Hash-based identification of sparse image tampering

Marco Tagliasacchi* *Member, IEEE*, Giuseppe Valenzise *Student Member, IEEE*, Stefano Tubaro *Member, IEEE*

***Abstract*— In the last decade, the increased possibility to produce, edit and disseminate multimedia contents has not been adequately balanced by similar advances in protecting these contents from unauthorized diffusion of forged copies. When the goal is to detect whether or not a digital content has been tampered with in order to alter its semantics, the use of multimedia hashes turns out to be an effective solution to offer proof of legitimacy and to possibly identify the introduced tampering. We propose an image hashing algorithm based on compressive sensing principles, which solves both the authentication and the tampering identification problems. The original content producer generates a hash using a small bit budget by quantizing a limited number of random projections of the authentic image. The content user receives the (possibly altered) image, and uses the hash to estimate the mean square error distortion between the original and the received image. In addition, if the introduced tampering is sparse in some orthonormal basis or redundant dictionary, an approximation is given in the pixel domain. We emphasize that the hash is universal, e.g. the same hash signature can be used to detect and identify different types of tampering. At the cost of additional complexity at the decoder, the proposed algorithm is robust to moderate content-preserving transformations including cropping, scaling and rotation. In addition, in order to keep the size of the hash small, hash encoding/decoding takes advantage of distributed source codes.**

***Index Terms*— Image Hashing, Compressive Sensing, Distributed Source Coding**

EDICS: COM-MMC, COM-DSSC, COM-WSE

I. Introduction

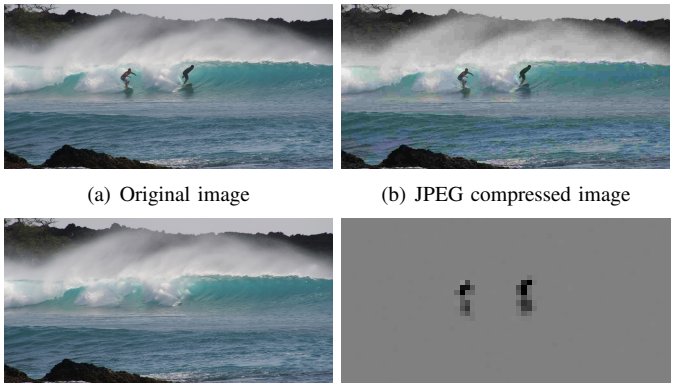
With the overwhelming diffusion of multimedia contents in every-day life, protecting the authenticity and the integrity of these contents from undesired manipulations has become an increasingly important research theme. Examples of potential applications may be found in video-surveillance, where it is necessary to have guarantees that the captured snapshots are genuine and have not been tampered with in any way, or in the case of digital contents transmitted over peer-to-peer (P2P) networks, where automatic mechanisms to assess the legitimacy of a copy are desirable. The risks for security are further exacerbated by the improved possibilities of tampering with media contents such as photos, an ability that would have traditionally required many hours of cumbersome work in a darkroom and that has become now a simple practice using a computer and some commercial software tools.

In the case of images, different versions of the same file might differ from the original because of processing such as transcoding or bitstream truncation. Other legitimate content-preserving alterations of the original picture are also possible, when the image is enhanced by means of photo editing tools. These modifications include, for instance, moderate geometrical transformations or slight brightness adjustments. In other cases, however, one could tamper with part of the image and possibly affect its semantic content in order to illegally abuse it, e.g. to manipulate public opinion or to influence the verdict of the jury in a criminal trial. Given these premises, it is not surprising that a great deal of attention has been turned to methods able to offer proof of authenticity of an image and, in the case of detected tampering, to identify which kind of attack has been carried out. The reasons why it is generally preferred to identify how the content has been tampered with are twofold: on one hand, given an estimate of *where* the content was manipulated one can establish whether or not the image file is still meaningful for the final user; on the other hand, in some circumstances it may be possible to recover the original semantics of the image content.

In this paper we propose a hashing technique based on compressive sensing principles, which is secure against malicious forgeries and robust w.r.t. legitimate content-preserving manipulations such as moderate rotations or cropping. The key tenet of the described technique is that, if the tampering is sparse enough (or it can be sparsified in some orthonormal basis or redundant dictionary), it can be localized by solving a convex optimization problem with constraints imposed by the transmitted hash. As pointed out later in this section, this approach enhances the current state of the art in tampering detection and localization by enabling a higher flexibility in the forgery reconstruction phase (which can be arbitrarily empowered at the decoder without requiring any change in the hash structure). Furthermore, this task is achieved with a small amount of overhead bits for the hash transmission, which makes the system particularly attractive for applications like Digital Rights Management (DRM) over the Internet.

For the sake of clarity, we anticipate here a few examples of the output produced by the proposed algorithm. Figure 1 shows two different versions of the same image (depicted in part (a) of the picture): in Figure 1(b), the image has been re-encoded using JPEG, whereas Figure 1(c) shows an example of a malicious attack. In both cases, the Peak Signal-to-Noise ratio (PSNR) with respect to the original is equal to 31.5dB. The information contained in the hash enables to: 1) estimate the distortion between the image and its original version; 2) identify the tampering in the pixel domain. Figure 1(d) shows the output of the proposed algorithm when the bit budget spent for the hash is as small as 0.005 bpp. This amounts to approximately 330 bytes for the 1024×512 image in Figure 5. Another example is illustrated in Figure 2. The difference between the original and the tampered image is

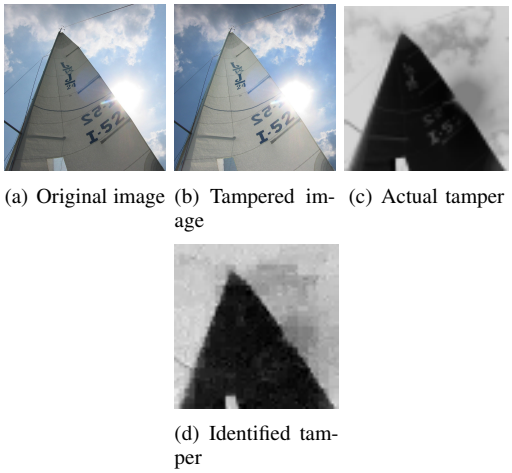
The authors are with Dipartimento di Elettronica e Informazione, Politecnico di Milano, P.zza Leonardo da Vinci, 32 20133 - Milano, Italy - Ph. +39-02-2399-7624 - FAX: +39-02-2399-7321 - E-mail: marco.tagliasacchi@polimi.it, valenzise@elet.polimi.it, tubaro@elet.polimi.it.
This work has been partially sponsored by the EU under Visnet II Network of Excellence. The material in this paper has been partially presented in the 15th IEEE International Conference on Image Processing, San Diego, CA, 2008.



(a) Original image (b) JPEG compressed image

(c) Tampered image (d) Identified tamper

Fig. 1. Identification of tampering sparse in the pixel domain.



(a) Original image (b) Tampered image (c) Actual tamper

(d) Identified tamper

Fig. 2. Identification of a tamper sparse in the wavelet domain.

shown in Figure 2(c). In this case, the tampering is not sparse in the pixel domain, but it is piecewise smooth. Hence, it can be represented by a sparse set of coefficients in the wavelet domain. Figure 2(d) depicts the tampering estimated by the proposed algorithm, where the hash bit budget has been set equal to 0.009 bpp. We emphasize the universality of our hash signature: the content owner produces a hash signature which only depends on the original image, without making any specific assumption about the actual tampering that may occur.

In the literature, two main approaches have been used to solve the problem of tampering identification. One class of techniques, which we refer to as "blind" approaches, quantify statistical correlations that result from specific forms of digital tampering, without requiring any type of additional signatures or side information beyond the analyzed content. These techniques have been popularized by Farid et al. in their works for detecting digital forgeries in scientific images [1] or photographic composites of people [2], as well as for exposing digital tampering through specular highlights on the eyes [3] or chromatic aberrations [4]. Although these methods have the unquestionable advantage of being able to be used also for images whose original copy is difficult to obtain or has been definitively lost, in practice they have the inconvenient drawback of requiring specific modeling for each kind of possible attack. Moreover, blind techniques are strongly suboptimal when the original image is available from the content producer. In this case, it is often more feasible to use some compact representation of the original content, in the form of a signature or watermark, to compare the presumably corrupted images with the original. Finally, blind approaches are based on general statistics of tampered images, but there is no guarantee that a well-faked image will not be judged as authentic.

A second set of methodologies for tampering detection postulates the availability of a concise representation of the original content's salient features. These techniques can be further divided into *watermarking* and *hash-based* schemes. Digital watermarking techniques embed information directly into the media data to ensure both data integrity and authentication. Content-fragile watermarks are generally used for checking the integrity of an image file: a mark designed to be robust w.r.t. legitimate, perceptually-irrelevant modifications (e.g. compression), and at the same time to be fragile w.r.t. perceptually and semantic significant alterations is inserted into the original host image. When a new copy of the media content is examined, possible tampering can be detected and localized by identifying the damage to the extracted watermark. Examples of this approach for the case of image content types are given in [5] and [6]. The authors of [7] propose an image authentication scheme that is able to localize tampering, by embedding a watermark in the wavelet coefficients of an image. If a tampering occurs, the system provides information on specific frequencies and space regions of the image that have been modified. The image watermarking system devised in [8] inserts a fragile watermark in the least significant bits of the image on a block-based fashion; when a portion of the image is tampered with, only the watermark in the corresponding blocks is destroyed, and the manipulation can be localized. Celik et. al. [9] extend this method by inserting a watermark in a hierarchical way, to improve robustness against vector quantization attacks. In [10], image protection and tampering localization is achieved by inserting two complementary watermarks in the original image, through a technique known as "cocktail watermarking".

In spite of their widespread diffusion as an image protection tool, watermarking schemes suffer from a series of disadvantages: they are not backward compatible with previously encoded contents (unmarked contents cannot be authenticated later); the original content is distorted by the watermark; the bit-rate required to compress a multimedia content might increase due to the embedded watermark. An alternative to image watermarking is the use of *multimedia hashes*, where a signature of the original content is embedded as part of the header information, or can be provided separately from the content upon a user's request. Examples of the use of content-hashing for media authentication or retrieval are given in [11][12][13][14]. In a general content-hashing scheme, the produced signature consists of a compact representation of some image features – selected to be robust to geometric distortions, filtering operations and various content-preserving

manipulations – so that legitimate copies can be authenticated using some distance/similarity measure directly on the hash. At the same time, using such features alone makes the system susceptible to forgery attacks, which may be carried out by an attacker that creates a new image with different visual content but with the same feature values. Thus, *security* mechanisms [15] must be combined into the feature extraction stage, e.g. by introducing some pseudorandom key in the hashing system. In [11], the feature extraction consists of a random tiling of the discrete wavelet transform subbands of the image. The means (or variances) of the pixel values in each random rectangle are used to form the feature vector, which is randomly quantized and compressed to generate a hash. The authors of [12] compute a singular value decomposition from random blocks in the image, and use the obtained intermediate features to produce a robust and secure image hash. Swaminathan et al. [15] exploit the rotation invariance of the Fourier-Mellin transform together with controlled randomization during image feature extraction. Using this method, it is possible to produce a secure hash signature which is robust to affine transformations, while keeping the hash size as small as 400 bits for a 512×512 image. The output of the feature extraction stage is usually quantized and converted to binary representation to reduce the bit-length of the hash, e.g. by means of a uniform or Lloyd-Max quantizer. In [11], a key-dependent randomized quantizer is proposed for hash quantization [11]. More recently, the use of Distributed Source Codes for hash encoding, pioneered by Johnson and Ramchandran [16], has been used for image hashing by Lin et al. [17]. In their work, the hash consists of syndrome bits produced by LDPC (Low Density Parity-Check Codes) encoding applied to quantized random projections of the original image. To perform authentication, a Slepian-Wolf decoder receives in input the hash and the (possibly tampered) image, which serves as side information. During a training phase, the maximum rate required to successfully decode authentic images (JPEG or JPEG2000 compressed versions of the original image up to a PSNR of 30 dB) is determined. If a malicious tampering occurs, it is expected that the correlation between the original and the tampered image is weaker, and LDPC decoding fails. Conversely, if decoding succeeds, the image is declared authentic. The bit-length of the hash for a 512×512 image with this system is about 100 bytes. This scheme has been later extended in [18] to perform tampering localization with a two-step process: first, image authentication proceeds as explained above; second, if authentication fails, at the cost of extra syndrome bits it is possible to localize the tampering in the pixel domain. The bit-rate required for this second step is determined by means of a training stage.

The proposed algorithm shares some similarities with the work in [17][18], especially in the hash generation phase. In fact, as in [18], we extract a feature vector consisting of pseudorandom projections in the pixel domain, computed with a macroblock granularity. Subsequently a hash is generated by storing the syndrome bits derived from LDPC encoding of the quantized projections. With respect to [18], the proposed algorithm is novel in the following aspects. First, by leveraging compressive sensing principles, we are able to identify a tampering that is not sparse in the pixel domain, but that can be represented by a sparse set of coefficients in some orthonormal basis or redundant dictionary. Second, we build on the recent results in the field of compressive estimation and detection [19][20] to perform image registration before tampering identification, in order to be robust to geometric transformations. Third, we analyze the problem of rate allocation, in such a way that the bit-rate spent for the hash can be determined at the encoder based on the maximum expected energy of the tampering. Thus, no training stage is required to determine the hash bit-rate.

The rest of this paper is organized as follows. Section II provides a very short summary (with references) of the background necessary to comprehend the basic building blocks of the system. Section III gives a general description of the proposed tampering identification scheme. In Section IV we give some guidelines about how to set the parameters of the system, by outlining a tampering model and using it for parameter tuning. Section V discusses the problem of rate allocation in the hypothesis that the system can protect up to some maximum tampering energy. Section VI explains how the system can be made robust to moderate content-preserving transformations. In Section VII, we discuss the experimental results obtained by applying the proposed algorithm on some example image manipulations, including logo insertion, brightness adjustment and cropping. Finally, Section VIII gives some concluding remarks.

II. Background

Before illustrating in detail the tampering identification system in Section III, in this section we provide a brief summary of two topics that play a central role in the proposed system. In Section II-A we discuss Wyner-Ziv coding, which enables to reduce the bit budget required to represent the hash signature. In Section II-B we illustrate the foundations of compressive sensing, a recent paradigm that is employed in our scheme in order to identify the tampering from a limited number of random projections, provided that it has a sparse representation in some orthonormal basis.

A. Wyner-Ziv coding

Consider the problem of communicating a continuous random variable X. Let Y denote another continuous random variable correlated to X. In a distributed source coding setting, the problem is to decode X to its quantized reconstruction $\hat{X}$ given a constraint on the distortion measure $D = E[d(X, \hat{X})]$ when the side information Y is available only at the decoder. Let us denote by $R_{X|Y}(D)$ the rate-distortion function for the case when Y is also available at the encoder, and by $R^{WZ}_{X|Y}(D)$ the case when only the decoder has access to Y. The Wyner-Ziv theorem [21] states that, in general, $R^{WZ}_{X|Y}(D) \geq R_{X|Y}(D)$, i.e. the rate needed to encode the source X, when the side information Y is known only at the decoder, is greater or equal than the rate needed when X and Y are jointly encoded, i.e. when also the encoder has access to Y. However $R^{WZ}_{X|Y}(D) = R_{X|Y}(D)$ when X and Y are jointly Gaussian memoryless sources and the distortion measure used is the Mean Square Error (MSE).

Recently, the Wyner-Ziv theorem has been applied in the area of video coding under the name of Distributed Video Coding (DVC). Most of the DVC schemes in the literature follow the approach described in [22] targeting low encoding complexity. The source X (pixel values or DCT coefficients of a frame) is quantized with 2^J levels, and the J bitplanes are independently encoded, computing parity bits by means of a turbo encoder. At the decoder, parity bits are used together with the side information Y (a motion-compensated predictor of the current frame) to "correct" Y into a quantized version of X, $\hat{X}$, performing turbo decoding, typically starting from the most significant bitplanes. To this end, the decoder needs to know the joint p.d.f. (probability density function) $p_{XY}(X,Y)$. More recently, LDPC codes have been adopted instead of turbo codes [23][24].

B. Compressive sensing

Compressive sensing (CS) is a recent paradigm that enables the reconstruction of a discrete signal from a limited number of random projections, provided that the signal can be represented by a small number of non-zero coefficients in some basis expansion.

Let $\mathbf{x} \in \mathbb{R}^n$ denote the signal of interest and $\mathbf{y} \in \mathbb{R}^m$, $m < n$, a number of linear random projections (measurements) obtained as $\mathbf{y} = \mathbf{A}\mathbf{x}$. The measurement matrix $\mathbf{A}$ must be chosen in such a way that it satisfies a *Restricted Isometry Property* (RIP) of order k [25], which says that all subsets of k columns taken from $\mathbf{A}$ are in fact nearly orthogonal or, equivalently, that linear measurements taken with $\mathbf{A}$ approximately preserve the Euclidean length (up to a m/n scaling) of k-sparse signals. The entries of the measurement matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ can be random samples from a given statistical distribution, e.g. Gaussian or Bernoulli. At first, let us assume that $\mathbf{x}$ is k-sparse, i.e. there are exactly $k \ll n$ nonzero components. The goal is to reconstruct $\mathbf{x}$ given the measurements $\mathbf{y}$ and the knowledge that $\mathbf{x}$ is sparse. This can be formulated as the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \|\mathbf{x}\|_0 \\ &\text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \end{aligned} \tag{1}$$

where the ℓ_0 norm (represented as $\|\cdot\|_0$) simply counts the number of nonzeros entries of $\mathbf{x}$ [26]. Unfortunately, an exact solution to this problem requires an exhaustive search over all the possible $\binom{n}{k}$ k-sparse solutions, and is therefore computationally intractable. Nonetheless, the recent results of compressive sensing [26] have shown that, if $\mathbf{x}$ is sufficiently sparse, an approximation of it can be recovered by solving the following ℓ_1 minimization problem:

$$\begin{aligned} &\text{minimize} \quad \|\mathbf{x}\|_1 \\ &\text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \end{aligned} \tag{2}$$

which can be immediately cast as a linear program. The solution of (2) is the same as (1) provided that the number of measurements satisfies $m \geq C \cdot k \log_2(n/k)$, where C is some small positive constant. Moreover, if $\mathbf{x}$ is not exactly sparse, but it is at least *compressible* (i.e. its coefficients decay as a power law), then solving (2) guarantees that the quality of the recovered signal is as good as if one knew ahead of time the location of the k largest values of $\mathbf{x}$ and decided to measure those directly [25].

These results also hold when the signal is not sparse, but it has a sparse representation in some orthonormal basis. Let $\mathbf{\Phi} \in \mathbb{R}^{n \times n}$ denote an orthonormal matrix, whose columns are the basis vectors. Let us assume that we can write $\mathbf{x} = \mathbf{\Phi}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is k-sparse. Given the measurements $\mathbf{y} = \mathbf{A}\mathbf{x}$, the signal $\mathbf{x}$ can be reconstructed by solving the following problem:

$$\begin{aligned} &\text{minimize} \quad \|\boldsymbol{\alpha}\|_1 \\ &\text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{\Phi}\boldsymbol{\alpha} \end{aligned} \tag{3}$$

For the case of noisy measurements, the signal model can be expressed as $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where the noise amplitude is assumed to be bounded, i.e. $\|\mathbf{z}\|_2 \leq \epsilon$. This situation occurs when the measurements are quantized. An approximation of the signal $\mathbf{x}$ can be obtained by solving the following problem:

$$\begin{aligned} &\text{minimize} \quad \|\boldsymbol{\alpha}\|_1 \\ &\text{subject to} \quad \|\mathbf{y} - \mathbf{A}\mathbf{\Phi}\boldsymbol{\alpha}\| \leq \epsilon \end{aligned} \tag{4}$$

Problem (4) is a special instance of a second order cone program (SOCP) [27] and can be solved in $O(n^3)$ computations. Nevertheless, several fast algorithms have been proposed in the literature that attempt to find a solution to (2). In this work, we adopt the SPGL1 algorithm [28], which is specifically designed for large scale sparse reconstruction problems.

III. Proposed hash-based tampering identification system

A. System overview

The proposed tampering identification scheme is depicted in Figure 3. The producer of the original content generates a small hash signature starting from the original image $\mathbf{X} \in \mathbb{R}^N$, where N denotes the total number of pixels. The content is distributed over a network consisting of possibly untrusted nodes and the received image $\bar{\mathbf{X}}$ might differ from the original. In order to reduce the size of the hash signature, we work with decimated images, denoted by the corresponding lowercase letters $\mathbf{x}$ and $\bar{\mathbf{x}}$. We model the tampering attack as an additive noise component followed by a geometric transformation. In formulae:

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{\Phi}\mathbf{e} \tag{5}$$

$$\bar{\mathbf{x}} = f(\tilde{\mathbf{x}}; \boldsymbol{\theta}) \tag{6}$$

In equation (5), $\mathbf{e}$ represents a sparse (or compressible) signal, while $\mathbf{\Phi}$ is an orthonormal matrix whose columns denote the basis vectors of an arbitrary basis. If $\mathbf{\Phi} = I$, i.e. the identity matrix, the model implies that the attack is localized in the pixel domain. In equation (6), $f(\cdot)$ denotes an arbitrary geometric transformation characterized by the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^K$. For example, in the case of shifts of an image, $\boldsymbol{\theta} \in \mathbb{R}^2$ represents the coordinates of the displacement vector in the 2D plane.

Users who want to authenticate the received image $\bar{\mathbf{X}}$ use the hash to estimate the MSE distortion between the original

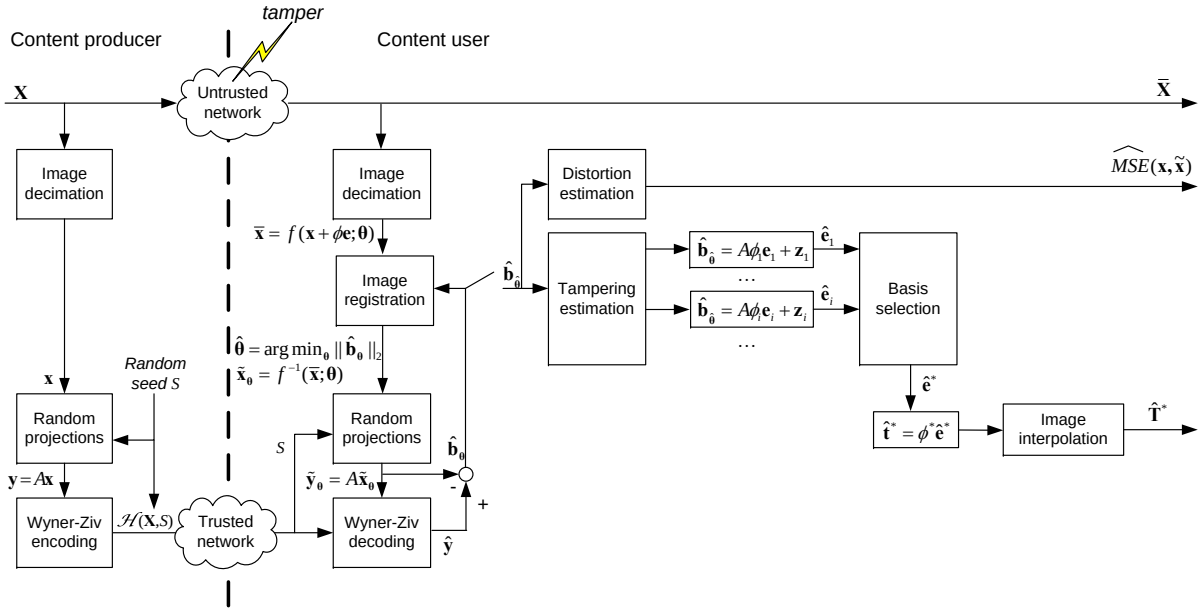


Fig. 3. Block diagram of the proposed tampering identification scheme.

subsampled image $\mathbf{x}$ and the geometric registered one $\tilde{\mathbf{x}}$. Starting from the random projections $\mathbf{y}$ and $\tilde{\mathbf{y}}$, the algorithm reconstructs an estimation of the tampering $\hat{\mathbf{e}}_i$ for a set of possible bases $\mathbf{\Phi}_i$. The most plausible reconstruction $\hat{\mathbf{e}}^*$ is then used to produce a tampering map $\mathbf{T}_i \in \mathbb{R}^N$ indicating the characteristics of the attack in the pixel domain, as illustrated by the examples in Figure 1(d) and Figure 2(d).

B. Hash generation

The original content producer generates the hash signature $\mathcal{H}(\mathbf{X}, S)$ as follows:

1) Image decimation: The original image $\mathbf{X}$ is partitioned into blocks of size $B \times B$. The average of the luminance component of each block is computed and stored in a vector $\mathbf{x} \in \mathbb{R}^n$, where n denotes the number of blocks in the image, i.e. $n = N/B^2$.

2) Random projections: A number of linear random projections $\mathbf{y} \in \mathbb{R}^m$, $m < n$, is produced as $\mathbf{y} = \mathbf{A}\mathbf{x}$. The entries of the matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ are sampled from a Gaussian distribution $\mathcal{N}(0, 1/n)$, generated using a random seed S, which will be sent as part of the hash to the user. The random seed S works as a sort of secret key to guarantee computational security against malicious attacks which may exploit the knowledge of the nullspace of the projection matrix $\mathbf{A}$ to break the system. The choice of the number of random projections m depends on the expected sparsity (in some arbitrary orthonormal basis or redundant dictionary).

3) Wyner-Ziv encoding: The random projections $\mathbf{y}$ are quantized with a uniform scalar quantizer with step size Δ. In order to reduce the number of bits needed to represent the hash, we do not send directly the quantization indexes. Instead, we observe that the random projections computed from the tampered image will be available at the decoder side. Therefore, we can perform lossy encoding with side information at the decoder, where the source to be encoded is $\mathbf{y}$ and the "noisy" random projections $\tilde{\mathbf{y}} = \mathbf{A}\tilde{\mathbf{x}}$ play the role of the side information. With respect to the distributed source coding notation introduced in Section II-A, we have $X = \mathbf{y}$ and $Y = \tilde{\mathbf{y}}$. At the end of this section we will show that $\mathbf{y}$ and $\tilde{\mathbf{y}}$ are independent and identically distributed (i.i.d.) and jointly Gaussian, thus the hypotheses of the Wyner-Ziv theorem hold. Following the approach widely adopted in the literature on distributed video coding [22], we perform bitplane extraction on the quantization bin indexes followed by Slepian-Wolf coding. The number of encoded bitplanes J is determined by:

$$J = \left\lceil \log_2 \frac{4\sigma_y}{\Delta} \right\rceil + 1, \tag{7}$$

where σ_y is the square root of the sample variance of the $\mathbf{y}$ vector. This ensures that the dynamic range of the granular region of the quantizer is $[-\Delta 2^{J-1}, +\Delta(2^{J-1} - 1)]$ ($\supseteq [-4\sigma_y, +4\sigma_y]$), and the probability of saturating the quantizer is less than 10^{-4}. Hence, each bitplane is encoded by sending syndrome bits generated by means of a LDPC code. Here, we devise two modes of operation.

- If a feedback channel is available, syndrome bits are stored and progressively transmitted to the content user upon request, until the hash can be successfully decoded.
- Conversely, if rate allocation is performed at bitplane level directly at encoding time, the content producer imposes the maximum tolerated MSE distortion between the original and the tampered image. Rate allocation is further discussed in Section V.

C. Tampering identification

The content user receives the (possibly tampered) image $\bar{\mathbf{X}}$ and requests the syndrome bits $\mathcal{H}(\mathbf{X}, S)$ of the hash and the random seed S necessary to generate it to the authentication server. On each user's request, a different seed S is used

in order to avoid that a malicious attack could exploit the knowledge of the nullspace of $\mathbf{A}$. Image authentication and tampering identification work as follows:

1) Image decimation: The received image $\bar{\mathbf{X}}$ is decimated as in the encoder to produce the vector $\bar{\mathbf{x}} \in \mathbb{R}^n$.

2) Image registration: The received decimated image is warped by means of an inverse geometric transformation $f^{-1}(\cdot)$ to match the original image. Let $\tilde{\mathbf{x}}_{\boldsymbol{\theta}}$ denote the warped image vector obtained with an arbitrary parameter vector $\boldsymbol{\theta}$:

$$\tilde{\mathbf{x}}_{\boldsymbol{\theta}} = f^{-1}(\bar{\mathbf{x}}, \boldsymbol{\theta}) \tag{8}$$

The parameter vector $\boldsymbol{\theta}$ is unknown, but it can be estimated using a "smashed filter" approach [19], i.e. by comparing the received random projections with the ones computed from the warped decimated image $\tilde{\mathbf{x}}_{\boldsymbol{\theta}}$.

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\hat{\mathbf{y}} - \tilde{\mathbf{y}}_{\boldsymbol{\theta}}\|_2 \tag{9}$$

The estimate $\hat{\boldsymbol{\theta}}$ can be further refined by applying the inverse transformation f^{-1} to the full resolution tampered image, as discussed in Section VI.

3) Random projections: Fix a value of $\boldsymbol{\theta}$, and compute the random projections $\tilde{\mathbf{y}}_{\boldsymbol{\theta}} = \mathbf{A}\tilde{\mathbf{x}}_{\boldsymbol{\theta}}$. Matrix $\mathbf{A}$ is known at the decoder side through the random key S transmitted with the hash.

4) Wyner-Ziv decoding: For a given value of $\boldsymbol{\theta}$, a quantized version $\hat{\mathbf{y}}$ of the random projections of the original image is obtained using the hash syndrome bits and $\tilde{\mathbf{y}}_{\boldsymbol{\theta}}$ as side information. Subsequently LDPC decoding is performed starting from the most significant bitplane.

- If a feedback channel is not available, decoding might fail when the actual distortion between original and the tampered image is higher than the maximum tolerated distortion determined by the original content producer.
- Conversely, if a feedback channel is available, decoding always succeeds. Unfortunately, this might require a large number of bits, since small geometric transformation might cause a large increase in MSE distortion between the original and the tampered image, thus requesting more syndrome bits. Therefore, in practice, it is advisable to impose an upper bound on the maximum number of hash bits, similarly to the no-feedback channel case discussed above.

The two steps are repeated for different values of $\boldsymbol{\theta}$ until the optimal value $\hat{\boldsymbol{\theta}}$ in (9) is attained. When Wyner-Ziv decoding fails for a given value of $\boldsymbol{\theta}$, set $||\hat{\mathbf{y}} - \tilde{\mathbf{y}}_{\boldsymbol{\theta}}||_2 = +\infty$. When Wyner-Ziv decoding fails for all values of $\boldsymbol{\theta}$, the image is declared to be unauthentic and no tampering identification can be provided.

5) Distortion estimation: If Wyner-Ziv decoding succeeds for some value of $\boldsymbol{\theta}$, an estimate of the distortion in terms of the MSE between the original and the received warped image is computed from $\hat{\mathbf{b}}_{\boldsymbol{\theta}} = \hat{\mathbf{y}} - \tilde{\mathbf{y}}_{\hat{\boldsymbol{\theta}}}$.

6) Tampering estimation: A sparse estimate $\hat{\mathbf{e}}_i$ of the tampering is obtained by solving the following optimization problem:

$$\hat{\mathbf{e}}_i = \arg\min \|\mathbf{e}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{b}}_{\boldsymbol{\theta}} - \mathbf{A}\boldsymbol{\Phi}_i\mathbf{e}\|_2 \leq \epsilon \tag{10}$$

where ϵ is chosen such that $\|\mathbf{z}_i\|_2 \leq \epsilon$. The determination of the proper value of ϵ is described in Appendix I.

7) Basis selection: The reconstruction process represented in equation (10) is carried out for different test bases $\boldsymbol{\Phi}_i$, chosen from some given basis dictionary: a larger set of test bases makes more likely that some sparse reconstruction is found, assuming that the attack can be well sparsified in any of these bases. Two orders of problems may arise at this point: 1) the tampering is actually sparse, and may be represented by a sparse set of coefficients in more than one basis (this is the case, for instance, of piece-wise polynomial attacks that are sparse in different wavelet expansions); 2) the tampering is not sparse at all (e.g. quantization noise) or it is not sparse in any of the test bases. In the first situation, an ambiguity arises about which is the best tampering estimate. The solution in this case is straightforward: once the reconstruction has been carried out for different test bases, we simply choose the one in which the tampering is the sparsest, as suggested by CS theory (Section II-B). In practice, instead of selecting the reconstruction with the minimum ℓ_0 metric, we search for the basis expansion $\hat{\mathbf{e}}^*$ that has the smallest ℓ_1 norm, in order to be robust w.r.t. reconstruction noise due to quantization:

$$\hat{\mathbf{e}}^* = \arg\min \|\hat{\mathbf{e}}_i\|_1 \tag{11}$$

In alternative to ℓ_1 norm, other metrics can be employed to take into account the different dynamic ranges of coefficients in different bases, as devised in [29].

The second problem is much tougher, and implies to understand whether the reconstruction $\hat{\mathbf{e}}^*$ is indeed correct. In fact, the tampering estimation represented by equation (10) always admits a sparse solution (with maximum sparsity of the reconstruction approximately equal to m), even when the actual tampering is not sparse, or the number of measurements is not sufficient. In principle, by looking at the recovered tampering $\hat{\mathbf{e}}^*$, one could infer whether or not the number of random projections m is compatible with the sparsity of the reconstruction, using the CS recovery assumption that $m \geq C \cdot k \log(n/k)$. However, this approach suffers from two main drawbacks. First, it is hard to apply to compressible tampering, for which there are not still clear results that tie the rate of decay of the coefficients with the number of measurements required for reconstruction. Second, it is difficult to estimate the true sparsity of the tampering from its reconstruction: in fact, the ℓ_0 norm of the recovered signal tends to be smaller than the one of the original signal, due to the quantization of the measurements. For these reasons, while our system is always able to provide an estimate $\hat{\mathbf{e}}^*$, the problem of automatically taking a decision about the credibility of the received image is subject of current investigations.

8) Tampering reconstruction: Once an estimate $\hat{\mathbf{e}}^*$ of the tampering has been obtained, the orthonormal transform is inverted, to produce $\hat{\mathbf{t}}^* = \boldsymbol{\Phi}^*\hat{\mathbf{e}}^*$, where $\boldsymbol{\Phi}^*$ corresponds to the basis giving the best ℓ_1 norm. Then, the signal $\mathbf{t}^*$ is arranged as a 2D array and interpolated at the original image resolution to get $\hat{\mathbf{T}}^*$.

The tampering identification algorithm presented in this section applies unaltered when the columns of the matrix $\boldsymbol{\Phi}_i$ represent the atoms of a redundant dictionary instead of

an orthonormal basis [30]. This could be useful when the tampering cannot be directly sparsified in a single orthonormal basis. As an example, consider the case of a logo added to a brightness adjusted image: while the logo is sparse in the pixel domain, brightness adjustments can be made sparse in the DCT or wavelet domain.

To complement the above description of the system, we first need to show that the conditions of the Wyner-Ziv theorem are satisfied (in order to justify the adoption of distributed source coding tools to encode the hash signature), and then to discuss how to estimate the MSE between the original and the tampered image from the random projections.

Before proceeding, we introduce a simplified tampering model that does not take into account geometric transformations, which are addressed in detail in Section VI. Therefore, we model the effect of tampering as

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{\Phi e} \tag{12}$$

The random projections $\mathbf{y}$ are encoded given that $\tilde{\mathbf{y}}$ is used as side information at the decoder using distributed source coding tools. The correlation noise is given by

$$\mathbf{b} = \tilde{\mathbf{y}} - \mathbf{y} = \mathbf{A}(\tilde{\mathbf{x}} - \mathbf{x}) = \mathbf{A\Phi e} = \mathbf{At} \tag{13}$$

In Appendix II we show that, if we model the signal $\mathbf{x}$ as a wide-sense stationary random process, the elements of the vectors $\mathbf{b}$, $\mathbf{y}$ and $\tilde{\mathbf{y}}$ are i.i.d. (independent and identically distributed) Gaussian. Moreover, $\mathbf{y}$ and $\tilde{\mathbf{y}}$ are jointly Gaussian. Therefore, the hypotheses of the Wyner-Ziv theorem are satisfied. The Wyner-Ziv decoder needs to know the joint distribution of $\mathbf{y}$ and $\tilde{\mathbf{y}}$, which in this case is completely determined by the variance of the correlation noise $\sigma^2_{\mathbf{b}}$ and of the source $\sigma^2_{\mathbf{y}}$. While the results proven by Wyner and Ziv are non-constructive and asymptotic in nature, here we adopt LDPC codes. Therefore, the actual rate-distortion curve attained in our practical implementation is given by:

$$\begin{aligned} R_{\mathbf{y}|\tilde{\mathbf{y}}}(D) &= R^{WZ}_{\mathbf{y}|\tilde{\mathbf{y}}}(D) + \Delta R = \frac{1}{2}\log_2 \frac{\sigma^2_{\mathbf{b}}\sigma^2_{\mathbf{y}}}{(\sigma^2_{\mathbf{b}} + \sigma^2_{\mathbf{y}})D} + \Delta R \\ &\simeq \frac{1}{2}\log_2 \frac{\sigma^2_{\mathbf{b}}}{D} + \Delta R \end{aligned} \tag{14}$$

where $R^{WZ}_{\mathbf{y}|\tilde{\mathbf{y}}}(D)$ denotes the rate-distortion function for jointly Gaussian sources when $D < \frac{\sigma^2_{\mathbf{b}}\sigma^2_{\mathbf{y}}}{(\sigma^2_{\mathbf{b}}+\sigma^2_{\mathbf{y}})}$ [31], and $\sigma^2_{\mathbf{b}}$ and $\sigma^2_{\mathbf{y}}$ are, respectively, the variances of the correlation noise and of the encoder projections. The approximation comes from the assumption that $\sigma^2_{\mathbf{b}} \ll \sigma^2_{\mathbf{y}}$ [21]; the rate overhead ΔR with respect to the Wyner-Ziv bound in our experiments is approximately equal to 0.6 bit/sample when the feedback channel is adopted, and the length m of the vector $\mathbf{y}$ is in the $[100, 1000]$ range.

When Wyner-Ziv decoding succeeds, the decoder reconstructs a quantized version $\hat{\mathbf{y}}$ of the original random projections $\mathbf{y}$. The approximate mean squared error $\widehat{\mathrm{MSE}}(\mathbf{x}, \tilde{\mathbf{x}})$ between the original and the tampered decimated image can be obtained by just the random projections, in virtue of the RIP (Section II-B):

$$\begin{aligned} \sigma^2_{\mathbf{b}} &= \frac{1}{m}\|\mathbf{b}\|_2^2 = \frac{1}{m}\|\mathbf{A\Phi e}\|_2^2 = \frac{1}{m}\|\mathbf{Ae}\|_2^2 \\ &\approx \frac{1}{n}\|\mathbf{e}\|_2^2 = \frac{1}{n}\|\tilde{\mathbf{x}} - \mathbf{x}\|_2^2 = \widehat{\mathrm{MSE}}(\mathbf{x}, \tilde{\mathbf{x}}) \end{aligned} \tag{15}$$

where the second equality follows from the orthonormality of $\mathbf{\Phi}$ and the approximate equality is related to the RIP.

IV. Parameter selection

When the content creator generates the hash, the following parameters need to be determined: the number of random projections m, the quantization step size Δ and, if no feedback channel is available, the maximum MSE distortion between the original and the tampered image, $\sigma^2_{\mathbf{b}}$, for which tamper identification can be attempted. In this section, we consider m and Δ, while $\sigma^2_{\mathbf{b}}$ influences the rate allocation, as explained in Section V.

In order to show the effect of the parameters on the system performance, we conducted Montecarlo simulations for two scenarios: sparse tampering and compressible tampering.

A. Sparse tampering

In this scenario, we consider a tampering signal $\mathbf{e} \in \mathbb{R}^n$ which is exactly sparse. This scenario applies, for example, to cases where the tamper affects a limited region of the image. We define the sparsity ratio k/n, where k denotes the number of nonzero entries of the vector $\mathbf{e}$. Let $\hat{\mathbf{e}}$ denote the estimated vector $\mathbf{e}$. In this scenario, we measure the performance of the system in terms of the probability of detecting the tampering, P_D, for a given false positive rate $\boldsymbol{\alpha}$:

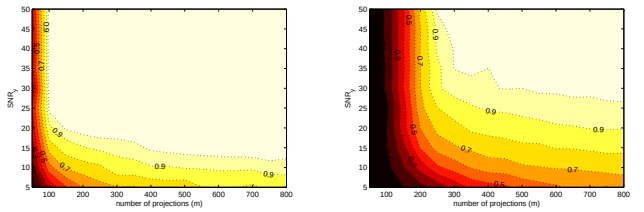
$$P_D = \Pr\{|\hat{e}(j)| > \tau \text{ and } |e(j)| > 0\} \tag{16}$$

$$\boldsymbol{\alpha} = \Pr\{|\hat{e}(j)| > \tau \text{ and } |e(j)| = 0\}, \tag{17}$$

where $\hat{e}(j)$, $j = 1 \ldots n$ is the j-th element of the estimated tampering vector $\hat{\mathbf{e}}$. The threshold τ has been adjusted so as to achieve $\boldsymbol{\alpha} < 10^{-3}$. Figure 4(a) and Figure 4(b) show the level sets of the probability of detection for k/n equal to 0.01 and 0.05 respectively, averaged over 100 realizations. The quantization of the projections is measured in terms of $SNR_{\mathbf{y}}$, which depends directly on the step size Δ:

$$SNR_{\mathbf{y}} = 10\log_{10}\frac{\sigma^2_{\mathbf{y}}}{D} \simeq 10\log_{10}\frac{12\sigma^2_{\mathbf{y}}}{\Delta^2} \tag{18}$$

We notice that, when the random projections are finely quantized (i.e. $SNR_y > 30$ dB), the probability of detection approaches 1 if the number of measurements is large enough with respect to the tamper sparsity. This behavior is similar to the case of noiseless compressive sensing, and (almost) perfect reconstruction of a sparse signal is achieved when $m > Ck\log_2(n/k)$. Conversely, when coarse quantization is employed, the probability of detection stays below 1, even when a large number of measurements is available. Therefore, as a rule of thumb, the quantization step size Δ should be chosen in such a way to attain $SNR_{\mathbf{y}} > 30$ dB, and the number of measurements m proportional to the expected number of nonzero coefficients k.



(a) $k/n = 0.01$ (b) $k/n = 0.05$

Fig. 4. Probability of detection (P_D) as a function of number of projections m and of the measure quantization SNR ($SNR_{\mathbf{y}}$) for the sparse tampering scenario. The contour levels identify the places of equal probability of detection, and span uniformly the interval $[0, 1]$ with a fixed step equal to 0.1.

B. Compressible tampering

In many cases, the tampering cannot be represented as an exactly sparse signal. This is the case, for example, of the brightness adjusted image in Figure 2, where the modification is global and cannot be described with a small set of non-zero coefficients. Compressible signals can be used to model this kind of tampering, as a generalization of exactly sparse signals. In compressible signals, a large fraction of the energy is concentrated in a small number of samples. For the purpose of Montecarlo simulations, we generate $\mathbf{e}$ by random permutation and random sign multiplication of the following signal [29]:

$$f_i = c|i|^{-\frac{1}{p}}, \quad i = 1, \ldots, n \tag{19}$$

where c is a constant that denotes a gain term, and p is the parameter that determines the "compressibility" of the signal.

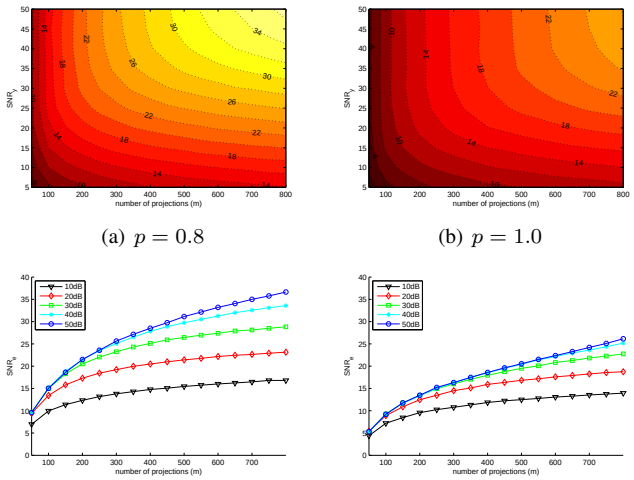
In this scenario, the performance of the system is determined based on the quality of the reconstructed tampering $\mathbf{e}$.

$$SNR_{\mathbf{e}} = 10 \log_{10} \frac{\|\mathbf{e}\|_2^2}{\|\mathbf{e} - \hat{\mathbf{e}}\|_2^2} \tag{20}$$

Figure 5(a) and Figure 5(b) show the value of $SNR_{\mathbf{e}}$ as a function of the number of projections m and of the quantization distortion of the measurements ($SNR_{\mathbf{y}}$), for p equal to 0.8 and 1.0 respectively. Unlike the sparse scenario, we observe a graceful improvement of the performance by increasing either m or $SNR_{\mathbf{y}}$. For the same values of the parameters, the SNR of the reconstructed signal is higher for more compressible signals ($p = 0.8$). In order to gain further insight, Figure 5(c) and Figure 5(d) show the same results under a different perspective, where we notice a (almost) linear dependency between $SNR_{\mathbf{e}}$ and $\log_{10} m$, for a fixed value of SNR_y. This suggests that the MSE distortion of the reconstructed tampering can be expressed as

$$\frac{\|\mathbf{e} - \hat{\mathbf{e}}\|_2^2}{n} \propto m^{-\beta} \tag{21}$$

In [32] it is shown that the exponent β is larger for the case of noiseless projections, indicating a faster decay of the reconstruction error when the number of projections is increased, with respect to the case of noisy projections. In our simulations, we have verified that β varies approximately between 0.7 and 1.5 for $p = 1$ and between 0.7 and 2.2 for $p = 0.8$. Also, in both cases the value of β tends to saturate



(a) $p = 0.8$ (b) $p = 1.0$

(c) $p = 0.8$ (d) $p = 1.0$

Fig. 5. Reconstruction performance for the compressible tampering scenario. (a) and (b) SNR of the reconstructed tampering ($SNR_{\mathbf{e}}$) in dB. Depicted in the graphs are the curve levels for $SNR_{\mathbf{e}}$. (c) and (d) $SNR_{\mathbf{e}}$ as a function of the number of projections for different quantization levels of the measurements $\mathbf{y}$. Each curve corresponds to different $SNR_{\mathbf{y}}$ between 10 and 50 dB.

when $SNR_{\mathbf{y}}$ is larger than 30 dB. Therefore, similarly to the sparse tampering scenario, the quantization step size Δ should be chosen in such a way to attain $SNR_{\mathbf{y}} > 30$ dB.

V. Rate allocation

In Section III-A we have shown that the correlation model between the original and the tampered random projections can be written as

$$\tilde{\mathbf{y}} = \mathbf{y} + \mathbf{b} \tag{22}$$

Hereafter we assume that $\mathbf{y}$ and $\mathbf{b}$ are statistically independent: this hypothesis is reasonable since the hash is computed at the content producer side, where no information about the actual tampering is available. Therefore, this represents a worst-case scenario from a rate allocation point of view.

Let $j = 1, \ldots, J$ denote the bitplane index and R_j the bitrate (in bits/symbol) needed to decode the j-th bitplane. As detailed in the previous section, the probability density function of $\mathbf{y}$ and $\mathbf{b}$ can be well approximated to be zero mean Gaussian, respectively with variance $\sigma_{\mathbf{y}}^2$ and $\sigma_{\mathbf{b}}^2$. The rate estimation algorithm receives in input the source variance $\sigma_{\mathbf{y}}^2$, the correlation noise variance $\sigma_{\mathbf{b}}^2$, the quantization step size Δ and the number of bitplanes to be encoded J and returns the average number of bits needed to decode each bitplane $R^j, \quad j = 1, \ldots, J$. The value of $\sigma_{\mathbf{y}}^2$ can be immediately estimated from the random projections at the time of hash generation. The value of $\sigma_{\mathbf{b}}^2$ is set to be equal to the maximum MSE distortion between the original and the tampered image, for which tampering identification can be attempted.

The rate allocated to each bitplane is given by:

$$R^j = H(\mathbf{y}^j | \tilde{\mathbf{y}}, \mathbf{y}^{j-1}, \mathbf{y}^{j-2}, \ldots, \mathbf{y}^1) + \Delta R^j \quad \text{[bits/sample]} \tag{23}$$

where $\mathbf{y}^j$ denotes the j-th bitplane of $\mathbf{y}$. In fact LDPC decoding of bitplane j exploits the knowledge of the real-valued side information $\tilde{\mathbf{y}}$ as well as previously decoded

bitplanes $\mathbf{y}^{j-1}, \mathbf{y}^{j-2}, \ldots, \mathbf{y}^1$. Since we use nonideal channel codes with a finite sequence length m to perform source coding, a rate overhead ΔR^j is added to Shannon's lower bound. Experimentally, we determined that a value $\Delta R^j = 0.15$ [bits/sample] is a suitable choice. This overhead depends on the type of channel codes adopted and has been estimated experimentally. The integral needed to compute the value of the conditional entropy in (23) is factored out in detail in our previous work [33]. For completeness we summarize the most relevant steps in Appendix III.

VI. Geometric transformations

As explained in the introduction, an image hashing system is supposed to be secure to forgery attacks as well as robust to content-preserving modifications such as moderate geometric transformations (e.g. cropping, scaling, rotation, etc.). It is possible that both the situations occur, i.e. the image is first geometrically transformed and then maliciously tampered with (or vice versa): in this case, before applying the tampering identification algorithm described in Section III-C, the image needs to be registered with respect to the original one.

The recent works on compressive detection and classification [20][34][19] come in help by providing a maximum-likelihood estimation framework based on the same principles of universality of the measurements adopted in CS reconstruction. Different from the basic CS setting (see Section II-B), where the goal is to reconstruct a signal given a small number of random incoherent projections, compressive classification aims at estimating some parameters of the signal rather than perfectly recovering the original. The key point is that a set of images of the same scene with different imaging parameters (translation, scaling, viewing angle, illumination, etc.) forms a low-dimensional, nonlinear manifold in the high-dimensional ambient image space. If the manifold is sufficiently smooth, projecting each point of the n-dimensional manifold to an m-dimensional subspace by taking random measures as explained in Section II-B preserves the essential structure of the manifold, provided that a sufficient number of measures are taken. The number of measurements m needed to produce a stable manifold embedding depends linearly on the dimension K of the manifold. When m is chosen in this way, the RIP (15) holds and distances are preserved in the embedding. This fact enables the registration of an image working directly in the low-dimensional projection space, by basically computing cross-correlation over the embedded manifold. The low-dimensional manifold is sampled by extracting random measurements from the received image, each time transformed by the warping function f using a different set of parameters $\boldsymbol{\theta}_i$. Then, the distance between the projections reconstructed with the hash and the test projections is computed, and the parameters that produce the minimum distance are used for registration. This technique has been named *smashed filter* in [20], and is detailed in the following.

Let $\bar{\mathbf{x}} = f(\bar{\mathbf{x}}; \theta)$ denote the vector obtained by rasterizing the decimated, geometrically transformed, tampered image. The vector $\boldsymbol{\theta} \in \mathbb{R}^K$ denotes the parameters of the transformation. Here we assume that the inverse transformation $f^{-1}(\cdot)$ exists, although this is only partially true in practice. The interpretation of $\boldsymbol{\theta}$ is obvious for the case of simple geometrical transformations. For cropping, $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ represents the horizontal and vertical coordinates of the top-left corner of the cropped selection (the size of the bounding box is inferred from the received image). In the case of rotation by an angle α, $\theta = \alpha$, etc. The registration is performed in two steps according to a multiresolution approach, in order to reduce the computational cost. First, a coarse registration is applied, which works only on the decimated representation of the tampered image $\bar{\mathbf{x}}$. Then, the registration refinement is carried out, considering the full resolution tampered image $\bar{\mathbf{X}}$. A schematic view of the algorithm is given below, while Figure 6 summarizes graphically the registration procedure.

1) Coarse registration:

- Initialization: Set $MSE^C = \infty$, $\hat{\boldsymbol{\theta}}^C = \emptyset$ and the set of candidate parameter vectors $\Theta^C = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L^C}\}$
- For each candidate vector $\boldsymbol{\theta}_i \in \Theta^C$:
  - Compute the warped decimated image vector $\tilde{\mathbf{x}}_{\boldsymbol{\theta}_i} = f^{-1}(\bar{\mathbf{x}}, \boldsymbol{\theta}_i)$
  - Compute the random projections $\tilde{\mathbf{y}}_{\boldsymbol{\theta}_i} = \mathbf{A}\tilde{\mathbf{x}}_{\boldsymbol{\theta}_i}$
  - Wyner-Ziv decode the vector $\hat{\mathbf{y}}$ using $\tilde{\mathbf{y}}_{\boldsymbol{\theta}_i}$ as side information.
  - If Wyner-Ziv decoding fails,
    - proceed to the next one $\boldsymbol{\theta}_{i+1}$ and set $MSE_i = +\infty$
  - Else, compute $MSE_i = \frac{||\hat{\mathbf{y}} - \tilde{\mathbf{y}}_{\boldsymbol{\theta}_i}||_2^2}{m}$.
    - If $MSE_i < MSE^C$, then $\hat{\boldsymbol{\theta}}^C = \boldsymbol{\theta}_i$ and $MSE^C = MSE_i$

2) Registration refinement:

- Initialization: Set $MSE^R = \infty$, $\hat{\boldsymbol{\theta}}^R = \emptyset$ and the set of candidate parameter vectors $\Theta^R = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{L^R}\}$, which depend on $\hat{\boldsymbol{\theta}}^C$
- For each candidate vector $\boldsymbol{\theta}_i \in \Theta^R$:
  - Compute the warped full resolution image $\tilde{\mathbf{X}}_{\boldsymbol{\theta}_i} = f^{-1}(\bar{\mathbf{X}}, \boldsymbol{\theta}_i)$.
  - Compute the decimated version of the warped image $\tilde{\mathbf{x}}_{\boldsymbol{\theta}_i}$ from $\tilde{\mathbf{X}}_{\boldsymbol{\theta}_i}$.
  - Compute the random projections $\tilde{\mathbf{y}}_{\boldsymbol{\theta}_i} = \mathbf{A}\tilde{\mathbf{x}}_{\boldsymbol{\theta}_i}$
  - Wyner-Ziv decode the vector $\hat{\mathbf{y}}$ using $\tilde{\mathbf{y}}_{\boldsymbol{\theta}_i}$ as side information.
  - If Wyner-Ziv decoding fails,
    - proceed to the next one $\boldsymbol{\theta}_{i+1}$ and set $MSE_i = +\infty$
  - Else, compute $MSE_i = \frac{||\hat{\mathbf{y}} - \tilde{\mathbf{y}}_{\boldsymbol{\theta}_i}||_2^2}{m}$.
    - If $MSE_i < MSE^R$, then $\hat{\boldsymbol{\theta}}^R = \boldsymbol{\theta}_i$; $MSE^R = MSE_i$

The registration algorithm presented above holds for any type of geometric transformation model that can be expressed in terms of a finite number of parameters, summarized by the vector $\boldsymbol{\theta} \in \mathbb{R}^K$. For more complex transformations including, for example, affine or perspective warping, the number of parameters K is larger, thus making the search for the best matching model computationally expensive. In fact, in these cases the number of candidates L_C and L_R to be explored during, respectively, the coarse and refinement phases might

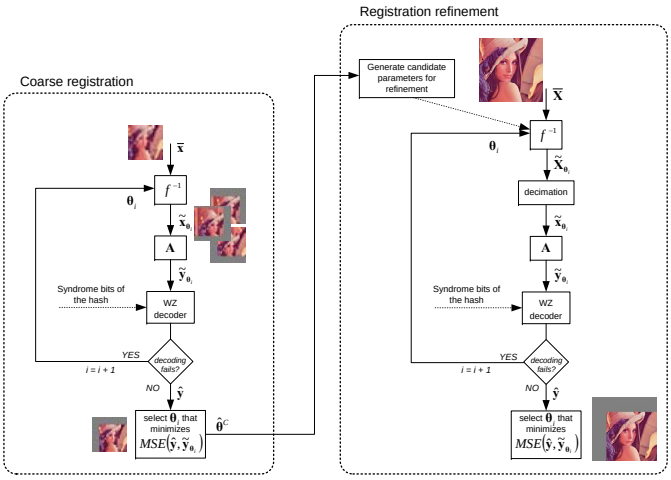
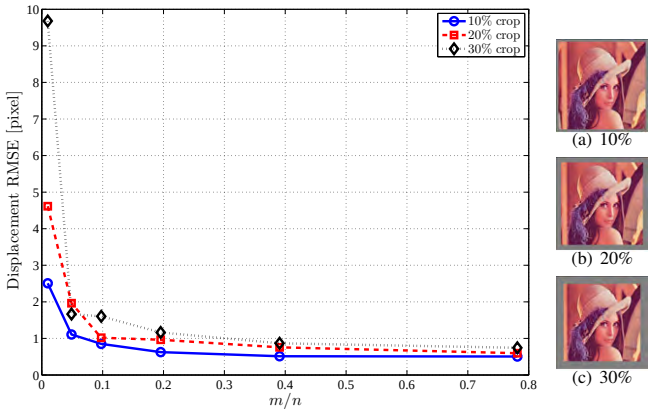


Fig. 6. Coarse image registration and refinement. Image thumbnails are proof of concepts for the case of a cropping transformation, for the Lena image. Note that the coarse and the refinement registration procedures substantially perform the same kind of search, with the only difference that in the second case the search (for the case of cropping) is carried out at the pixel granularity.



(a) Displacement estimation error

Fig. 7. Robustness towards different levels of cropping. Parts (b)-(d) give an insight of the amount of the effective cropping on the Lena image.

(a) Tampered (29.1 dB) (b) True tamper (c) Detected (pixel) (d) Detected (DCT) (e) Detected (Haar)

Fig. 8. A tampering example: logo insertion.

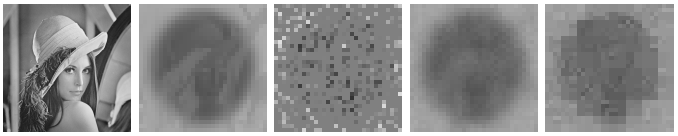
be extremely large. The definition of fast search strategies able to explore such a vast parameter space is left to future investigations.

The robustness against geometric transformations in influenced by two factors. On the one hand, the rate allocation employed to avoid the use of feedback channel poses a limit on the maximum MSE of the transformed image w.r.t. the original (the MSE of the registered image, MSE^R, cannot be larger than the maximum distortion $\sigma^2_{\mathbf{b}}$ hypothesized to allocate the rate for the hash). On the other hand, the number of projections influences the quality of the image registration process. However, the number of measurements required for image registration is in general much smaller than the number of projections required for completely reconstructing the tampering, and thus using an m-dimensional measurement vector generally suffices for the task of image registration. Figure 7 shows the average accuracy of image registration in terms of RMSE (Root Mean Square Error), expressed in pixel, for different m/n ratios, when the geometrical transformation used is cropping. The adopted test image is Lena, with a spatial resolution of 512×512 pixels; the cropping has been carried out by extracting from the original picture 100 cropped selections of different areas (respectively, 70, 80 and 90% of the original surface) with randomly chosen $\boldsymbol{\theta} = [\theta_1, \theta_2]$ top-left corners. In the case of small cropping (80% of the original area retained, i.e. 20% cropping), using $m/n = 0.2$ it is possible to register the image w.r.t. the original with a sub-pixel accuracy. This error could be further reduced by adding to the system some local maximum interpolation (e.g. parabolic interpolation) in the refinement step. If some tampering is introduced to the original picture besides cropping, the results in Figure 7 do not change significantly. This validates the fact that much less measurements are in general required for image registration rather than for tampering reconstruction.

VII. Experimental results and discussion

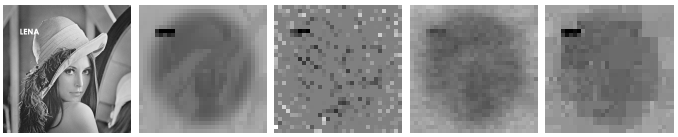
In this section we provide a walk-through of the proposed system by means of an example. We discuss the results obtained on a single image, Lena. In fact, we want to point out that the performance of the algorithm does not depend on the actual image content, but rather on the way the image has been tampered with. Therefore, instead of applying the same manipulation to a set of images, we prefer to consider a diverse set of manipulations including logo insertion, brightness adjustment, cropping and JPEG compression. We consider the luminance component only, at resolution $N = 512 \times 512$. The block size B is equal to 16: this turns out to be a good compromise between the accuracy of localization and the bit-rate of the hash. The decimated vector $\mathbf{x}$ thus contains $n = 1024$ elements. The content producer generates a hash by computing $m = 450$ random projections and quantizes them with a step size $\Delta = 10$ in order to achieve $SNR_{\mathbf{y}} = 33$ dB, according to the guidelines presented in Section IV. When no feedback channel is available, rate allocation is performed by setting the maximum expected distortion $\sigma^2_{\mathbf{b}} < 650$, which approximately corresponds to a maximum tolerated image distortion on the decimated image of 20 dB in terms of PSNR. The resulting hash has size 0.0077 bits/pixel (4.5 bits/projection). In order to demonstrate the universality of the hash construction, the same hash is used to detect several different tampering attacks.

Figure 8 shows the case of a logo added to the original image. The PSNR between the original and the tampered image is equal to 29.1 dB. The actual tampering signal, which is depicted in Figure 8(b), is exactly sparse in the pixel domain with $k/n = 0.031$. Figure 8(c)-8(e) show the estimated



(a) Tampered (25.2 dB) (b) True tamper (c) Detected (pixel) (d) Detected (DCT) (e) Detected (Haar)

Fig. 9. A tampering example: brightness adjustment.



(a) Tampered (23.7 dB) (b) True tamper (c) Detected (pixel) (d) Detected (DCT) (e) Detected (Haar)

Fig. 10. A tampering example: logo insertion on a brightness adjusted image.

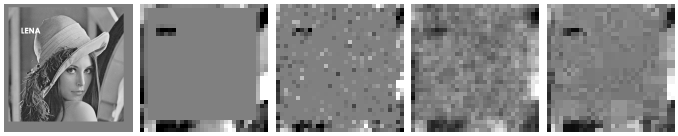
tampering, obtained by solving the optimization problem (10) with different bases $\mathbf{\Phi}_i$. In this particular example, we tried the reconstruction in the pixel, DCT or Haar wavelet domain by setting the matrix $\mathbf{\Phi}_i$ equal, respectively, to the identity, DCT or Haar basis matrices. Table I shows the ℓ_1 norm of the reconstructed tampering $\hat{e}$ in the three domains. As explained in Section III-C, we are finally interested in recovery the sparsest (in the ℓ_0 sense) reconstruction, and the ℓ_1 norm is a good proxy for this task. Therefore, we select the sparsest reconstruction as the one with the minimum ℓ_1 norm: accordingly, in this case we deem the reconstruction in the pixel domain as the best one. If a feedback channel is available, and hash bits could be received upon request, the hash size would be smaller and equal to 0.0056 bits/pixel (3.3 bits/projection). For comparison, avoiding Wyner-Ziv coding and performing direct uniform scalar quantization of the random projection would require 0.010 bits/pixel (5.9 bit/random projection). Due to the relatively high sparsity level in this specific example, a good reconstruction of the tampering map in the pixel domain could have been possible with as little as $m = 150$ random projections, thus reducing the size to 0.0025 bits/pixel (or 0.0018 bits/pixel if a feedback channel is available).

Figure 9 illustrates another modification to the original image content, consisting of a spatially-adaptive brightness adjustment. In this case, the PSNR of the tampered image is equal to 25.2 dB. As before, we illustrate the reconstructed tampering maps in the pixel, DCT and Haar wavelet domains. The tampering signal is compressible, rather than exactly

TABLE I

l_1 NORM OF THE RECONSTRUCTED TAMPERING SIGNAL ($\|\hat{\mathbf{e}}\|_1$) IN THE TESTED BASES.

Tampering	pixel	DCT	Haar
Logo	**460**	625	1335
Brightness adj.	4727	**1548**	2606
Logo + Brighness adj.	4829	**2499**	3150
Logo + Crop	5943	5831	**5356**
JPEG	185	239	**160**



(a) Tampered (20.6 dB) (b) True tamper (c) Detected (pixel) (d) Detected (DCT) (e) Detected (Haar)

Fig. 11. A tampering example: logo insertion on a cropped image.

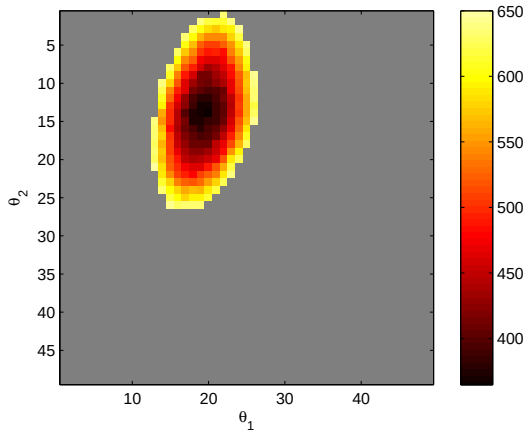


Fig. 12. Error surface explored during image registration. MSE vs. θ. The gray area corresponds to MSE values for which WZ decoding fails.

sparse. The reconstruction in the pixel domain is not a good approximation of the original, since the number of random projections of the hash is too low compared to the one required by the sparsity of the tampering in that basis. On the contrary, recovering the tampering in the Haar wavelet and especially in the DCT domain (which achieves the minimum l_1 norm of the reconstructed tampering signal) produces much more meaningful results. This can be easily justified by observing the smooth behavior of the tampering in this case, which can therefore be represented by few low-frequency coefficients.

A more challenging example is shown in Figure 10, where a logo has been inserted on a brightness adjusted image, obtaining a PSNR equal to 23.7 dB. As before, the tampering signal is compressible, and reasonable reconstructions are obtained both in the Haar wavelet and DCT domain. We argue that this scenario could benefit from the use of redundant dictionaries, where different atoms are used to represent local singularities (the logo) or (piecewise) smooth signals (brightness adjustment). We emphasize the fact that the hash construction is future-proof, i.e. a better reconstruction of the tampering signal could be obtained from the same hash, if a better sparsifying basis or redundant dictionary is discovered after the hash generation.

Finally, Figure 11 demonstrates the robustness of the proposed system to moderate cropping. The cropped image retains 80% of the original area, and the PSNR of the padded and registered cropped image (shown in Figure 11(a)) with respect to the original is equal to 20.65 dB. First, the system employs the decoded random projections to perform image registration. In this example, the top-left corner of the cropped image is at $\boldsymbol{\theta} = [13, 18]$. The coarse search returns the estimate

(a) Tampered (29.4 dB) (b) True tamper (c) Detected (pixel) (d) Detected (DCT) (e) Detected (Haar)

Fig. 13. A tampering example: JPEG compression.

$\hat{\boldsymbol{\theta}}^C = [16, 16]$, which is further refined to $\hat{\boldsymbol{\theta}}^R = [13, 18]$ when working at full resolution. Figure 12 illustrates the error surface explored during the refinement step, where each point represents the value of the MSE between $\mathbf{y}_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{y}}$ for a given value of the parameter $\boldsymbol{\theta}$. We notice that Wyner-Ziv decoding fails for values of the parameter $\boldsymbol{\theta}$ such that the MSE would exceed the threshold $\sigma^2_{\mathbf{b}} = 650$. Conversely, when Wyner-Ziv coding succeeds, the MSE can be computed and minimized. Tampering identification is carried out after image registration. In this example, the best reconstruction is achieved using the Haar wavalet as sparsifying basis. We notice that the system treats as tampering also the missing portions of the image removed by the cropping operation, thus decreasing the actual sparsity (or compressibility) of the tampering signal.

To conclude, we illustrate an example to show the limitations of the proposed system when the tampering introduced in the image does not have a sparse representation in any basis or redundant dictionary. Figure 13 shows the same image compressed using JPEG (PSNR = 29.4 dB). By construction, the algorithm searches for the sparsest representation in a given basis consistent with the constraints imposed by the hash. None of the previous bases succeeds in representing the tampering signal correctly. Therefore, we argue that the proposed system is semi-automatic, in the sense that human inspection of the automatically selected tampering map is needed to validate the estimated modification. The automatic detection of this scenario will be subject of future research investigations.

VIII. Conclusions

Multimedia hashes are an effective tool for authenticating legitimate copies of images and for identifying possible manipulations, which may have been introduced in the picture to alter its semantic content. In many applications, it is also desirable to deal with small hashes to account for the bandwidth/storage available budget. In this paper, we have described an image hashing system that targets both these two conflicting requirements: namely, we encode the hash with a Wyner-Ziv codec to reduce the bit-length of the proposed hash, and we employ compressive sensing tools to obtain an estimate of the tampering at the content user side. The core assumption of our system is that the tampering actually has some *structure*, a postulate that in our setting is intended as the requirement that the manipulation admits a sparse (or compressible) representation in some basis. This turns out to be the case for many meaningful content-changing modifications, provided that we find an adequate dictionary in which to represent them. The most prominent feature of the proposed system is perhaps its *universality*: the proposed hash is build upon a set of pseudo-random projections of the decimated image coefficients, and from the same hash the user can reconstruct the tampering in different, several dictionaries; if a new basis has to be tested, no changes need to be carried out on the hash signature. In addition, using recent results on compressive classification, the produced hash is also robust to moderate geometrical transformation, which can be inverted through a simple image registration procedure: for example, we are able to tolerate up to 20% of cropping without increasing the hash bit-length.

A critical aspect of the proposed system regards whether it could be used in a completely automatic fashion. While we are able to provide the user with different estimates of the tampering (one for each basis at the user's disposal), we cannot produce an ultimate decision about whether the manipulation preserves or not the semantic content of the original picture. We prefer instead to leave this choice to the end user, who can take advantage of the produced results to finalize his decision. Again, the content legitimacy of a copy is intimately related to the structure of the attack, thus a promising road towards a partial understanding of the authenticity of the received picture passes through the knowledge of the sparsity of the actual tampering from its reconstruction. To the authors' knowledge, this is still an open issue in CS theory, especially when the attack is compressible rather than sparse. Future research will investigate how to adapt the system to the unknown sparseness pattern, e.g. by modifying the system architecture embedding some feedback channel to acquire progressively more measurements.

Another important aspect regards robustness to geometrical transformations. As outlined in Section VI, the procedure that we propose for registering the image content in the compressed domain is computationally demanding, as it actually implies a full search (or a hierarchical search, using a refinement step) of the parameter space. As is known, this kind of strategies suffer from the so-called "curse of dimensionality", i.e. the dimension of the search space increases exponentially with the number of parameters. This may seriously compromise the applicability of our approach with more complex transformations, thus we believe that finding some fast search strategy for these situations is a promising research direction.

Appendix I
Determination of error norm bound ϵ in the ℓ_1 reconstruction

The value of the error norm bound ϵ in (10) can be determined as follows.

$$\|\hat{\mathbf{b}} - \mathbf{A}\boldsymbol{\Phi}\mathbf{e}\|_2^2 = \|z\|_2^2 = \sum_{i=1}^{m} z_i^2 \tag{24}$$

Since z_i is the quantization noise introduced by a uniform scalar quantizer, we have $z_i \sim U[-\Delta/2, +\Delta/2]$ [35]. Therefore, we can write [36]:

$$E[\|z\|_2^2] = m\frac{\Delta^2}{12} \tag{25}$$

$$\text{std}[\|z\|_2^2] = \sqrt{m}\frac{\Delta^2}{6\sqrt{5}} \tag{26}$$

Hence, we can adopt the following upper bound on ϵ:

$$\epsilon^2 \leq m\frac{\Delta^2}{12} + \lambda\sqrt{m}\frac{\Delta^2}{6\sqrt{5}} \tag{27}$$

where λ has been set equal to 2 in our system.

Appendix II
Determination of statistical properties of $\mathbf{b}$, $\mathbf{y}$ and $\tilde{\mathbf{y}}$

Let the tampering signal $\mathbf{t} = [t_1, \ldots, t_n]^T$ be modeled as the realization of a real-valued wide-sense stationary signal with mean $\mu = E[t_i]$ and autocorrelation sequence $r_k = E[t_i t_{i-k}]$. Since $\mathbf{b} = A\mathbf{t}$, the first order statistics of $\mathbf{b}$ can be obtained as follows:

$$E_{\mathbf{t}}[b_i] = E_{\mathbf{t}}[\sum_{j=1}^{n} a_{ij}t_j] = \sum_{j=1}^{n} a_{ij}E_{\mathbf{t}}[t_j] = \mu\sum_{j=1}^{n} a_{ij}. \tag{28}$$

where $E_{\mathbf{t}}[\cdot]$ denotes the expectation of the realizations of the tampering signal $\mathbf{t}$. By taking expectation also over the rows of the matrix A, $E_A[\cdot]$, we obtain:

$$E_A[\mu\sum_{j=1}^{n} a_{ij}] = \mu\sum_{j=1}^{n} E_A[a_{ij}] = 0, \tag{29}$$

where the last equality follows from the construction of the matrix A.

As for the second order statistics, we can write:

$$\begin{aligned} E_{\mathbf{t}}[b_i b_j] &= E_{\mathbf{t}}[a_i^T\mathbf{t}\cdot a_j^T\mathbf{t}] = E_{\mathbf{t}}[a_i^T\mathbf{t}\mathbf{t}^T a_j] \\ &= a_i^T E[\mathbf{t}\mathbf{t}^T]a_j = \sum_{l=1}^{n}\sum_{k=1}^{n} r_{l-k}a_{il}a_{jk}. \end{aligned} \tag{30}$$

By taking expectation also over the rows of the matrix A, we obtain:

- if $i \neq j$

$$\begin{aligned} E_A[\sum_{l=1}^{n}\sum_{k=1}^{n} r_{l-k}a_{il}a_{jk}] &= \sum_{l=1}^{n}\sum_{k=1}^{n} r_{l-k}E_A[a_{il}a_{jk}] \\ &= \sum_{l=1}^{n}\sum_{k=1}^{n} r_{l-k}E_A[a_{il}]E_A[a_{jk}] = 0 \end{aligned} \tag{31}$$

- if $i = j$

$$\begin{aligned} E_A[\sum_{l=1}^{n}\sum_{k=1}^{n} r_{l-k}a_{il}a_{ik}] &= \sum_{l=1}^{n}\sum_{k=1}^{n} r_{l-k}E_A[a_{il}a_{ik}] \\ &= nr_0E_A[a_{il}^2] = r_0 \end{aligned} \tag{32}$$

Therefore we can conclude that, regardless of the statistics of the vector $\mathbf{t}$, the vector $\mathbf{b}$ is a realization of a zero-mean wide-sense stationary random process with diagonal autocorrelation matrix [37]. In addition, by the central limit theorem, when n is large the elements of the vector $\mathbf{b}$ are samples of a Gaussian distribution, since each element is the sum of (arbitrarily distributed) random variables. Hence, the vector $\mathbf{b}$ is a realization of a zero-mean i.i.d. Gaussian random sequence. Using the same arguments, it is possible to show that also $\mathbf{y}$ and $\tilde{\mathbf{y}}$ are i.i.d. Gaussian.

Appendix III
Determination of the conditional entropy for rate allocation

For the sake of clarity, we adopt the common notation used in the distributed source coding literature as introduced in Section II-A. The random projections $\mathbf{y}$ obtained from the original image play the role of the source X, while the projections $\tilde{\mathbf{y}}$ computed from the tampered image that of the side information Y. The correlation noise $\mathbf{b}$ is denoted with the random variable Z. Uppercase letters indicate random variables, while lowercase letters are used for scalar values.

The Shannon's lower bound on the number of bits is given by [31]

$$R^j \geq H(X^j|Y, X^{j-1}, X^{j-2}, \ldots, X^1) \quad \text{[bits/sample]}, \tag{33}$$

where X^j denotes the j-th bitplane of the source X. In fact LDPC decoding of bitplane j exploits the knowledge of the real-valued side information Y as well as previously decoded bitplanes $X^{j-1}, X^{j-2}, \ldots, X^1$. The value of R^j from equation (33) can be readily computed by numerical integration. In fact, the expression of the entropy in (33) can be written as

$$\begin{aligned} &H(X^j|Y, X^{j-1}, X^{j-2}, \ldots, X^1) = \\ &\quad = \sum_{q=1}^{2^{j-1}} p(q)H(X^j|Y, Q=q) \\ &= \sum_{q=1}^{2^{j-1}} p(q)\int_{-\infty}^{+\infty} p_{Y|q}(y)H(X^j|Y=y, Q=q)dy, \end{aligned} \tag{34}$$

where Q denotes the quantization bin index obtained decoding up to bitplane $j-1$. The value of $H(X^j|Y=y, Q=q)$ represents the entropy of the binary source X^j when the side information assumes the specific value y and the source X is known to be within quantization bin q, i.e.,

$$H(X^j|Y=y, Q=q) = -p_0\log_2 p_0 - (1-p_0)\log_2(1-p_0), \tag{35}$$

and

$$\begin{aligned} p_0 &= \Pr\{X^j = 0|Y=y, Q=q\} \\ &= \frac{\int_{L_q}^{(L_q+U_q)/2} p_X(x)p_Z(x-y)dx}{\int_{L_q}^{U_q} p_X(x)p_Z(x-y)dx}, \end{aligned} \tag{36}$$

where L_q and U_q are the lower and upper thresholds of the quantization bin q.

The expression $p_{Y|q}(y)$ in (34) represents the marginal distribution of Y, conditioned on $X \in q$, and it can be obtained from the knowledge of the joint distribution $p_{XY}(x,y)$

$$p_{Y|q}(y) = \frac{\int_{L_q}^{U_q} p_{XY}(x,y)dx}{\int_{L_q}^{U_q} p_X(x)dx} = \frac{\int_{L_q}^{U_q} p_X(x)p_Z(x-y)dx}{\int_{L_q}^{U_q} p_X(x)dx}, \tag{37}$$

where the second equality follows from the additive correlation model $Y = X + Z$. Finally, equation (34) can be evaluated by means of numerical integration over y.

REFERENCES

[1] H. Farid, "Exposing digital forgeries in scientific images," in *ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006.

[2] M.K. Johnson and H. Farid, "Detecting photographic composites of people," in *6th International Workshop on Digital Watermarking*, Guangzhou, China, 2007.

[3] M.K. Johnson and H. Farid, "Exposing digital forgeries through specular highlights on the eye," in *9th Int. Workshop on Information Hiding*, Saint Malo, France, 2007.

[4] M.K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *ACM Multimedia and Security Workshop*, Geneva, Switzerland, 2006.

[5] J. Fridrich, "Image watermarking for tamper detection," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, USA, 1998.

[6] J.J. Eggers and B. Girod, "Blind watermarking applied to image authentication," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, USA, 2001.

[7] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1167–1180, 1999.

[8] P.W. Wong, "A public key watermark for image verification and authentication," in *Proc. IEEE Int. Conf. Image Processing*, Chicago, USA, 1998.

[9] M.U. Celik, G. Sharma, E. Saber, and A.M. Tekalp, "Hierarchical Watermarking for Secure Image Authentication With Localization," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 585, 2002.

[10] C.S. Lu, S.K. Huang, C.J. Sze, and H.Y.M. Liao, "Cocktail watermarking for digital image protection," *IEEE Trans. on Multimedia*, vol. 2, no. 4, pp. 209–224, 2000.

[11] R. Venkatesan, S.M. Koon, M.H. Jakubowski, and P. Moulin, "Robust image hashing," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, CA, 2000.

[12] S.S. Kozat, R. Venkatesan, and M.K. Mihcak, "Robust perceptual image hashing via matrix invariants," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, 2004.

[13] V. Monga, D. Vats, and B.L. Evans, "Image authentication under geometric attacks via structure matching," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Amsterdam, 2005.

[14] S. Roy and Q. Sun, "Robust Hash for Detecting and Localizing Image Tampering," in *Proc. IEEE Int. Conf. Image Processing*, S.Antonio, USA, 2007.

[15] A. Swaminathan, Yinian Mao, and Min Wu, "Robust and secure image hashing," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 215–230, June 2006.

[16] M. Johnson and K. Ramchandran, "Dither-based secure image hashing using distributed coding," in *Proc. IEEE Int. Conf. Image Processing*, 2003.

[17] Y.C. Lin, D. Varodayan, and B. Girod, "Image authentication based on distributed source coding," in *Proc. IEEE Int. Conf. Image Processing*, S.Antonio, USA, September 2007, vol. 3.

[18] Y.C. Lin, D. Varodayan, and B. Girod, "Spatial Models for Localization of Image Tampering Using Distributed Source Codes," in *Picture Coding Symposium*, Lisbon, Portugal, 2007.

[19] M.F. Duarte, M.A. Davenport, M.B. Wakin, J.N. Laska, D. Takhar, K.F. Kelly, and R.G. Baraniuk, "Multiscale random projections for compressive classification," in *Proc. IEEE Int. Conf. Image Processing*, S.Antonio, USA, 2007.

[20] M.A. Davenport, M.F. Duarte, M.B. Wakin, J.N. Laska, D. Takhar, K.F. Kelly, and R.G. Baraniuk, "The smashed filter for compressive classification and target recognition," in *Computat. Image. V*, San Jose, USA, 2007.

[21] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 1–10, 1976.

[22] B. Girod, A.M. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, 2005.

[23] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, "The DISCOVER Codec: Architecture, Techniques and Evaluation," *Picture Coding Symposium*.

[24] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Processing*, vol. 86, no. 11, pp. 3123–3130, 2006.

[25] E.J. Candés and M.B. Wakin, "An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, March 2008.

[26] E. Candés, "Compressive sampling," in *International Congress of Mathematicians*, Madrid, Spain, 2006.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[28] E. van den Berg and M. P. Friedlander, "In pursuit of a root," Tech. Rep. TR-2007-19, Department of Computer Science, University of British Columbia, June 2007, Preprint available at `http://www.optimization-online.org/DB_HTML/2007/06/1708.html`.

[29] E.J. Candes, M.B. Wakin, and S.P. Boyd, "Enhancing Sparsity by Reweighted ℓ_1 Minimization," *Preprint*, 2007.

[30] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed sensing and redundant dictionaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2210–2219, May 2008.

[31] T.M. Cover and J.A. Thomas, *Elements of information theory*, Wiley New York, 1991.

[32] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4036–4048, Sept. 2006.

[33] R. Bernardini, M. Naccari, R. Rinaldo, M. Tagliasacchi, S. Tubaro, and P. Zontone, "Rate allocation for robust video streaming based on distributed video coding," *Signal Processing: Image Communication*, vol. 23, no. 5, pp. 391–403, 2008.

[34] M.A. Davenport, M.B. Wakin, and R.G. Baraniuk, "Detection and estimation with compressive measurements," Tech. Rep., Tech. Rep. TREE0610, Rice University ECE Department, 2006.

[35] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Springer, 1992.

[36] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics*, 2005.

[37] A. Papoulis and S.U. Pillai, *Probability, random variables, and stochastic processes*, McGraw-Hill New York, 1965.