A COMPRESSIVE-SENSING BASED WATERMARKING SCHEME FOR SPARSE IMAGE TAMPERING IDENTIFICATION

G. Valenzise, M. Tagliasacchi, S. Tubaro

Politecnico di Milano
Dip. Elettronica e Informazione - Italy

G. Cancelli, M. Barni

Università di Siena
Dip. Ingegneria dell'Informazione - Italy

ABSTRACT

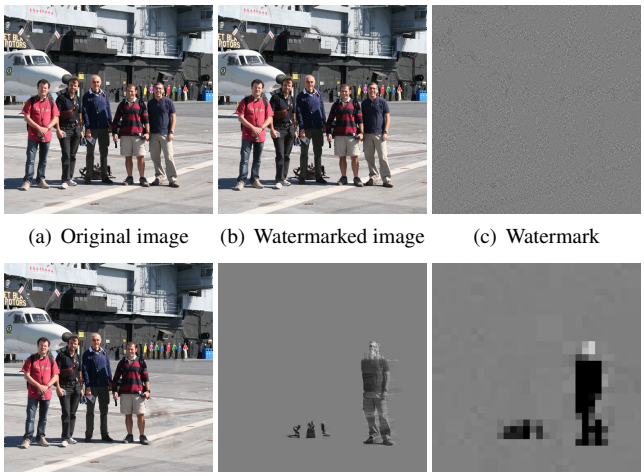
In this paper we describe a robust watermarking scheme for image tampering identification and localization. A compact representation of the image is first produced by assembling a feature vector consisting of pseudo-random projections of the decimated image. Then, the quantized projections are encoded to form a hash, which is robustly embedded as a watermark in the image. By recovering the watermark the random projections are obtained, and then used to estimate the distortion of the received image. If tampering is sufficiently sparse or compressible in some basis description, a map of the introduced modification is recovered. The system relies on Compressive Sensing and Distributed Source Coding principles to reduce the size of the hash of a 1024×1024 image, to about 4,000 bits. With this hash length, tampering sparse up to 20% and with a tampering energy around a PSNR of 15 dB can be successfully localized.

***Index Terms*—** Watermarking, Compressive Sensing, Image Tampering Localization

1. INTRODUCTION

In the past few years, tampering with images and disseminating user-created contents through peer-to-peer or social networks have become an easy and cheap task. These developments brought along serious risks in the credibility of the circulated contents, since illegitimate and perhaps viciously modified images might be diffused in order to completely twist the original meaning of the picture. Watermarking schemes have been traditionally used to defend and verify the authenticity and integrity of image contents. In particular, *Content-fragile* (or semi-fragile) watermarks [1, 2] are generally used for this purpose: a mark designed to be robust with respect to legitimate, perceptually-irrelevant modifications, and at the same time to be fragile with respect to perceptually and semantic significant alterations is embedded directly in the image data. When a new copy of the media content is examined, a possible tampering can be detected by identifying the damage to the extracted watermark.

In practice, even if an image has been maliciously tampered with, this does not necessarily imply that its semantic content is definitively lost. Consider the picture in Figure 1(a), which depicts five people visiting the Midway aircraft carrier. The image has been marked with an imperceptible watermark to produce the picture in Figure 1(b); the visual imperceptibility of the mark is supported by Figure 1(c), which shows the difference between the original and the watermarked image, with depth range enhanced for presentation convenience. The original picture has been altered in Figure 1(d) by removing the person on the right of the photo, however the picture can still be used as an evidence that the person in the center of the group was actually on the flight deck of the ship. For this to be possible, it is necessary that the tampering is not only revealed but the areas interested by it localized (like in Figure 1(e) and 1(f)).



(a) Original image (b) Watermarked image (c) Watermark

(d) Tampered image (e) Tampering (f) Tampering map

Fig. 1. Example of the identification capabilities of the proposed watermarking scheme.

In addition to watermarking, the problem of tampering identification and localization has been tackled by means of multimedia hashing [3, 4, 5]. Hashing techniques extract a compact representation of the image data from the original picture to form a hash, which is supposed to be stored and transmitted to the user *separately* from the image content through some secure channel. While this approach has some advantages with respect to watermarking (e.g. it does not modify the content of the image and enables the authentication of unmarked contents), it suffers from the considerable drawback of requiring ad-hoc architectures to be implemented (e.g. an authentication server which sends the hash to the user upon request).

In a previous work [5] some of the authors of this paper have described an image hashing system for detecting and localizing sparse image tampering based on compressive sensing and distributed source coding principles. First, a small set of quantized pseudo-random projections are extracted from a decimated version of the original image; then, the hash is generated by storing the syndrome bits derived from LDPC (Low Density Parity-check Codes) encoding of the quantized projections. When the content user receives the (possibly) tampered image, it tries to decode the

Contact authors: Giuseppe Valenzise (valenzise@elet.polimi.it) and Giacomo Cancelli (giacomo.cancelli@unisi.it)

hash using the received image as side information. If the transmitted image is not excessively distorted, the hash can be successfully recovered (up to quantization errors) and can be used to estimate a tampering map, provided that the tampering is sparse, by solving a convex optimization problem. In [5], the hash transmission required the availability of a trusted authentication server, which is not always feasible in practical situations. In this paper we relax that assumption by embedding the hash signature directly in the image content, using a robust watermarking technique. This turns out to be a non-trivial task due to the necessity of coping with the conflicting requirements of watermark robustness (so that the hash is correctly recovered even in the presence of tampering), payload (the size of the hash can not be reduced below a certain threshold) and localization accuracy (calling for a larger hash). We also generalize the tampering localization procedure to the case where the attack is sparse (or compressible) in any orthonormal basis or redundant dictionary available at the content user side.

The rest of this paper is organized as follows. Section 2 provides a brief review of the main concepts of compressive sensing and distributed source coding; Section 3 details the building blocks of the system; finally, Section 4 discusses the experimental results and Section 5 gives some concluding remarks.

2. BACKGROUND

2.1. Compressive Sensing

Compressive sensing theory asserts that it is possible to perfectly recover a signal from a limited number of incoherent, non-adaptive linear measurements, provided that the signal can be represented by a small number of non-zero coefficients in some basis expansion. Let $\mathbf{s} \in \mathbb{R}^N$ be a k-sparse vector, i.e. only k out of the N elements of $\mathbf{s}$ are nonzero. Suppose we can write the signal to be acquired $\mathbf{x} \in \mathbb{R}^N$ as $\mathbf{x} = \mathbf{\Phi s}$, i.e. it can be represented by a few basis vector in the orthonormal basis $\mathbf{\Phi}$ using the coefficients $\mathbf{s}$. Let $\mathbf{y} \in \mathbb{R}^n$, $n < N$, be a number of linear random projections (measurements) obtained as $\mathbf{y} = \mathbf{Ax}$. If the measurement matrix $\mathbf{A}$ satisfies a *Restricted Isometry Property* (RIP) [6], it can be shown [7] that solving the following optimization problem:

$$\min \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A\Phi s}. \tag{1}$$

is equivalent to finding the sparsest solution $\mathbf{s}$ to $\mathbf{y} = \mathbf{A\Phi s}$, provided that the number of measurements satisfies $n \geq Ck \log(N/k)$. In practice, the RIP is satisfied whenever the columns of matrix $\mathbf{A}$ are incoherent with the basis $\mathbf{\Phi}$ in which the signal is sparse; it turns out that sampling the entries of matrix $\mathbf{A}$ from a Gaussian distribution with zero mean and variance $1/N$ provides a measurement basis which is incoherent with overwhelming probability with any other given basis. In most practical applications, measurements are affected by noise (e.g. quantization noise). Let us consider noisy measurements $\mathbf{y} = \mathbf{Ax} + \mathbf{z}$, where $\mathbf{z}$ is a norm-bounded noise, i.e. $\|\mathbf{z}\|_2 \leq \sigma$. An approximation of the original signal $\mathbf{x}$ can be obtained by solving the modified problem:

$$\min \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A\Phi s}\|_2 \leq \sigma. \tag{2}$$

Problem (2) is an instance of a second order cone program and can be solved in $O(n^3)$ time. 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 [8], which is specifically designed for large scale sparse reconstruction problems.

2.2. Distributed source coding (DSC)

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 [9] states that, in general, $R^{WZ}_{X|Y}(D) \geq R_{X|Y}(D)$ but $R^{WZ}_{X|Y}(D) = R_{X|Y}(D)$ for jointly Gaussian memoryless sources and mean square error (MSE) as distortion measure.

Practical DSC implementations start from the side information Y to reconstruct the original signal X through error correcting codes such as LDPC codes. To this end, the source X is quantized at the encoder with 2^J levels, and the J bitplanes are independently encoded, computing syndrome bits by means of a LDPC encoder. At the decoder, syndrome bits are used together with the side information Y to "correct" Y into a quantized version of X, $\hat{X}$, performing LDPC decoding, typically starting from the most significant bitplanes. If the p.d.f. of X and $Y - X$ are known, one could attempt to allocate the syndrome bits directly at the encoder, removing the need for a feedback channel iterative bit request. In this case, decoding will work only when the distortion of the side information with respect to the original signal is no larger than the maximum allowed by the number of transmitted syndrome bits.

3. DESCRIPTION OF THE SYSTEM

The proposed tampering identification and localization scheme is depicted in Figure 2. The producer of the original image content embeds the watermark in the image before distributing the content to users. When a content user receives the marked image, he can obtain two kinds of information from the tampering identification system: an estimate of the distortion between the original and the tampered image, and a map of the introduced tampering. In the following, we consider in detail the modules depicted in Figure 2 for both content producer and content user.

The original content producer generates the watermarked image $\mathbf{X}_W$ as follows:

1) Block based averaging: The original image $\mathbf{X} \in \mathbb{R}^N$ 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 $\mathbf{y} = \mathbf{Ax}$. 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, to be be embedded in the watermark.

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 to form the hash $\mathcal{H}(\mathbf{X}, S)$. The rate allocated to the hash depends on the expected distortion between the original and the tampered image [5].

4) Watermark embedding: The signature $\mathcal{H}(\mathbf{X}, S)$ is embedded within the image through a high-payload and robust watermark scheme described in [10] by producing $\mathbf{X}_W$, i.e. the authenticated version of the original image $\mathbf{X}$.

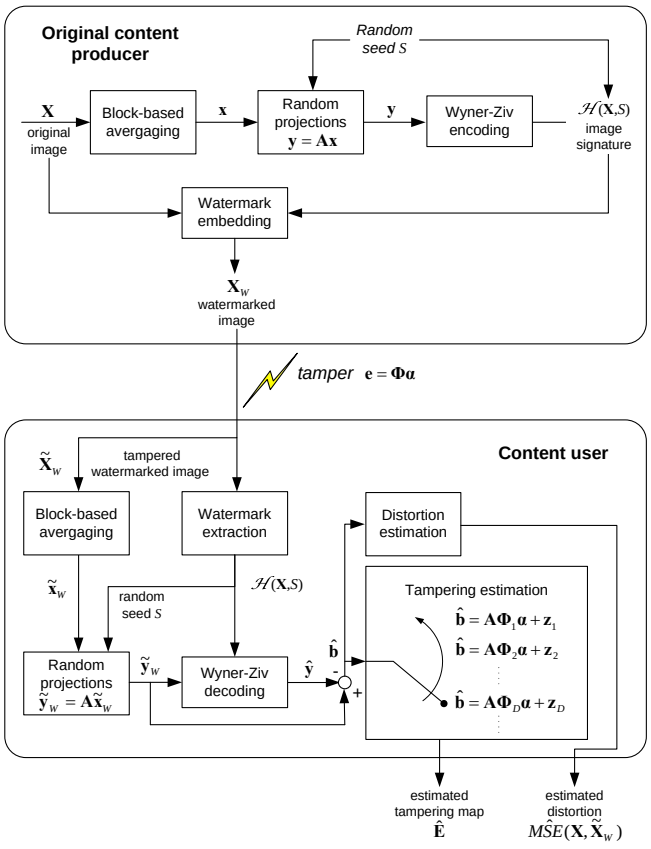


Fig. 2. Block diagram of the proposed system.

The content user receives image $\tilde{\mathbf{X}}_W$ and performs the following actions to check its authenticity:

1) Watermark extraction: First of all, the watermark algorithm check the presence of the watermark. If the watermark is not present then either the image was not previously authenticated or the applied tamper was too strong. In both cases a potential tamper is detected even though the proposed scheme is not able to define a tampered region. Otherwise, the watermark $\mathcal{H}(\mathbf{X}, S)$ is extracted.

2) Block based averaging: as before, but on the image $\tilde{\mathbf{X}}_W$, producing the vector $\tilde{\mathbf{x}}_W$.

3) Random projections: $\tilde{\mathbf{y}}_W = \mathbf{A}\tilde{\mathbf{x}}_W$, where $\mathbf{A}$ is generated using the seed S extracted from the watermark.

4) Wyner-Ziv decoding: A quantized version $\hat{\mathbf{y}}$ is obtained using the hash syndrome bits and $\tilde{\mathbf{y}}_W$ 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 number of bits allocated to the hash) decoding might fail. In this case, the image is declared to be unauthentic and no tampering localization can be provided.

5) 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 as:

$$\hat{MSE}(\mathbf{X}, \tilde{\mathbf{X}}_W) \approx \frac{1}{m}\|\hat{\mathbf{b}}\|_2^2 = \frac{1}{m}\|\hat{\mathbf{y}} - \tilde{\mathbf{y}}\|_2^2. \quad (3)$$

6) Tampering estimation: Provided that the tampering signal $\mathbf{e}$ is sparse or compressible in some orthonormal basis $\mathbf{\Phi}$, i.e. it can be written as $\mathbf{e} = \mathbf{\Phi}\boldsymbol{\alpha}$ where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is a sparse vector, an estimate of the tampering $\mathbf{e} = \tilde{\mathbf{x}} - \mathbf{x}$ is obtained by observing that $\mathbf{b} = \tilde{\mathbf{y}}_W - \hat{\mathbf{y}} \approx \mathbf{A}(\tilde{\mathbf{x}} - \mathbf{x}) = \mathbf{A}\mathbf{e} = \mathbf{A}\mathbf{\Phi}\boldsymbol{\alpha}$. Thus, we can replace (2) with

$$\min \|\mathbf{e}\|_1 \quad \text{s.t.} \quad \|\mathbf{b} - \mathbf{A}\mathbf{\Phi}\boldsymbol{\alpha}\|_2 \leq \epsilon, \quad (4)$$

where $\epsilon^2 = m\frac{\Delta^2}{12} + \lambda\sqrt{m}\frac{\Delta^2}{6\sqrt{5}}$ [11]. A tampering map is visualized by interpolating the tampering signal $\hat{\mathbf{e}}$ to the original resolution.

4. SYSTEM TUNING AND RESULTS

We tested the proposed system in order to evaluate both the robustness of the watermarking against attacks and the tampering reconstruction capabilities. The results reported here refer to the 1024×1024 *Kyoto* color image, reported in Figure 3(a). The hash length has been fixed to 4004 bits using a rate-allocation algorithm for the syndrome bits [5] and fixing the number of measurements in such a way that the maximum sparsity of the tampering is about 20%.

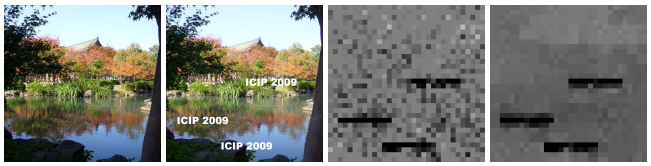
From a watermarking point of view, the main problem of the proposed system is the necessity of trading off robustness for payload size. Indeed, even if the hash to be hidden within the watermark is rather small, embedding about 4000 bits in a robust way (while keeping the image quality virtually intact) is a rather difficult task. In our implementation of the scheme depicted in the previous section, we adopt a modified version of the watermarking algorithm described in [10]. First, we embed the watermark in the full frame DFT coefficients, instead of the block-DCT ones. The watermark occupies quite a large portion of the spectrum, namely from diagonal 30 to 364 for a 1024×1024 image. Finally we embed the watermark in all the color bands. In this way, we were able to embed the necessary 4000 bits in a rather robust way. With this payload size, we are able to guarantee a PSNR of the watermarked image above 34 dB, resulting is a barely visible watermark (see figure 1 for an example).

To assess the quality of tampering identification, we compute the normalized Mean Square Error (MSE_N) of the reconstructed tampering signal with respect to the true one, i.e.:

$$MSE_N = \frac{\|\mathbf{e} - \hat{\mathbf{e}}\|_2^2}{\|\mathbf{e}\|_2^2}. \quad (5)$$

In order to make results reproducible, we have first simulated a tampering attack where logos are added to the image; this approximates quite well sparse attacks in the pixel domain. In a second phase, we have introduced a non pixel-sparse modification, namely a gamma correction (with fixed parameter $\gamma = 0.8$) which has the effect of boosting tones towards brighter colors. In this second testbed the tampering is represented by brightness adjustment plus the logos (see Figure 3(b) for an example).

Figure 4(a) shows the normalized reconstruction MSE for the first experiment (pixel-sparse logos), as a function of the number of logos. Here the sparsity of the attack ranges from about 1% (1 logo) to 12% (15 logos). The red solid line refers to the proposed system, while the blue line reports the reconstruction performance of the system in [5]. In [5] the hash is built in the same way as described in Section 3, with the important difference that it is not embedded in the original image as a mark, but it is sent separately to the decoder. In order to evaluate the cost that has to be paid for the architectural simplification introduced by the watermarking-based system, we have two consider to aspects. First, the mark inevitably introduces additional noise to the image, which might affect the performance of the tampering reconstruction. However, the noise introduced by the



(a) (b) (c) (d)

Fig. 3. Identification of the tampering (brightness adjustment plus 3 logos) in two different bases. (a) Watermarked image; (b) Tampered image; (c) Tampering reconstructed in the pixel domain; (d) Tampering reconstructed in the Haar domain

watermark is mostly concentrated in the high-frequency details of the image; since the hash is computed on a decimated version of the image, the net effect of the mark noise is negligible, and the performance of the hashing and watermarking systems are practically the same, as shown in the figure. The second important limitation of the watermarking-based system is that the hash can be extracted without losses only if the strength of the attack is not too heavy. In our experiments, we were able to extract correctly the mark with images corrupted with up to 9 logos, which correspond to a PSNR of about 15 dB: that's why the watermarking curve in Figure 4(a) ends before the hash's one.

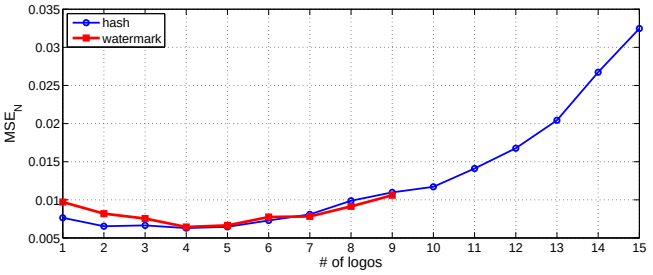
For the the second kind of attacks (brightness plus logos), the performance of the tampering reconstruction in the pixel domain decreases, as illustrated by the two solid curves in Figure 4(b). The justification for the higher error with a lower number of logos in this case comes from compressive sensing theory, which asserts that, even if the signal is not k-sparse the reconstruction process will return the largest k coefficients of the signal. When more logos are present, the energy of the tampering is mostly concentrated in the logos rather than in the brightness adjusted background (which has lower energy). Thus, with a larger number of logos, in the pixel domain we are able to lower the reconstruction MSE, which is in any case still unacceptable from a visual point of view (see Figure 3(c)). The limitations of the reconstruction in the pixel domain can be overcome by attempting the ℓ_1 recovery in (4) in a wavelet base, e.g. in the Haar domain, which is known to sparsify piece-wise smooth signals. In this case, even if the Haar wavelet coefficients are still not sparse, their energy is packed in a few coefficients, a fact that enables a higher quality reconstruction, as illustrated by the two dashed lines of Figure 4(b) and, visually, in Figure 3(d).

5. CONCLUSIONS

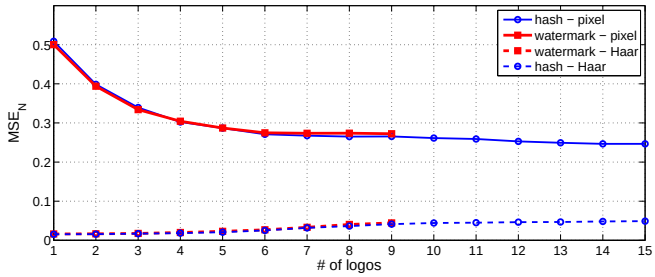
In this paper we have proposed a watermarking system which embeds a small hash in the host image in order to guarantee protection and localization of possible forgeries. Even if embedding the hash as a watermark gives no guarantees on its lossless extraction at the decoder, this solution removes the assumption of an auxiliary, secure channel to transmit the hash made in previous works, and thus potentially enables the application of the system as a practical tampering protection solution.

6. REFERENCES

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



(a) Logos



(b) Brightness+logos

Fig. 4. Normalized MSE.

[2] 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.

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

[4] 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.

[5] M. Tagliasacchi, G. Valenzise, and S. Tubaro, "Localization of sparse image tampering via random projections," in *Proc. IEEE Int. Conf. Image Processing*, San Diego, USA, 2008.

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

[7] D.L. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[8] 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`.

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

[10] A. Abrardo and M. Barni, "Fixed-Distortion Orthogonal Dirty Paper Coding for Perceptual Still Image Watermarking," in *Information Hiding: 6th International Workshop, IH 2004, Toronto, Canada, May 23-25 2004, Revised Selected Papers*. Springer, 2004.

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