LOCALIZATION OF SPARSE IMAGE TAMPERING VIA RANDOM PROJECTIONS

M. Tagliasacchi, G. Valenzise, S. Tubaro

Dipartimento di Elettronica e Informazione – Politecnico di Milano,
P.za Leonardo da Vinci, 32, 20133 Milan – Italy
{tagliasa, valenzise, tubaro}@elet.polimi.it

ABSTRACT

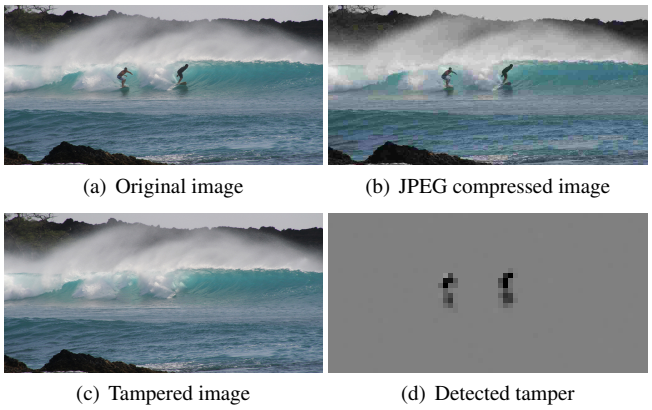
Hashes can be used to provide authentication of multimedia contents. In the case of images, a hash can be used to detect whether the data has been modified in an illegitimate way. When the authentication check fails, it might be useful to localize the tampering in the spatial domain. This paper proposes an algorithm based on compressive sensing principles, which solves both the authentication and the localization problems. The encoder produces a hash using a small bit budget by quantizing a limited number of random projections of the authentic image. The decoder uses the hash to estimate the distortion between the original and the received image. In addition, if the attack is sparse, it can be also localized. In order to keep the size of the hash small, encoding/decoding takes advantage of distributed source codes. This paper also investigates experimentally the tradeoff between the rate allocated to the hash and the performance achieved in terms of tampering localization.

***Index Terms*—** Image tampering, compressive sensing, distributed source coding

1. INTRODUCTION

The delivery of multimedia contents in peer-to-peer networks might give rise to different versions of the same multimedia object at different nodes. In the case of images, some versions might differ from the original because of processing due, for instance, to transcoding or bitstream truncation. In other cases, malicious attacks might occur by tampering part of the image and possibly affecting its semantic content. Two examples of these modifications are given in Figure 1, which shows two different versions of the same image (depicted in part (a) of the figure): 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 PSNR with respect to the original is equal to 31.5dB. In this paper we propose to add a small hash to the image bitstream. At the receiver, the information contained in the hash enables to: 1) estimate the distortion between the image and its original version; 2) localize the tampering, if the attack is sparse. 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 1.

Multimedia hashes for tampering localization have been recently explored in the literature. In [1], the authors propose a system that performs image authentication using distributed source codes. The hash consists of syndrome bits produced by LDPC 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. If decoding succeeds, the image is declared authentic. This scheme has been later extended in [2] to perform tampering localization, at the cost of extra syndrome bits. Another interesting algorithm is presented in [3], where the goal is to produce a multimedia hash that is robust to some legitimate image manipulations (JPEG encoding, cropping, scaling, rotation) and sensitive to illegal manipulations (like image tampering). The hash consists of two parts: the first part for authentication only, based on quantized local regional descriptors, which have been shown to be robust to several geometric transformations; the second part for tampering localization, based on local quantized histograms of edge directions.



(a) Original image (b) JPEG compressed image

(c) Tampered image (d) Detected tamper

Fig. 1. Example of tampering localization

In the literature, image watermarking has been used to solve the problem of tampering localization [4][5]. A fragile watermark is inserted into the image when it is created, and extracted during the authentication phase. Tampering can be localized by identifying the damage to the watermark. Watermarking based schemes suffer from the following disadvantages: 1) watermarking authentication is not backward compatible with previously encoded contents (unmarked contents cannot be authenticated later by just retrieving the corresponding hash); 2) the image content is distorted by the watermark; 3) the bit-rate required to compress an image might increase due to the embedded watermark. Conversely, image hashing embeds a signature of the original content as part of the header information, or can provide a hash separately from the image content upon a user's request. In order to limit the rate overhead, the size of the hash needs to be as small as possible. At the same time, the goal of tampering localization calls for increasing the hash size, in order to capture as much as possible about the original image. In this paper we explicitly target these conflicting requirements by proposing a hashing technique based on compressive sensing principles. The key tenet is that, if the tampering is sparse enough (or it can be sparsified in

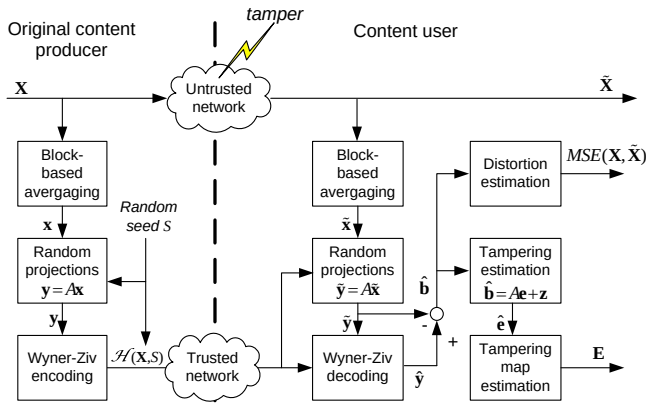


Fig. 2. Block diagram of the proposed tampering localization scheme

some orthonormal basis or redundant dictionary), it can be localized by means of a limited number of random projections of the original image. In addition, in order to keep the size of the hash as small as possible, the hash information is encoded by exploiting distributed source coding tools.

The rest of this paper is organized as follows. Section 2 gives a general description of the proposed tampering localization system; we give some guidelines about how to set the parameters of the system, by outlining a tampering model in Section 3 and using it for parameter tuning in Section 4. The results of tampering localization are presented in Section 5. Finally, Section 6 gives some concluding remarks.

2. SYSTEM OVERVIEW

The proposed tampering detection and localization scheme is depicted in Figure 2. 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. Users who want to authenticate the received image $\tilde{\mathbf{X}}$ use the hash to estimate the mean square error distortion between the original image $\mathbf{X}$ and the received one $\tilde{\mathbf{X}}$. In addition, if the tampering is sparse, the algorithm produces a tampering map $\mathbf{E} \in \mathbb{R}^N$ indicating the location of the malicious attack. In our system, we assume that the (noiseless) hash bits are sent upon request from a trusted authentication server and encrypted so that their integrity can be guaranteed [2].

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

1. *Block based averaging*: 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 is 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 $\mathcal{N}(0, 1/n)$, using some random seed S, which will be sent as part of the hash to the user.
3. *Wyner-Ziv encoding*: The random projections $\mathbf{y}$ are quantized with a uniform scalar quantizer with step size Δ. Bitplane extraction is performed on the quantization bin indexes. Each bitplane is encoded by sending syndrome bits generated by means of an LDPC code. The rate allocated to the hash depends on the expected distortion between the original and the tampered image, as explained in Section 4.

The content user receives the (possibly tampered) image $\tilde{\mathbf{X}}$ and requests the syndrome bits and the random seed of the hash $\mathcal{H}(\mathbf{X}, S)$ 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 localization works as follows:

1. *Block based averaging*: as before, but on the image $\tilde{X}$, producing the vector $\tilde{\mathbf{x}}$.
2. *Random projections*: $\tilde{\mathbf{y}} = \mathbf{A}\tilde{\mathbf{x}}$.
3. *Wyner-Ziv decoding*: A quantized version $\hat{\mathbf{y}}$ is obtained using the hash syndrome bits and $\tilde{\mathbf{y}}$ as side information. LDPC decoding is performed starting from the most significant bitplane. If the actual distortion between the original and the tampered image is higher than the maximum distortion expected by the original content producer (which determines the rate allocated to the hash) decoding might fail. In this case, the image is declared to be unauthentic and no tampering localization can be provided.
4. *Distortion estimation*: If Wyner-Ziv decoding succeeds, an estimate of the distortion in terms of the mean square error (MSE) between the original and the received image is computed from $\hat{\mathbf{b}} = \hat{\mathbf{y}} - \tilde{\mathbf{y}}$.
5. *Tampering estimation*: An estimate of the tampering $\mathbf{e} = \tilde{\mathbf{x}} - \mathbf{x}$ is obtained by solving the following undetermined system of linear equations:

$$\hat{\mathbf{b}} = \mathbf{A}\mathbf{e} + \mathbf{z}, \tag{1}$$

where $\mathbf{z}$ is the hash quantization noise. There exists an infinite number of solutions to (1); however, in the hypothesis that $\mathbf{e}$ is sparse, the optimal way for recovering $\mathbf{e}$ is to seek the *sparsest* solution of (1), i.e. the one that minimizes $\|\mathbf{e}\|_0$, where the ℓ_0 norm $\|\cdot\|_0$ simply counts the number of nonzeros entries of $\mathbf{e}$ [6]. Unfortunately, such a problem is NP hard and it is difficult to solve in practice. Nonetheless, recent literature about Compressive Sensing [6] has shown that, if $\mathbf{e}$ is sufficiently sparse, an approximation of it can be recovered by solving the following ℓ_1 minimization problem:

$$\hat{\mathbf{e}} = \min \|\mathbf{e}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{b}} - \mathbf{A}\mathbf{e}\|_2 \leq \epsilon \tag{2}$$

where ϵ is chosen such that $\|\mathbf{z}\|_2 \leq \epsilon$. Problem (2) is a special instance of a second order cone program (SOCP) [6] and can be solved in $O(n^3)$ time. Nevertheless, several fast algorithms have been proposed in the literature that attempt to find the sparsest $\mathbf{e}$ satisfying the constraint $\|\mathbf{b} - \mathbf{A}\mathbf{e}\|_2 \leq \epsilon$. In our experiments, we adopt the SPGL1 algorithm [7], which is specifically designed for large scale sparse reconstruction problems. If $\mathbf{e}$ is not sparse enough with respect to the number of projections m, as further discussed in Section 4, the solution found does not fulfil the constraint. In such cases, it is not possible to perform tampering localization.
6. *Tampering map estimation*. If a sparse solution to the problem (2) can be found, the vector $\mathbf{e}$ is interpolated to produce $\mathbf{E} \in \mathbb{R}^N$, which represents a map of the estimated tampering. If the application requires the knowledge only of the

tampering location, the map $\mathbf{E}$ can be thresholded to produce a binary decision (tampered vs. not tampered) with the granularity of a $B \times B$ block.

We notice that the proposed system declares an image as unauthentic only when Wyner-Ziv decoding fails. If Wyner-Ziv decoding succeeds, a richer information is provided, in terms of the distortion induced by the tamper and its sparsity. For example, in some applications an image could be deemed authentic if the distortion does not exceed a user defined threshold and the tamper is not sparse, i.e. is distributed across the image. This approach enables a flexible authentication scheme that can be tailored to the specific application requirements.

3. TAMPERING MODEL

Before detailing how to tune the parameters of the proposed algorithm, we introduce a simplified tampering model that enables to provide some guidelines for parameter setting. We model the the effect of the tampering as

$$\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{e}, \tag{3}$$

where $\mathbf{e} \in \mathbb{R}^n$ is a vector with k non zero components. By letting $k \ll n$ we are able to model spatially sparse attacks, whereas when $k \to n$ the tamper tends to be uniformly distributed across the image. Let the set I_k, $|I_k| = k$, denote the support of the nonzero components of $\mathbf{e}$. Also, assume that the k nonzero entries are i.i.d. $\sim \mathcal{N}(0, \sigma_s^2)$. Clearly, the mean square error between the original and the tampered image is

$$\sigma_e^2 = \frac{1}{n} E[\|\mathbf{e}\|_2^2] = k/n \cdot \sigma_s^2. \tag{4}$$

Although this model is rather simple, the Gaussian assumption is justified by the central limit theorem, since the block averaging module is summing together up to B^2 tampered pixels whose statistics are not explicitly modeled. In the following, we assume that the original content producer and the content user agree on what is the maximum level of the attack against which tampering localization can be successfully accomplished, i.e. the maximum number of nonzero components k and the maximum "intensity" of the attack σ_s^2 are known a-priori.

Let $\mathbf{b}$ denote the error produced in the random projections by the tampering attack:

$$\mathbf{b} = \tilde{\mathbf{y}} - \mathbf{y} = \mathbf{A}(\tilde{\mathbf{x}} - \mathbf{x}) = \mathbf{A}\mathbf{e}. \tag{5}$$

Each entry of $\mathbf{b}$ is given by a dot product between a row of $\mathbf{A}$ and the nonzero components of $\mathbf{e}$, i.e. it is a scalar product between normal i.i.d. vectors, $b_i = \sum_{j \in I_k} a_{ij} e_j$. Using a standard statistical argument [8], it can be shown that $\sigma_b^2 = \frac{1}{n} E[\|\mathbf{b}\|_2^2] = \sigma_e^2$. In addition, if k is sufficiently large, we can invoke the central limit theorem and state that $\mathbf{b}$ is i.i.d. Gaussian.

4. PARAMETERS SETTING

A crucial aspect for obtaining a correct tamper detection and localization using the procedure described above is to carefully choose the bit budget spent for encoding the random projections $\mathbf{y}$ with the Wyner-Ziv codec and the number m of these projections.

First, let us now turn our attention to the rate R needed to encode the hash signature in terms of bits/random projection. Exploiting distributed source coding principles, the original content producer encodes $\mathbf{y}$ without a deterministic knowledge of the side information $\tilde{\mathbf{y}}$, which is disclosed at the content user side only. We notice that the hypothesis of the Wyner-Ziv theorem are satisfied, since, as shown above, the correlation noise $\mathbf{b} = \tilde{\mathbf{y}} - \mathbf{y}$ is i.i.d. Gaussian. To perform Wyner-Ziv encoding, the original content producer needs an estimate of the correlation noise variance $\sigma_b^2 = k/n \cdot \sigma_s^2$. According to the Wyner-Ziv theorem, for a given target mean square error distortion D, the rate needed to lossy encode the random projections $\mathbf{y}$ is the same as if $\tilde{\mathbf{y}}$ were available at the encoder. For Gaussian i.i.d. sources the rate-distortion curve is given by $R^{WZ}(D) = 1/2 \log_2(\sigma_b^2/D)$ bits/sample. In the practical Wyner-Ziv codec implemented in our system, due to the use of LDPC codes at the bitplane level and a simple uniform scalar quantization rule, the operational rate-distortion curve is given by $R(D) = R^{WZ}(D) + \Delta R$, where $\Delta R = 1.12$ bit/sample at high rates. The value of ΔR is comparable with the one obtained by a uniform scalar quantizer applied directly to the Gaussian i.i.d. source $\mathbf{b}$ (which yields $\Delta R = 1/2 \log_2(\gamma^2/3) = 1.21$ bits/sample [9], with the overload factor γ equal to 4), despite $\mathbf{b}$ is unknown at the encoder.

Let us consider now distortion estimation and localization of tampering. Let $\hat{\mathbf{y}}$ denote the quantized version of $\mathbf{y}$ reconstructed by the content user after Wyner-Ziv decoding, and $\hat{\mathbf{b}} = \hat{\mathbf{y}} - \tilde{\mathbf{y}}$. We can write

$$\hat{\mathbf{b}} = A(\mathbf{x} + \mathbf{e}) - Q(A\mathbf{x}) = A\mathbf{e} + A\mathbf{x} - Q(A\mathbf{x}) = A\mathbf{e} + \mathbf{z} \tag{6}$$

where $Q(\cdot)$ denotes the operation of quantization and $\mathbf{z}$ the quantization noise introduced by the Wyner-Ziv coding process ($\sigma_z^2 = \frac{1}{n} E[\|\mathbf{z}\|_2^2] = D$). Assuming that the error introduced by tampering and quantization are uncorrelated, we can obtain an estimate of the distortion due to tampering as

$$\hat{\sigma}_e^2 = \sigma_b^2 = \frac{1}{n} \|\hat{\mathbf{b}}\|_2^2 - D \tag{7}$$

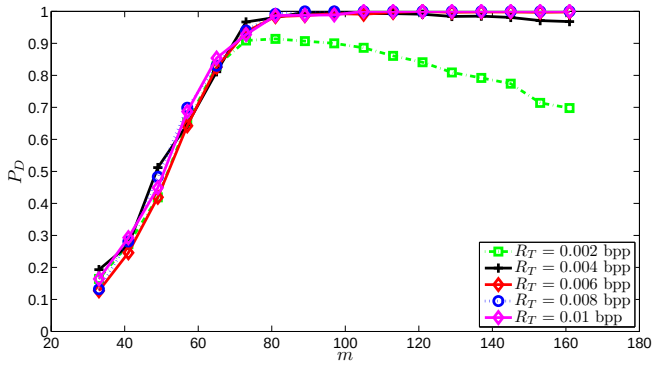
In order to localize the tamper, we need to solve problem (2). Let Δ denote the quantizer step size used to achieve the target distortion D. At high rates and for smooth pdf's $D \simeq \Delta^2/12$. For the case of uniform scalar quantization, the norm of the quantization error is upper bounded, i.e. $\|\mathbf{z}\|_2 \leq \sqrt{n}\frac{\Delta}{2}$. Since from equation (6) we know that $\|\hat{\mathbf{b}} - A\mathbf{e}\|_2 = \|\mathbf{z}\|_2$, we set $\epsilon = \sqrt{n}\frac{\Delta}{2}$ in (2). Finally, in order to find a k-sparse estimate of the tampering $\mathbf{e}$ solving (2), compressive sensing theory dictates that the number of random projections m must satisfy:

$$m \geq Ck \log(n/k) \tag{8}$$

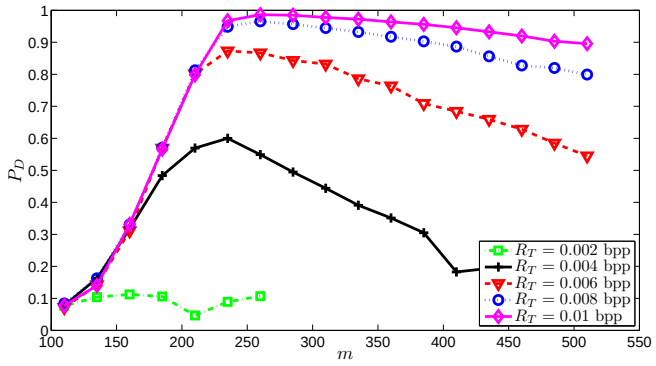
where the constant C generally depends on the actual algorithm used to solve problem (2). In Section 5 we experimentally evaluate C when the goal is to localize the support of the vector $\mathbf{e}$ rather than reconstructing $\mathbf{e}$ exactly.

5. EXPERIMENTAL RESULTS

We have simulated the tampering localization system for different sparsity conditions of the attack on some 512×512 pixel images. Following the procedure outlined in Section 2, we divide the image into 16×16 non-overlapping blocks and assemble the vector $\mathbf{x}$ with $n = 1024$ elements by computing the average luminance of each block. To produce the tamper, we add some Gaussian noise with variance $\sigma_s^2 = 1000$ at k random positions of $\mathbf{x}$. Assuming that the bit-rate for the Wyner-Ziv encoding of the random measures $\mathbf{y}$ is correctly allocated as devised in Section 4, at the content user's side $\mathbf{y}$ can be decoded using the side information of the hash with



(a) $k/n = 0.01$



(b) $k/n = 0.05$

Fig. 3. Probability of detection at zero false detection rate (P_D) for different total bit budget R_T.

a mean square error that is equal to the distortion D introduced by the Wyner-Ziv codec. To evaluate the localization performance, we threshold the reconstructed error $\hat{\mathbf{e}}$ and compare the estimated tampering locations with the actual ones. By varying the threshold τ it is possible to build a ROC curve for a given number of measurements m and a total rate for the syndrome bits of $R_T = mR(D)/N$ bits per pixel. We define P_D, the probability of detection at zero false detection rate, as the highest value of true positive rate in the ROC curve for which the false positive rate is zero. It gives information about the tamper detection power of the system when no false tampering warning is allowed.

Figure 3(a) shows the probability of detection P_D for different rates R allocated to the hash, for an increasing number of measurements m, when $k/n = 0.01$. The different curves correspond to different total rate constraints R_T for the hash signature: the more the number of measurements for a given rate, the smaller the number of "bits/measurement" R allocated. Increasing the number of bits per measurement, the probability of detection P_D increases, finally saturating at the maximum value of 1. When the size of the measurement vector, m, is less than the bound in (8), the ℓ_1 reconstruction may give rise to a large number of false detections, i.e. only a small fraction of the true tampering locations is actually recognized. This is because the reconstructed vector $\mathbf{e}$ has a number of spurious nonzero entries that were not present in the original tampering. The maximum value of P_D is reached as condition (8) begins to hold. Since the maximum value of P_D in Figure 3(a) is reached around $m = 80$, in can be easily verified that the constant C is equal to about 1.72 for a 1%-sparse tampering. At low bit-rates, the probability of detection decreases for a large number of measurements (see $R_T = 0.002$ bpp in Figure 3(a)): the reason why this phenomenon occurs is that the quantization noise affecting the reconstructed random measurements is no longer negligible as the number of bits allocated to each measurements becomes too small, so that the reconstruction of $\mathbf{e}$ is seriously compromised. This is more evident in Figure 3(b), which shows a similar experiment for the case of 5%-sparse tampering. In this case, the maximum P_D is reached approximatively for $m \geq 250$ ($C = 1.63$). Here we can see that, by increasing the sparsity of the signal, in virtue of equation (8), a larger number of measurements $\mathbf{y}$ is needed to solve (2). Therefore, using the same rate budgets of Figure 3(a), the number of bits allocated to each measurement is much smaller than the case with $k/n = 0.01$, and obtain a probability of detection $P_D = 1$ it is necessary to spend more bits for the hash. Nevertheless, we remark that for a typical attack, the sparsity of the tampering is less than 5%; for the picture shown in Figure 1, we had $k/n = 0.03$. It should be noted that this sparsity was computed in the spatial domain, but a preliminary transformation to some other basis (e.g. wavelet or Fourier domain) may further sparsify the tampering.

6. CONCLUSIONS

In this paper we have presented a hash-based system for authenticating an image and localizing possible modifications, in the case that the tampering is sufficiently sparse in the spatial domain. By leveraging compressive sensing and distributed source coding principles, we have built a small hash signature of the image; furthermore, if some assumptions about the tampering model hold, it is possible to estimate the number of bits to allocate to the hash. In future research, we will investigate the possibility of addressing tampering models that can be sparsified in a domain different from the spatial one (e.g. using wavelet or Fourier transform).

7. REFERENCES

[1] Y.C. Lin, D. Varodayan, and B. Girod, "Image authentication based on distributed source coding," in *IEEE International Conference on Image Processing*, S.Antonio, TX, September 2007, vol. 3.

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

[3] S. Roy and Q. Sun, "Robust Hash for Detecting and Localizing Image Tampering," in *IEEE International Conference on Image Processing*, S.Antonio, TX, 2007, vol. 6.

[4] J. Fridrich, "Image watermarking for tamper detection," in *IEEE International Conference on Image Processing*, Chicago, October 1998, vol. 2.

[5] J.J. Eggers and B. Girod, "Blind watermarking applied to image authentication," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, 2001, vol. 3.

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

[7] 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`.

[8] V.V. Khlobystov and V.K. Zadiraka, "Distribution density of scalar product of Gaussian vectors," *Cybernetics and Systems Analysis*, vol. 8, no. 3, pp. 477–481, 1972.

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