). Specifically part (b) of Figure 7 reports the case of an MPHA steganalyzer tuned on a 5×5 or 6×6 window applied to stego-images containing messages embedded by using a 4×4 partition. As it can be seen the performance of the steganalyzer decrease significantly. Similar results are obtained for when the embedder uses a larger randomization window.

We now turn the attention to non-targeted steganalysis and to the comparison between MPSteg-color and ± 1-Steg applied in the pixel domain

Before presenting the ROC curves, it is instructive to consider the PSNR obtained by applying ± 1-steg in the pixel and in the MP domains. Such results are given in Table I for different MPSteg-color window sizes and different payloads. The average PSNR is obtained by taking the average on the linear quantities and then passing to the logarithmic scale. As expected, by considering that the atoms of the MP decomposition has a support larger than a single pixel, MPSteg-color

TABLE I
PSNR VALUES (DB) FOR DIFFERENT PAYLOAD AND SETTINGS.

Payload	MPSteg-color			±1-Steg
	6×6 window	5×5 window	4×4 window	
0.14 bpp	55.20	55.25	55.30	59.70
0.20 bpp	-	53.62	53.69	58.12
0.32 bpp	-	-	51.64	56.14

(a) Cover image (b) Stego image

Fig. 8. Perceptual invisibility of the stego-message. The stego (b) and the cover (a) images can not be distinguished (payload = 0.3158 bpp, 4×4 partition, 51.40dB).

results in a lower PSNR, hence suggesting that any advantage in terms of undetectability (if any) will be due to the better hiding properties of the MP domain.

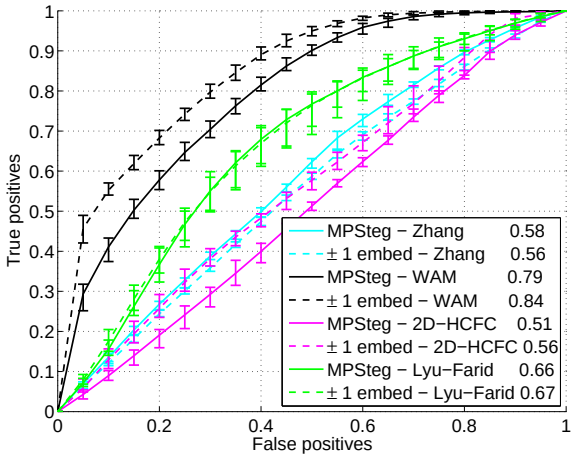
Despite the lower PSNR, the presence of the stego message can not be noticed perceptually as it is exemplified in Figure 8 where the stego-image (right) cannot be distinguished from the original one (left) even if the largest possible payload is used (0.3687bpp) for a PSNR of 51.22dB.

Figure 9 compares the detectability of MPSteg-color with that of ± 1-Steg, for three different window sizes (and different payloads). In the legend, the Area Under Curve (AUC) value is also given for each steganalyzer as an overall measure of classification accuracy.

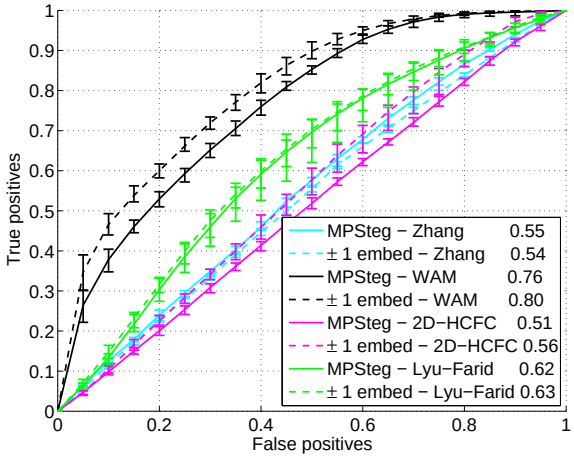
We can see that WAM is the only steganalyzer capable to distinguish the stego-images with a significant level of accuracy. Even in this case, though, the message embedded in the MP domain is less detectable than the one embedded in the pixel domain.

Slightly better results (from the steganalyzer point of view) are obtained for a 4×4 window (larger payload), however, the general behavior of the various algorithms does not change.

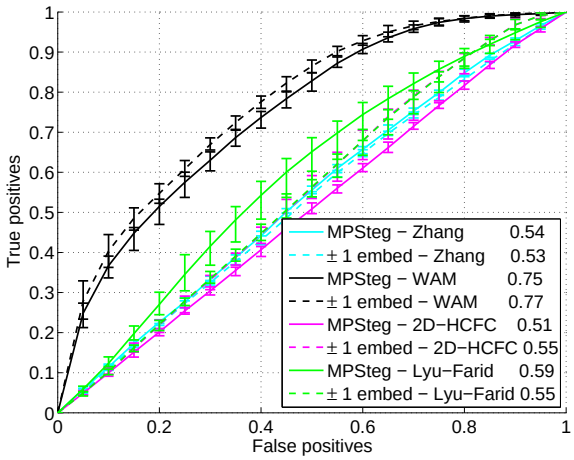
In order to evaluate the dependence of MPSteg-color detectability on the size of the randomization window, the ROC curves obtained for different sizes are plotted altogether in Figure 10. In this case we pay our attention to a specific steganalyzer, and we use the maximum admissible payload for all the used windows (i.e. those attainable with the 6×6 windows) that is 0.1391 bpp. The Zhang, WAM, 2D-HCFC and Lyu-Farid steganalyzers are respectively shows in Figures 10(a), 10(b), 10(c) and 10(d). We see in Figure 10(c) that 2D-HCFC steganalyzer is not able to detect MPSteg-color. The same effect can be seen in Figure 10(a). Instead, the



(a) MPSteg-color with window 4×4 - payload: 0.3159 bpp



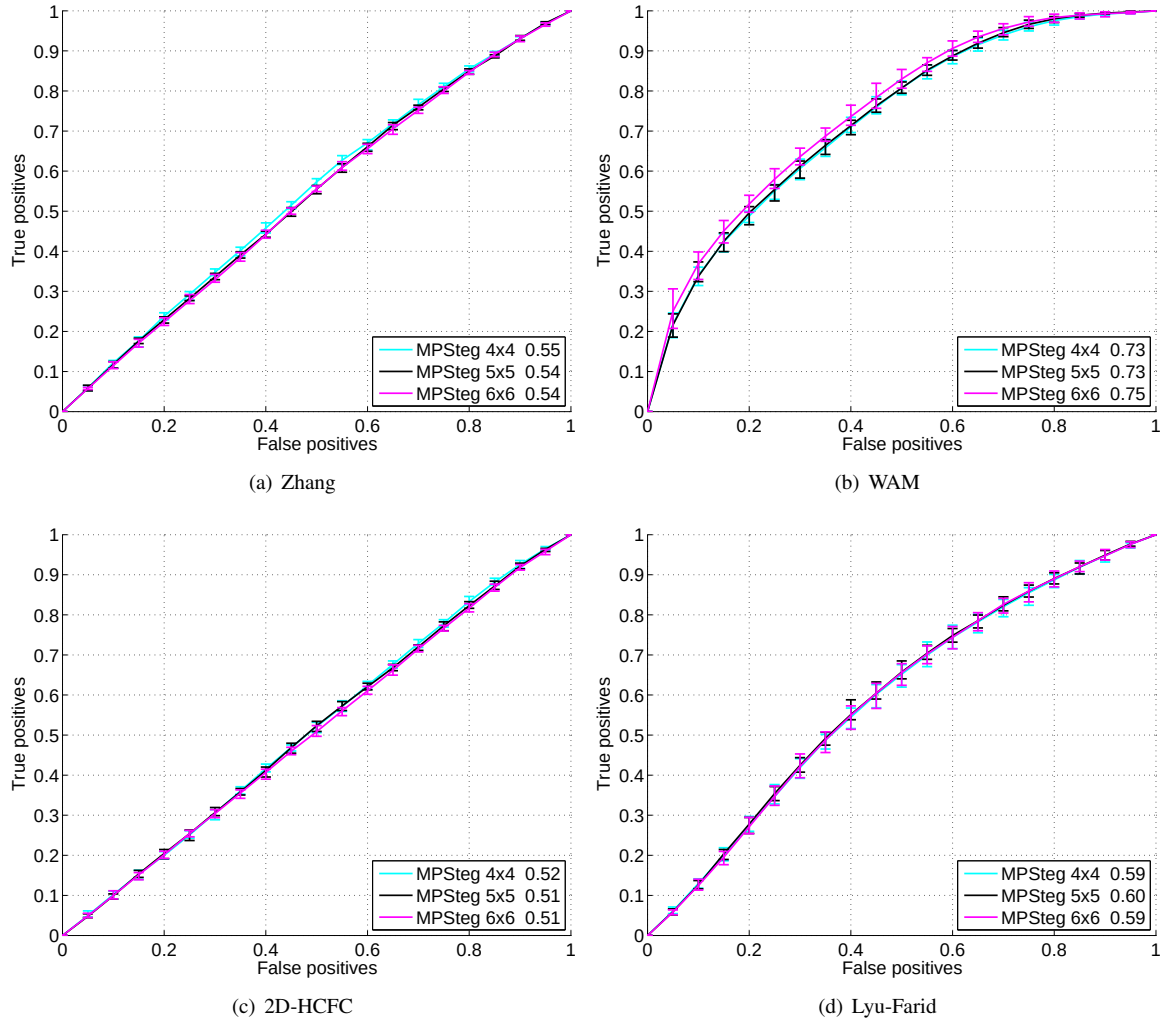
(b) MPSteg-color with window 5×5 - payload: 0.2002 bpp



(c) MPSteg-color with window 6×6 - payload: 0.1391 bpp

Fig. 9. Comparison between MPSteg-color (solid line) and ± 1-Steg (dashed line) with 4 different steganalyzers.

performance of Lyu-Farid's and WAM steganalyzers do not depend on the size of the partitioning window. A possible explanation for this behavior is that for the 6×6 case we are using the maximum admissible payload, hence approximately half of the MP coefficients are changed, while this is not the case with the 4×4 window. In addition, the additional



(a) Zhang (b) WAM

(c) 2D-HCFC (d) Lyu-Farid

Fig. 10. MPSteg-color detection performance on 4 different steganalyzers.

randomization allowed by the 6×6 window is a way to improve the security against targeted steganalyzers - as it is shown in Figure 7 - explicitly designed to detected a message embedded in the MP domain, the same advantage is not expected for other steganalyzers.

4) Computational Complexity: Although a lot of tricks are used to reduce the execution time, the MP exhaustive research to define the decomposition path at each step is really onerous and it is the bottleneck of the whole system. We develop the prototype of our scheme in MATLAB and we use a c-mex function in the kernel of exhaustive research in order to reduce as much as possible the computational time. Table II shows the execution time at the embedding phases (decomposition step, message embedding and image reconstruction) when the MATLAB code is execute on Intel Xeon at 3.00GHz. Even though the source code could be improved and a different language could be chosen, the decomposition step - that is used to the receiver side too - is the most critical part of the proposed steganography.

TABLE II
AVERAGE EXECUTION TIME OF EMBEDDING PHASES FOR IMAGES 512×512 OF SIZE, WINDOW 4×4 AND FULL PAYLOAD (0.32 BPP).

Decomposition	Embedding	Reconstruction
13830	14.78	2.5

VI. CONCLUSION

A new algorithm for embedding a stego message into colors images represented by means of high redundant basis decomposition has been presented. The problems of previous schemes proposed in this sense have been solved, with particular attention to security against targeted steganalyzers. Indeed, we have shown that without proper countermeasures, messages hidden by means of previous steganographic algorithms working in the MP domain are easily detectable.

In addition to preventing the above security pitfall, the new scheme proposed in this paper slightly increases the admissible payload, by adding a decomposition refinement step.

The security of the new scheme has been extensively tested against both targeted and general purpose steganalyzers, showing the validity of the proposed approach. In particular, the good hiding properties of the MP domain are demonstrated by comparing the undetectability of a ± 1-steg message embedded

in the pixel with that of a ± 1-steg message embedded in the MP domain, with the latter being less detectable than the former despite a higher embedding distortion. A few significant improvements of the proposed scheme are possible, either to augment the payload or diminish the detectability. Specifically, the wet paper coding approach may be applied to remove the constraint that message embedding cannot produce zero coefficients, and matrix embedding can be applied to decrease the embedding distortion.

ACKNOWLEDGMENTS

The authors would like to thank J. Fridrich and M. Goljan for providing the source code of the WAM steganalyzer.

The work described in this paper has been partially supported by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

REFERENCES

[1] P. Vandergheynst and P. Frossard, "Image coding using redundant dictionaries," *Document And Image Compression*, 2006.

[2] G. Cancelli, M. Barni, and G. Menegaz, "MPSteg: hiding a message in the Matching Pursuit domain," *Proceedings of SPIE*, vol. 6072, 2006.

[3] G. Cancelli and M. Barni, "MPSteg-color: A new steganographic technique for color images," *Information Hiding: 9th International Workshop, Ih 2007, Saint Malo, France, June 11-13*, vol. 4567, pp. 1–15, 2007.

[4] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[5] P. Jost, P. Vandergheynst, and P. Frossard, "Redundant image representations in security applications," *International Conference on Image Processing (ICIP), 2004*, vol. 4, 2004.

[6] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," *IEEE Transactions on Signal Processing*, vol. 53, no. 10 Part 2, pp. 3923–3935, 2005.

[7] M. Goljan, J. Fridrich, and T. Holotyak, "New blind steganalysis and its implications," *Proceedings of SPIE*, vol. 6072, pp. 1–13, 2006.

[8] K. Solanki, A. Sarkar, and B. Manjunath, "YASS: yet another steganographic scheme that resists blind steganalysis," *Information Hiding: 9th International Workshop, IH 2007, Saint Malo, France, June 11-13*, vol. 4567, pp. 16–31, 2007.

[9] Z. Fan and R. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *Image Processing, IEEE Transactions on*, vol. 12, no. 2, pp. 230–235, 2003.

[10] J. Zhang, I. Cox, and G. Doërr, "Steganalysis for LSB Matching in images with high-frequency noise," *IEEE 9th Workshop on Multimedia Signal Processing (MMSP), 2007*, pp. 385–388, 2007.

[11] A. Ker, "Steganalysis of LSB matching in grayscale images," *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 441–444, 2005.

[12] J. Harmsen, "Steganalysis of additive noise modelable information hiding," Ph.D. dissertation, Rensselaer Polytechnic Institute, 2003.

[13] S. Lyu and H. Farid, "Detecting hidden messages using higher-order statistics and support vector machines," *Information Hiding: 5th International Workshop, IH 2003*, pp. 340–354, 2003.

[14] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," *Machine Learning*, vol. 31, 2004.

[15] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Wiley-Interscience, 2000.

APPENDIX A
PROOF OF THEOREM 1

Theorem 1: Let $\mathcal{I} = \mathcal{R}^0$ be an image and let $\vec{g}_\gamma = (g_{\gamma_1}, \ldots, g_{\gamma_n})$ be a decomposition path. We suppose that the atoms are binary valued, i.e. they take only values 0 or 1. Let assume that the MP decomposition coefficients are computed iteratively by means of the following operations:

$$c_k = \max\{c \geq 0 : \mathcal{R}^{k-1} - c g_{\gamma_k} \geq 0 \quad \textit{for every pixel}\} \tag{21}$$

$$\mathcal{R}^k = \mathcal{R}^{k-1} - c_k g_{\gamma_k}, \tag{22}$$

and let $\vec{c} = (c_1, c_2 \ldots c_n)$ be the coefficient vector built after n iterations. Let c_k be an element of $\vec{c}$ with $c_k \neq 0$, and let $\vec{c}'$ be a modified version of $\vec{c}$ where c_k has been replaced by c'_k. If we apply the MP decomposition to the modified image

$$\mathcal{I}' = \sum_{i=1, i\neq k}^{n} c_i \cdot g_{\gamma_i} + c'_k g_{\gamma_k} + \mathcal{R}^n \tag{23}$$

by using the decomposition path $\vec{g}_\gamma$, we re-obtain exactly the same vector $\vec{c}'$ and the same residual $\mathcal{R}^n$.

Proof:

To prove the theorem we introduce some notations. We indicate by $S(g_{\gamma_k})$ the support of the atom (γ_k)[9]. This notation, and the fact that $g_{\gamma_k}(x,y) \in \{0,1\}\ \forall (x,y)$, permits us to rewrite the rule for the computation of c_k as follows:

$$c_k = \min_{(x,y)\in S(g_{\gamma_k})} \mathcal{R}^{k-1}(x,y). \tag{24}$$

We indicate by j_k the coordinates for which the above minimum is reached, i.e.:

$$j_k = \arg \min_{(x,y)\in S(g_{\gamma_k})} \mathcal{R}^{k-1}(x,y). \tag{25}$$

Note that after the update we will always have $\mathcal{R}^k(j_k) = 0$. We also find it useful to define the set $\mathcal{J}_k = \bigcup_{i=1}^{k} j_i$, with $\mathcal{J}_0 = \emptyset$. In the following we will indicate with $\mathcal{R}$ the residuals computed by applying the decomposition path $\vec{g}_\gamma$ to $\mathcal{I}$, while we will indicate with $\mathcal{R}'$ the residuals obtained by applying the same decomposition path to $\mathcal{I}'$. A similar notation applies to the other symbols we have defined. Let now c_k be a non-zero element of $\vec{c}$. We surely have $S(g_{\gamma_k}) \cap \mathcal{J}_{k-1} = \emptyset$ since otherwise we would have $c_k = 0$. Let us show first that by applying the MP to $\mathcal{I}'$ the coefficients of the atoms g_{γ_h} with $h < k$ do not change. Without loss of generality let h be the first element for which c_h may have changed. Two cases are possible: $S(g_{\gamma_k}) \cap S(g_{\gamma_h}) = \emptyset$ or $S(g_{\gamma_k}) \cap S(g_{\gamma_h}) \neq \emptyset$. In the first case it is evident that the weight c_h can not change, since a modification of the weight assigned to g_{γ_k} cannot have any impact on (24) given that the minimization is performed on $S(g_{\gamma_h})$.

When the intersection between $S(g_{\gamma_h})$ and $S(g_{\gamma_k})$ is non-empty the proof is split in two parts, the former considers the case $c'_k > c_k$, the latter the case $c'_k < c_k$. When $c'_k > c_k$ some of the values in $\mathcal{R}'^{h-1}$ are increased, however $\mathcal{R}'^{h-1}(j_h)$ does not change since $S(g_{\gamma_k}) \cap \mathcal{J}_{k-1} = \emptyset$, hence leaving the choice of j_h and the computation of the weight c_h unchanged.

If $c'_k < c_k$, some values in $\mathcal{R}'^{h-1}$ are decreased while leaving $\mathcal{R}'^{h-1}(j_h)$ unchanged. However, $\forall (x,y) \in S(g_{\gamma_k}) \cap S(g_{\gamma_h})$ we have $\mathcal{R}^{k-1}(x,y) \leq \mathcal{R}^h(x,y)$ since due to the

[9]The support of an atom is defined as the set of coordinates (x,y) for which $g_{\gamma_k}(x,y) \neq 0$

particular update rule we adopted, at each iteration the values in the residual cannot increase. For this reason at the h-th selection step, the modification of the k-th coefficient cannot decrease the residual by more than $\mathcal{R}^{h-1} - c_h$ (remember that $c_h = \mathcal{R}^{h-1}(j_h)$). In other words, $\mathcal{R}'^{h-1}(x,y)$ computed on the modified image $\mathcal{I}'$ will satisfy the relation $\mathcal{R}'^{h-1}(x,y) \geq \mathcal{R}'^{h-1}(j_h)$ hence ensuring that $c'_h = c_h$.

We must now show that the components $h \geq k$ of the vector $\vec{c}$ do not change as well. Let us start with the case $h = k$. Since no coefficient has changed until position k, when the MP is applied to the image $\mathcal{I}'$ we have

$$c''_k = \min_{(x,y)\in S(g_{\gamma_k})} \left[\mathcal{R}^{k-1}(x,y) + (c'_k - c_k) g_{\gamma_k}(x,y)\right]. \quad (26)$$

From equation (26) it is evident that

$$c''_k = c'_k = \min_{(x,y)\in S(g_{\gamma_k})} \mathcal{R}^{k-1}(x,y), \quad (27)$$

since the term $(c'_k - c_k) g_{\gamma_k}$ introduces a constant bias on all the points of $S(g_{\gamma_k})$.

As to the case $h > k$ it is trivial to show that $c'_h = c_h$ given that the residual after the k-th step will be the same for $\mathcal{I}$ and $\mathcal{I}'$. ■

APPENDIX B
FISHER LINEAR DISCRIMINANT ANALYSIS

Fisher Linear Discriminant (FLD) analysis basically seeks directions that are efficient for discrimination. The goal is to find an orientation $\mathbf{u}$ for which the samples in the dataset, once projected onto it, are well separated. Let us assume that a dataset $\mathcal{D}$ is made of N d-dimensional samples $\mathbf{x_1}, \ldots, \mathbf{x}_N$, N_1 being in a subset $\mathcal{D}_1$ corresponding to one class and N_2 being in a subset $\mathcal{D}_2$ corresponding to the other class. The first step of FLD analysis consists in computing the d-dimensional sample mean of each class:

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{x}\in\mathcal{D}_i} \mathbf{x}. \quad (28)$$

Next, the scatter matrix $\mathbf{S}_W = \mathbf{S}_1 + \mathbf{S}_2$ is computed using the following definitions:

$$\mathbf{S}_i = \sum_{\mathbf{x}\in\mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t. \quad (29)$$

Finally, the direction of projection $\mathbf{u}$ is given by:

$$\mathbf{u} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2). \quad (30)$$

This vector $\mathbf{u}$ defines a linear function $y = \mathbf{u}^t\mathbf{x}$ which yields the maximum ratio of between-class scatter to within-class scatter. The interested reader is redirected to [15] for further details (pp. 117–121).

Giacomo Cancelli was born in Siena, Italy, in 1980. He received the Informatics Engineer degree from the University of Siena, Siena, Italy, in 2005. Since 2005 he has been with the Department of Information Engineering, University of Siena, Italy, where he is currently pursuing the Ph.D. degree in the area of data hiding. His main research interests are in steganalysis, steganography and watermarking.

Mauro Barni graduated in electronic engineering at the University of Florence in 1991. He received the PhD in informatics and telecommunications in October 1995. He has carried out his research activity for over 15 years first at the Department of Electronics and Telecommunication of the University of Florence, then at the Department of Information Engineering of the University of Siena where he works as associate Professor. During the last decade he has been studying the application of image processing techniques to copyright protection and authentication of multimedia (digital watermarking). He is author/co-author of about 180 papers published in international journals and conference proceedings, and holds three patents in the field of digital watermarking. He is co-author of the book "Watermarking Systems Engineering: Enabling Digital Assets Security and other Applications", published by Dekker Inc. in February 2004.

He participated to several National and European research projects on diverse topics, including computer vision, multimedia signal processing, remote sensing, digital watermarking, IPR protection. In particular he is the coordinator of the project SPEED - Signal Processing in the EncryptEd Domain funded by the EC under the FP6 (FET - program).

He is the editor in chief of the EURASIP Journal on Information Security. He serves as associate editor of the IEEE Trans. on Circuits and system for Video Technology, the IEEE Signal Processing Letters, the IEt Proceedings on Information Security. Prof. Barni is a member of the IEEE Information Forensic and Security technical Committee (IFS-TC) of the IEEE Signal Processing Society. He is a senior member of the IEEE and EURASIP.