Information Sciences 325 (2015) 33–47

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Compressive image sensing for fast recovery from limited samples: A variation on compressive sensing

Chun-Shien Lu*, Hung-Wei Chen

Institute of Information Science, Academia Sinica, Taipei, Taiwan, ROC

ARTICLE INFO

Article history:
Received 27 June 2014
Revised 4 June 2015
Accepted 4 July 2015
Available online 11 July 2015

Keywords:
Compressed sensing
Reconstruction
Sampling
Sparsity
Transform

ABSTRACT

In order to attain better reconstruction quality from compressive sensing (CS) of images, exploitation of the dependency or correlation patterns among the transform coefficients commonly has been employed. In this paper, we study a new image sensing technique, called compressive image sensing (CIS), with computational complexity $O(m^2)$, where m denotes the length of a measurement vector $y = \phi x$, which is sampled from the signal x of length n via the sampling matrix ϕ with dimensionality $m \times n$. CIS is basically a variation on compressive sampling.

The contributions of CIS include: (i) reconstruction speed is extremely fast due to a closed-form solution being derived; (ii) certain reconstruction accuracy is preserved because significant components of x can be reconstructed with higher priority via an elaborately designed ϕ; and (iii) in addition to conventional 1D sensing, we also study 2D separate sensing to enable simultaneous acquisition and compression of large-sized images.

1. Introduction

In this section, we describe the background of compressed sensing in Section 1.1, discuss related work in Section 1.2, and describe the contributions and overview of proposed method in Section 1.3, before providing the outline of this paper in Section 1.4.

1.1. Background on compressed sensing

Compressed/Compressive sensing (CS) has received considerable attention recently due to its revolutionary development in simultaneously sensing and compressing signals with certain sparsity. Moreover, the architecture of the so-called single-pixel camera [19,30] has promoted the practicality of compressed sensing for images. CS is mainly composed of two steps. Let x denote a k-sparse signal of length n to be sensed, let ϕ of dimensionality $m \times n$ represent a sampling matrix, and let y be the measurement of length m. At the encoder, a signal x simultaneously is sensed and compressed via random projection, and the obtained samples are called measurements y in the context of compressed sensing. They are related via random projection as:

$$y = \phi x. \tag{1}$$

The measurement rate is defined as $0 < \frac{m}{n} < 1$ or $0 < \frac{m}{n} << 1$, which indicates the compression ratio (without quantization) without storing the original signal of length n. At the decoder, the original signal x to be sensed can be perfectly recovered by

* Corresponding author. Tel.: +886 2 27883799x1513; fax: +886 2 27824814.
E-mail address: lcs@iis.sinica.edu.tw (C.-S. Lu).

http://dx.doi.org/10.1016/j.ins.2015.07.017
0020-0255/

means of convex optimization or greedy algorithms if the relationship between m and k, i.e.,

$$m = O\left(k \cdot \log\left(\frac{n}{k}\right)\right), \tag{2}$$

is satisfied [7].

For convex optimization-based CS algorithms, sparse signal recovery will be time-consuming and intractable if ℓ_0-minimization is adopted. ℓ_0-minimization seeks to find k non-zero entries of a signal if the signal is k-sparse in either the time/space or transform (e.g., DCT or wavelet) domain. The solution can become more tractable if the constraint of ℓ_0-minimization is relaxed and ℓ_1-minimization is used instead. Several algorithms relying on ℓ_1-minimization have been presented in the literature.

In addition to convex optimization, non-convex programming (or greedy) algorithms, like Orthogonal Matching Pursuit (OMP) [41], are an alternative for sparse signal recovery. Basically, OMP has been recognized as a "fast" algorithm with time complexity $O(kmn)$ with reasonable reconstruction quality in some cases.

On the other hand, in the context of compressed sensing (CS) [15], the constraint of sparsity enables the possibility of sparse signal recovery to use the measurements with the number (far) fewer than the original signal length. Moreover, the measurements generated from random projection of the original signal via a sampling matrix are weighted equally; i.e., no measurement is more significant than the others. Thus, CS inherently is weakened in handling less-sparse signals, such as highly textured images. The problem here is if we can yield weighted measurements so that less sparse signals can be quickly reconstructed while maintaining good reconstruction quality. Namely, we seek to find approximate reconstruction instead of an exact reconstruct for multimedia data that permit certain content loss.

1.2. Related work: CS methods exploiting known sparsity patterns or partially known support

In the compressed sensing literature, many studies have explored the structure or correlation inherent in the transformed coefficients in order to better reconstruct the signal from its corresponding measurement vector. Inspired by the concept of JPEG2000 compression, the tree structure of wavelet transform has been exploited popularly.

In [16,17], instead of capturing non-adaptive or universal measurements, the authors propose attaining adaptive transform coefficients by exploiting the tree structure of the Haar wavelet. In terms of image quality and recovery speed, the so-called adaptive compressed sensing framework demonstrates its superiority over its non-adaptive counterparts.

In [24], a tree-structured Bayesian compressed sensing framework is proposed, wherein the hierarchical statistical models of wavelet and DCT were adopted and Markov chain Monte Carlo (MCMC) inference was employed. The computationally inefficient MCMC mechanism later is replaced with variational analysis in [25] to speed up recovery. Results show that their methods can achieve both accurate and fast CS recovery. The paradigm in [24,25] belongs to probabilistic structured sparsity [2].

Moreover, the concept of clustered sparsity has received considerable attention in compressed sensing. As summarized in [2] and Table I of [46], many existing CS algorithms [3,11,12,20,21,26,40] exploiting clustered sparsity need to know some pre-defined information, such as numbers, sizes, and positions of clusters, along with the degree of sparsity. In [46], the proposed Bayesian compressed sensing method, a kind of nonparametric recovery algorithm, could make use of clustered sparsity without relying on prior information. Basically, [46] is inspired by [25] in that variational analysis was used in place of MCMC for Bayesian inference in order to guarantee convergence within finite iterations. The major difference between [25] and [46] is that the former employs a directional graphical model for the tree structure of wavelet coefficients, while the latter uses an undirectional graphical model. Furthermore, in order to target the problem of reconstructing structured-sparse signals, belief propagation is employed in [39], which resembles turbo equalization from digital communications. The clustered sparsity-based compressed sensing methods mentioned above belong to deterministic structured sparsity [2].

It should be noted that, in [3], both tree structure and structured sparsity are considered and incorporated into two state-of-the-art CS algorithms, which are CoSaMP [34] and iterative hard thresholding (IHT) [4].

Recently, a so-called N-BOMP (N-way block OMP) method [5] has been developed based on exploiting Kronecker product and block sparsity. The authors prove the equivalence between the Tucker model and Kronecker representation for multiway arrays, thus, Kronecker structure can be used to solve the Tucker model-based underdetermined linear systems within compressive sensing. N-BOMP outperforms the existing tensor-based CS algorithms in that block sparsity of tensor is exploited such that the Kronecker dictionary can be used to speed recovery and improve reconstruction quality. Nevertheless, these advantages come from (also indicated in Subsection 7.2.1 of [5]) the assumption that, for a 2D image, it is pre-processed in advance to possess a perfect block sparsity pattern in that the important/significant coefficients in some transform domains fall within the specified block sparsity pattern while other insignificant coefficients are removed entirely. Therefore, N-BOMP is able to obtain reconstruction quality far better than the existing tensor CS algorithms under the prerequisite/restriction. Later, without making any assumptions about the sparsity pattern, Caiafa and Cichocki [6] present a fast non-iterative tensor compressive sensing method. It, however, assumes that the signal to be sensed and recovered has low multilinear-rank, leading to redundant sensing. This means that, under the same measurement rate, the reconstructed quality is (remarkably) lower than other CS algorithms.

In [29], we propose the use of tree structure sparsity pattern (TSSP) in tensor compressive sensing. TSSP can help to quickly find significant wavelet coefficients and save the execution time to calculate the maximum correlations in greedy algorithms. Its weakness is that there is no fast recovery algorithm that can exploit TSSP.

In addition to the aforementioned sparsity patterns, including the tree structure and clustered/block sparsity, other models of transform coefficients, including Laplacian scale mixtures [8], piecewise autoregressive model [44], Laplace prior [1], and

Gaussian Mixture Models [49] also have been employed within the compressed sensing framework. The nice property of structured sparsity mentioned above has been applied to a number of image processing applications beyond reconstruction. In [47], the inverse problems of inpainting and deblurring are solved via the proposed structured sparse model selection algorithm. The key is that the sparsity of local windows partitioned from an image can be better controlled. Basically, stable inversion can be achieved because the degree of freedom in selecting models is equal to the number of bases, and is considerably lower than overcomplete dictionary methods. Further, a work, called piecewise linear estimator (PLE), extended from [47] is presented in [48].

On the other hand, the paradigm of partially known support-based CS methods can be found in [9,10,42]. The idea is that the reconstruction quality can be improved if the prior knowledge of the support of the signal to be sensed is involved in the so-called modified CS framework. Typically, the known support of a signal in the wavelet transform domain can be the low-frequency subbands, which approximate/capture the original signal well. Specifically, Vaswani and Lu's work [42] is basis pursuit (BS) with partially known support while Carrillo et al.'s works [9,10] belong to greedy algorithms with partially known support.

1.3. Contributions and overview of our method

In this paper, we address the problems of accomplishing fast and accurate recovery of images in the framework of compressed sensing. The goal is to achieve so-called compressed image sensing (CIS) with approximate reconstruction. Our method basically is analogous to the aforementioned CS methods exploiting either known sparsity patterns or partially known support of a signal. Nevertheless, the methodology is fundamentally different in that we focus on the design of a new sampling matrix while the others, in essence, use some prior knowledge in either convex optimization-based or greedy-based CS algorithms.

Our first contribution is to investigate an elaborate design of the sampling matrix ϕ that can directly capture "important" measurements. With these important measurements, the quality original signal can be reconstructed sparsely based on the important (corresponding to low-frequency) components in a transform domain. In our CIS method, the qualities of reconstructed images mimic those of JPEG compressed images. The designed sampling matrix can be embedded readily into mobile devices with camera functionality to simultaneously capture and compress signals, and the sampled measurements can be transmitted efficiently to the decoder or remote server for turbo fast recovery of the captured signals. In particular, our CIS algorithm can be applied to a scenario where mobile device to mobile device (M2M) is considered. In addition, the bottleneck of distributed compressive video sensing (DCVS) [18,27,35] at the decoder now can be solved if the proposed method is used.

The second contribution of this paper is to study different sensing strategies for image reconstruction. More specifically, in addition to the commonly adopted 1D sensing, we also propose a 2D sensing strategy and demonstrate its impact on compressed image sensing and reconstruction. We find that 2D sensing indeed can benefit sensing and reconstruction of images. In addition, compressed sensing of large images is a challenge for compressed sensing of signals in a 1D form. We show that better reconstruction quality and speed can be achieved by means of 2D separate sensing.

1.4. Outline of this paper

The rest of this paper is organized as follows. The use of compressed sensing for capturing natural images is called compressed image sensing (CIS) and is discussed in Section 2. In Section 3, the idea behind our method and the proposed fast CIS recovery algorithm are described. Some characteristics of our method are discussed in Section 4. In Section 5, we provide extensive simulations to verify the proposed method in terms of reconstruction quality and computation speed. Finally, conclusions are given in Section 6.

2. Compressed sensing of digital images: problem statement

Inspired by the development of compressed sensing [15] and single-pixel cameras [19,30], it is possible to sense and recover an image with as few measurements as possible if the image to be sensed is sufficiently sparse. In order to achieve the goal of compressed image sensing (CIS) for fast image reconstruction from limited samples, the critical problems described in the following subsections need to be solved.

2.1. Sensing strategy

As an illustrative example, consider that the size of the sampling matrix required to sense an image of size 128×128 is as huge as $m \times 16384$ (assuming the use of 1D sensing, as in Eq. (1)), which occupies several Gb for 32-bit single precision floating point. Most current desktop machines cannot afford the storage overhead for such a sensing matrix. In addition, a large sensing matrix will incur significant computational overhead during the process of random projection (Eq. (1)).

There are two solutions to this problem, despite almost all compressed sensing algorithms being developed for 1D sensing. For the purpose of compressed image sensing, one common strategy adopted is to divide an image into several patches/blocks of reasonable sizes and to arrange each patch in terms of 1D form so as to adapt to the existing CS algorithms. This is called block sensing [22,23,32,33]. In fact, each block signal is treated as a 1D signal for sensing and recovery.

Although block-based image sensing seems to be feasible, it still incurs the sensor calibration problem. The other solution is to employ 2D separate sensing strategies [36,37]. That is, separate sensing is conducted along the row and column directions.

Moreover, due to the separate sensing strategy, the problem of storage overhead for storing a sampling matrix in a resource-limited sensor can be efficiently solved. As can be seen later, the storage overhead for 2D sensing of an image of size 1024×1024 is the same as that for 1D sensing of an image patch of size 32×32. Thus, it is apparent that, due to the constraint of storage overhead, 1D block-based sensing is unfavorable in sensing images/patches of large sizes. Such separable imaging operators can be realized by optical implementations [38]. Different from [36,37], which are based on Kronecker product, in this paper, the principle of linear operations [31] is employed to derive our sampling matrix and sensing strategies.

2.2. (Non-)sparsity of images

The under-deterministic problem raised by compressed sensing in solving x from y via either convex optimization or greedy algorithms can become deterministic if the sparsity of a signal x is satisfied. In practice, a considerable portion of natural images, however, is not sparse enough (with large k). In order to achieve the minimum information loss, $|y - \phi x|$, which is the common objective function of CS-based optimization mechanisms, m must be chosen to be large enough to satisfy Eq. (2), thereby leading to a larger measurement rate.

On the contrary, if m is still chosen to maintain reasonably moderate-to-small measurement rates for large k, then the reconstruction quality would be not good due to the constraint of the objective function for minimizing information loss being difficult to satisfy. This observation is consistent with our findings that the current state-of-the-art CS algorithms do not work well for compressed sensing and recovery of images.

3. Proposed method: compressed image sensing with low-frequency measurement sampling and fast recovery

In this section, we will present a new compressed image sensing algorithm via an elaborate design of sampling matrices and the study of different sensing strategies. We first describe the motivations behind our method in Section 3.1. Then, the proposed CIS algorithm, based on an elaborately designed sampling matrix, is presented in Section 3.2. We investigate how it can be conducted via 1D block sensing and 2D separate sensing. In Section 3.3, we discuss and compare two strategies for extracting important components or measurements. This paper is an extended and complete version of our prior work (one page only) [14] in terms of methodological descriptions and analyses, technical comparison with related works, and extensive simulations.

3.1. Motivations

Although it is promising to take the concept of clustered sparsity or tree-structure of transform coefficients into consideration within the compressed sensing framework, we find two weaknesses for this paradigm of compressed sensing.

The first thing we notice is that the CS inversion time still is not computationally efficient. The crux of the matter is that the inference for exploiting some specific sparsity patterns is time-consuming. In view of this, we seek an alternative so our CIS recovery time can be reduced significantly while the reconstruction quality obtained from our method and state-of-the-art algorithms can be comparable.

In order not to spend time in tracing larger transform coefficients, we propose sampling only those transform coefficients that are situated at lower frequencies. This is reasonable because, in image compression like JPEG, the higher frequency components will be quantized with larger quantization intervals while the lower frequency components can be preserved with higher priority. Inspired by the principle of JPEG compression, our CIS recovery quality will be designed to mimic JPEG compressed images. That is, we do not seek "perfect reconstruction", which is difficult to achieve, due to natural images usually not being sparse.

The most important but unique characteristic that distinguishes our method from the existing ones is that our method can only sample m important measurements with m equal to the desired degree of sparsity k and can reconstruct quickly the original signal x approximately from the sampled measurement vector y with computational complexity $O(m^2)$. We have to emphasize that the locations of so-called important measurements in an image refer to low frequencies, which are readily known and available, and are not required to be searched. The extensive simulation results indicate that our method indeed is very fast and can keep reconstruction quality up to the degree of JPEG compression approximately.

The second concern is that compressed sensing conventionally relies on the assumption of sparsity to reconstruct the original signal from (far) fewer measurements. Nevertheless, many natural images inherently containing textured components are a kind of non-sparse signals. The assumption of sparsity and the exploitation of structured sparsity do not conform to the property of less sparse signals. In view of this, another goal of our method is to target this problem. Our strategy is intuitive and empirical observations [45] again suggest that it is better to preserve important measurements sampled from lower-frequency components in some transform domains.

Our method is different from those reviewed in Section 1.2 in that we focus on the design of a sampling matrix while others are concerned with exploiting the dependency or correlation patterns among the transform coefficients.

3.2. Extremely fast compressed sensing recovery via sampling matrix design

We start from the random projection, $y = \phi x$, and observe that, if important information of x can be sampled and stored in y, then it is possible to approximately reconstruct x with fewer important measurements rapidly. The goal is feasible by designing a new and novel sampling matrix. Our method has been incorporated into single-pixel cameras for compressed image sensing.

In this subsection, we describe the proposed 1D sensing of an image patch/block and 2D separate sensing of a whole image, respectively.

3.2.1. 1D block sensing

For 1D sensing of an image patch, we introduce a 1D linear operator T and impose it to random projection to obtain:

$$Ty = T(\phi x), \tag{3}$$

where x is regarded as a small image or an image patch of reasonable size. Eq. (3) is further derived based on the principle of linear operations [31] as:

$$Ty = T(\phi x) = (T^2\phi)(Tx), \tag{4}$$

where T^2 denotes the 2D linear operator. Please refer to Appendix A for details. Note that the sizes of T's, depending on their arguments, are different in reality, but for the sake of simplifying notations, the same T is used instead.

Eq. (4) reveals that the positions at lower frequencies of transformed vector Tx indicate important transformed coefficients and Ty indicates important measurements since they are linear combinations of significant transformed coefficients.

In order to sample "important" transformed coefficients from Tx and speed recovery, we design a new sampling matrix, $(T^2\phi)^z$, by setting the last $n-m$ columns of $T^2\phi$ to be zeros. This implies that the non-zero columns of $(T^2\phi)^z$ form a full-rank matrix with rank m. Once $(T^2\phi)^z$ is built in the transform domain, it is inversely transformed back to the time/space domain and an elaborately designed sampling matrix can be expressed as:

$$\Phi = (T^2)^{-1}(T^2\phi)^z, \tag{5}$$

where our designed sampling matrix Φ involves a random matrix ϕ and 2D linear operator T^2. The pseudo code for our sampling matrix design in sampling low-frequency measurements is depicted in Algorithm 1.

Algorithm 1 Sampling matrix generation for low-frequency measurement sensing.

Input: Gaussian random matrix ϕ of size $m \times n$; $m < n$.
1: Apply 2D transform T^2 to ϕ and obtain $T^2\phi$.
2: Set the last $n-m$ columns of $T^2\phi$ to be zeros, and obtain $(T^2\phi)^z$.
3: Invert and transform the matrix $(T^2\phi)^z$ via $(T^2)^{-1}$, and obtain $\Phi = (T^2)^{-1}(T^2\phi)^z$. (See Eq. (5))
Output: Sampling Matrix, Φ, that aims to capture low-frequency measurements.

Now, Φ is stored in the sensors for the purpose of compressed sensing. According to Eqs. (4) and (5), we have the following derivations:

$$y = \Phi x \;\Rightarrow\; Ty = (T^2\Phi)(Tx) = (T^2\phi)^z(Tx). \tag{6}$$

Recall that the last $n-m$ columns of $(T^2\phi)^z$ are set to zeros. This means that we only sample the lower-frequency components in Tx by truncating the remaining higher-frequency components.

In order to speed up sparse signal recovery, let Φ^s denote the submatrix of dimensionality $m \times m$ by discarding the zero columns of $(T^2\phi)^z$ and let $(Tx)^s$ denote the $m \times 1$ vector by discarding the last $n-m$ transformed coefficients. Therefore, we can derive from Eq. (6) to obtain:

$$\begin{aligned} Ty &= \Phi^s(Tx)^s \Rightarrow \\ (\Phi^s)^{-1}Ty &= (\Phi^s)^{-1}\Phi^s(Tx)^s = (Tx)^s. \end{aligned} \tag{7}$$

It is evident that the signal x can be approximately and quickly recovered via the following signal sensing and signal recovery processes. For the purpose of sensing, the measurement y is available at the encoder via random projection in Eq. (6). For the purpose of recovery, y is first processed at the decoder via Eq. (7), and then $(\Phi^s)^{-1}Ty$ in Eq. (7) is padded with $n-m$ zero values (to obtain Tx) and inversely transformed via T^{-1}.

We have to clarify that $(\Phi^s)^{-1}T$ can be calculated in advance with a fixed sampling rate for use at the decoder. Therefore, the computational complexity of our CIS recovery algorithm merely comes from processing y via $(\Phi^s)^{-1}Ty$.

Basically, our method proposes executing conventional DCT or wavelet transform within the framework of compressed sensing if T is selected to be DCT or wavelet transform. Nevertheless, it is not suitable to just compute the transformation (via T) of a signal to be sensed and store a proportion of low-frequency components (by discarding high-frequency components, as done in this paper). This is because some parts to be deleted still are sampled in advance, which follows the same mechanism of conventional sampling. On the contrary, the ultimate goal of our method can, in fact, achieve the same effect of sampling+compression within the framework of compressed sensing, as explained in the previous paragraphs.

3.2.2. 2D separate sensing

Here, we investigate how to directly sense a whole 2D image via separate sensing in order to alleviate storage overhead of storing a sampling matrix. According to Eq. (A.3) of Appendix A, 2D transform is conducted on the 2D image, which is no longer

divided into patches and is no longer arranged in terms of 1D form. Therefore, we can get 2D sensing of an image $x_{n\times n}$ via the sampling matrix $\phi_{m\times n}$ as:

$$y_{m\times m} = \phi_{m\times n} x_{n\times n} \phi^t_{n\times m}, \tag{8}$$

where $\phi^t_{n\times m}$ denotes the transpose of $\phi_{m\times n}$. Eq. (8) also reveals the measurement rate of $\frac{m^2}{n^2}$ in two dimensional sensing.

Similar to Eq. (A.3), let $S_{m\times m}$ and $S_{n\times n}$ stand for two 1D linear operators with respective transpose represented as $S^t_{m\times m}$ and $S^t_{n\times n}$. We have $S_{n\times n}S^t_{n\times n} = I_{n\times n} = S^t_{n\times n}S_{n\times n}$, where $I_{n\times n}$ denotes an identity matrix of dimensionality $n\times n$. Then, we can further derive by imposing the 2D transform on the measurement matrix $y_{m\times m}$ of Eq. (8) to attain:

$$\begin{aligned} T^2(y) &= S_{m\times m} y_{m\times m} S^t_{m\times m} \\ &= S_{m\times m}\phi_{m\times n} x_{n\times n} \phi^t_{n\times m} S^t_{m\times m} \\ &= S_{m\times m}\phi_{m\times n}(S^t_{n\times n}S_{n\times n}) x_{n\times n} \\ (S^t_{n\times n}S_{n\times n})\phi^t_{n\times m}S^t_{m\times m} &= (S_{m\times m}\phi_{m\times n}S^t_{n\times n})(S_{n\times n}x_{n\times n}S^t_{n\times n}) \\ (S_{n\times n}\phi^t_{n\times m}S^t_{m\times m}) &= T^2(\phi)T^2(x)T^2(\phi^t). \end{aligned} \tag{9}$$

Then, the sampling matrix can be derived in a way similar to Eqs. (4) and (5) for 2D separate sensing. More specifically, we design a new 2D separable sensing matrix by setting the last $n-m$ columns of $T^2(\phi)$ to be zeros and setting the last $n-m$ rows of $T^2(\phi^t)$ to be zeros with the remaining derivations similar to 1D sensing. Another merit of 2D sensing is that it enables compressed sensing of large images without resorting to block-based sensing.

3.3. Strategies for extracting important components/measurements

As described above, our sampling matrix is designed based on the principle that the important components/measurements of an image to be sensed are captured in advance at the encoder. Thus, the elaborately designed sampling matrix will benefit the decoder in reconstructing the sensed images with high quality in a computationally efficient manner because important components have been sieved out at the encoder. Such a characteristic of our method is different from that of greedy algorithms like OMP, which finds the significant components (or transform coefficients) at the decoder via a greedy strategy.

On the other hand, as mentioned earlier, our method keeps important information at the positions of lower frequencies (without preserving high-frequency details), which, however, are not guaranteed to contain information of the images to be sensed. OMP, however, tries to find larger transform coefficients in a signal but, unfortunately, the significant coefficients may be obscure and difficult to select due to the effect of non-sparsity of signals. The potential weakness of OMP can be understood from the process of its greedy recovery. It can be seen in Section 5 that OMP exhibits comparable reconstruction results but performs far faster than other state-of-the-art CS algorithms. Nevertheless, it is important to note that our method significantly outperforms OMP, which has been considered to be rather efficient in terms of recovery speed and quality.

4. Discussions

In this section, we will discuss the following issues in order to better reveal the characteristics of the proposed compressed image sensing method: A. the difference between our method and frequency/space-frequency transforms, B. the computational complexity of CS recovery, C. the mutual incoherence between the sampling matrix and dictionary, D. sparsity vs. reconstruction quality, and E. approximate recovery vs. perfect reconstruction.

4.1. Difference between our method and frequency/space-frequency transforms

It is worth noting that the principle of designing Φ in Eq. (5) still starts with the random sampling matrix ϕ conventionally adopted in CS, which is selected as the foundation of our design. The characteristic unique to our method is that a 2D linear operator T^2, as described in Appendix A, is applied to ϕ, followed by setting the last $n-m$ columns to be zeros. In addition, both our CIS method and frequency-space transforms are capable of capturing lower-frequency components. Our CIS method, however, is built within the framework of CS while other conventional frequency or frequency-space transforms are not.

4.2. Computational complexity of CS recovery

The principle of our method is to preserve the top k-lowest frequency components of Tx. We have $m = k$, and only two linear transforms, $(\Phi^s)^{-1}T$ and T^{-1}, as described in the last paragraph of Section 3.2, are required for approximate signal recovery. Thus, the computational complexity of recovery is in the order of $O(m^2)$. As we shall see in Section 5, our method is the fastest CS recovery algorithm among the state-of-the-art methods used for comparison.

4.3. Mutual incoherence

In the literature, a sampling matrix is usually a Gaussian random matrix or a Bernoulli random matrix taking the value ± 1 with equal probability. A good sampling matrix, which preserves incoherence [7] with a dictionary or transform basis, is

desired for efficient recovery. Mutual incoherence leads to perfect recovery with a higher probability under the condition that $m \geq ck \cdot O(\log(n/k))$ holds.

Our results show that the random matrix conventionally adopted in the compressed sensing literature is more incoherent than the sampling matrix proposed in this paper with an orthonormal matrix (as a dictionary). We, however, have to point out that, since the sparsity assumption rarely is satisfied with respect to natural images, which are usually less sparse, our method theoretically sacrifices mutual incoherence between the sampling matrix and the corresponding dictionary but is practically feasible and extremely fast in reconstructing images with better quality.

4.4. Sparsity vs. reconstruction quality

Following the above descriptions, the impact of sparsity on reconstruction quality is discussed here. According to our empirical observations, almost all compressed sensing algorithms proposed so far fail to reconstruct a natural image with quality (in terms of PSNR or SSIM [43]) superior to simple interpolation. For example, consider a natural image, Peppers, of size 50×50. When a bicubic interpolation technique is adopted to generate an enlarged image of size 100×100, the obtained PSNR is 26.76dB (between the original Peppers and the resultant interpolated Peppers). Nevertheless, within the framework of compressed sensing, if the original image has the size of 100×100 and the number of corresponding measurements is 50×50, the image obtained from the measurements via OMP has PSNR 17.06dB, which is far lower than the one obtained using bicubic interpolation. Similar results can also be observed from many other CS recovery algorithms.

In view of these results, we propose a new CS recovery algorithm that does not follow the convention of CS. More specifically, our method is an alternative to CS in the sense that we seek to pursue approximate recovery instead of theoretical perfect reconstruction, which is the ultimate goal of CS. Our observations and results are supported strongly by the fact that natural images are not highly or sufficiently sparse, which somewhat violates the fundamental assumption of CS. Nevertheless, we also have to clarify that the random sampling matrix inherently used in compressed sensing may be better than the one presented in our paper for those applications (e.g., [13,28]) exhibiting sufficient sparsity.

4.5. Approximate recovery vs. perfect reconstruction

It is known that, under the constraint of $m \geq ck \cdot O(\log \frac{n}{k})$, conventional compressed sensing methods can achieve perfect reconstruction if the available number of measurements and the sparsity of the signal to be reconstructed satisfy the above relation. Nevertheless, perfect reconstruction usually is not practical because, for many signals with large k, sometimes only small M is available, leading to violation of $m \geq ck \cdot O(\log \frac{n}{k})$.

In this paper, we aim to explore a more practical solution to CS recovery. More specifically, based on the elaborate design of the sampling matrix, we do not seek perfect reconstruction but seek approximate reconstruction. Depending on the selected m, which is assumed to be k, extremely fast and approximate reconstruction can be achieved. Approximate reconstruction has been useful for many CS-based image/video applications [18,27,35,47,48].

In this paper, as described in Appendix A, a 1D DCT or Haar matrix is exploited as T to design the sampling matrix Φ such that the original signal x can be approximately reconstructed from as many measurements as the number of transform coefficients sampled via Eqs. (6) and (7).

5. Simulation results

Several simulations were conducted to verify the performance of the proposed compressed image sensing method along with different sensing strategies in terms of reconstruction quality and speed.

State-of-the-art CS algorithms [15], including orthogonal matching pursuit (OMP) [41], Lasso, TS-BCS-MCMC [24], TS-BCS-VB [25], and model-based CS (MCS) [3],[1] were chosen for comparison under different measurement rates (MRs). There are two signal models, wavelet trees and block sparsity, used in MCS. In our simulations conducted here, 2D wavelet trees and block sparsity in [3] were adopted. The default settings of all source codes were employed in our simulations to better guarantee the good performance of the aforementioned methods for fair comparison.

All simulations were conducted in Matlab 7.11 (R2010b) with an Intel CPU Core i7 930 (2.80 GHz) and 6 GB RAM under OS Windows 7 Enterprise edition 64-bit. For simulations of image sensing, several images with different sizes and sparsities, including Baboon, Barbara, Cameraman, Flintstones, Lena, and Peppers, were adopted.

For 1D sensing, all of the methods used for comparison were conducted in a block-wise manner. That is, image sensing and recovery were executed for each 32×32 block [32]. In our method, as derived in Appendix A, two Haar matrices, $H_{1024\times 1024}$ and $H_{(1024\times MR)\times(1024\times MR)}$ were employed. If RAM is larger, however, then the block size can be permitted to be larger. If the block size is equal to its corresponding image size, we call it a whole image sensing.

For 2D sensing in our method, we currently only employ 2D DCT because the design of 2D wavelet filters in terms of matrix forms needs to be studied further. In addition, 2D sensing was verified in terms of whole image sensing (an image is regarded as a block) in simulations, where we particularly demonstrate that whole large-sized image sensing is feasible for 2D sensing.

[1] Our simulations conducted on MCS show that block sparsity is better than 2D tree in both reconstruction quality and speed.

Table A.1
Recovery quality comparison of CS algorithms under different measurement rates (MRs) for Cameraman image.

Methods	Metrics	MR (1.56%)	MR (6.25%)	MR (12.5%)	MR (25.0%)
OMP	PSNR(dB)	7.41	15.86	17.47	20.06
(Sparsify toolbox)	SSIM	0.04	0.35	0.41	0.53
Lasso	PSNR(dB)	13.82	17.24	19.04	21.76
(Sparsify lab)	SSIM	0.33	0.43	0.49	0.62
Model-based CS	PSNR(dB)	6.38	9.18	14.92	21.99
(Block sparsity) [3]	SSIM	0.06	0.08	0.26	0.64
TS-BCS-VB	PSNR(dB)	17.11	19.10	20.20	22.62
[25]	SSIM	0.46	0.56	0.63	0.73
TS-BCS-MCMC	PSNR(dB)	16.87	19.33	20.59	23.11
(Wavelet tree) [24]	SSIM	0.46	0.55	0.64	0.74
TS-BCS-MCMC	PSNR(dB)	7.68	18.10	21.08	23.87
(DCT tree) [24]	SSIM	0.08	0.48	0.60	0.75
Our method (1D sensing:	PSNR(dB)	18.32	19.84	22.15	24.82
Haar wavelet-based)	SSIM	0.57	0.64	0.73	0.82
Our method (1D sensing:	PSNR(dB)	18.42	19.38	22.36	25.60
DCT-based)	SSIM	0.58	0.61	0.69	0.79
Our method (2D sensing:	PSNR(dB)	20.74	23.26	24.94	27.46
DCT-based)	SSIM	0.58	0.70	0.78	0.87

Table A.2
Recovery quality comparison of CS algorithms under different measurement rates (MRs) for Barbara image.

Methods	Metrics	MR (1.56%)	MR (6.25%)	MR (12.5%)	MR (25.0%)
OMP	PSNR(dB)	15.84	18.00	20.26	23.12
(Sparsify toolbox)	SSIM	0.19	0.31	0.47	0.64
Lasso	PSNR(dB)	14.76	19.15	21.70	24.89
(Sparsify lab)	SSIM	0.19	0.38	0.54	0.72
Model-based CS	PSNR(dB)	7.13	9.81	16.32	23.82
(Block sparsity) [3]	SSIM	0.04	0.06	0.31	0.69
TS-BCS-VB	PSNR(dB)	18.46	20.31	21.37	22.35
[25]	SSIM	0.37	0.46	0.54	0.62
TS-BCS-MCMC	PSNR(dB)	18.19	20.85	22.04	23.31
(Wavelet tree) [24]	SSIM	0.36	0.48	0.55	0.65
TS-BCS-MCMC	PSNR(dB)	8.81	18.07	22.49	24.57
(DCT tree) [24]	SSIM	0.06	0.43	0.59	0.73
Our method (1D sensing:	PSNR(dB)	19.25	21.18	22.98	25.49
Haar wavelet-based)	SSIM	0.48	0.56	0.65	0.79
Our method (1D sensing:	PSNR(dB)	19.31	20.27	22.60	25.41
DCT-based)	SSIM	0.48	0.52	0.61	0.77
Our method (2D sensing:	PSNR(dB)	22.47	23.92	24.54	25.70
DCT-based)	SSIM	0.56	0.67	0.73	0.82

It should be noted that all other algorithms used for comparison employ 1D sensing and are not applicable to the whole image sensing due to intractable storage and computation overhead, as discussed in Section 2.1.

5.1. Reconstruction quality

The recovery quality is measured in terms of PSNR (in dB) and structural similarity (SSIM, $0 \leq \text{SSIM} \leq 1$) indexing [43], respectively, where bigger is better. Tables A.1 and A.2 show the comparisons of reconstruction qualities under different measurement rates (MRs) for images of different sparsities, including a sparse image, Cameraman, of size 256×256 and a less-sparse image, Barbara, of size 512×512. It is surprising to find from Tables A.1 and A.2 that our method significantly outperforms or is comparable to all of the algorithms used for comparison no matter whether either structured sparsity or tree structure is taken into consideration or not. DCT- and Haar wavelet-based 1D sensing in our method exhibit comparable reconstruction results. Nevertheless, our method requires considerably less time than the methods used for comparison, as will be shown in the next subsection.

Figs. A.1 and A.2 show the reconstructed Cameraman and Barbara images under the measurement rate 12.5% using the CS algorithms shown in Tables A.1 and A.2, respectively, for perceptual verification. We can observe that, although our algorithms do not aim to keep high-frequency details, the reconstruction results indeed show that the results obtained from our Haar wavelet-based 1D sensing and DCT-based 2D separate sensing exhibit better perceptual qualities than others.

It is also interesting to find that the reconstruction quality of the Barbara image is better than that of the Cameraman image in terms of PSNR in most cases, which violates the intuition that Cameraman is sparser than Barbara in the transform (either

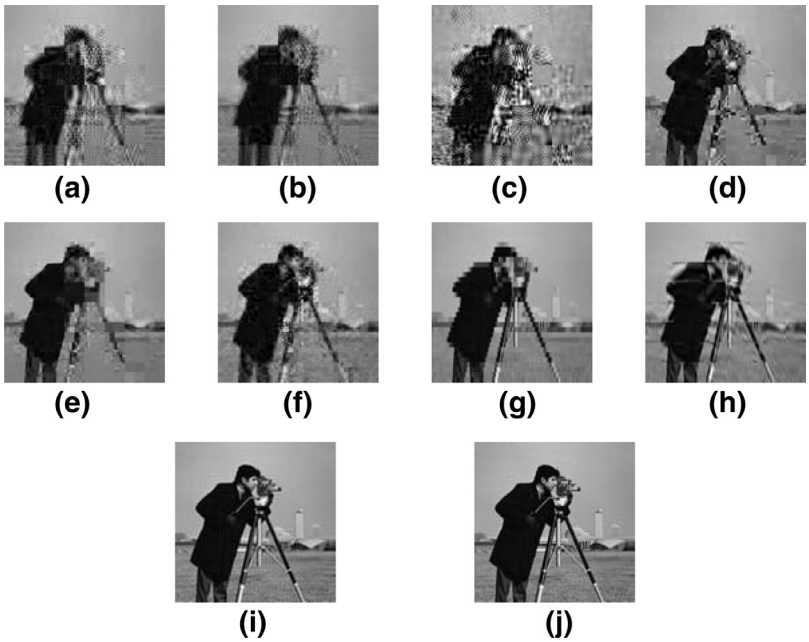

Fig. A.1. CS recovered Cameraman images with MR 12.5% using (a) OMP; (b) Lasso; (c) MCS [3] with block sparsity; (d) TS-BCS-VB [25]; (e) TS-BCS-MCMC [24] with wavelet tree; (f) TS-BCS-MCMC [24] with DCT tree; (g) our method with Haar wavelet-based 1D sensing; (h) our method with DCT-based 1D sensing; (i) our result with DCT-based 2D separate (whole image) sensing; and (j) original Cameraman. All of the obtained values of PSNR and SSIM can be found in Table A.1.

Table A.3
Recovery speed (in seconds) comparison of CS algorithms under different measurement rates (MRs) for Cameraman image.

Methods	Time	MR (1.56%)	MR (6.25%)	MR (12.5%)	MR (25.0%)
OMP	Exe.	0.37	0.69	1.33	3.61
(Sparsify toolbox)	CPU	1.53	2.78	5.35	14.43
Lasso	Exe.	1.94	8.65	24.06	90.10
(Sparsify lab)	CPU	7.66	34.63	96.21	359.46
Model-based CS [3]	Exe.	0.30	0.52	0.77	2.03
(Block sparsity)	CPU	1.19	2.07	3.04	8.11
TS-BCS-VB [25]	Exe.	226.03	244.69	253.46	280.38
	CPU	230.51	941.64	972.90	1071.54
TS-BCS-MCMC [24]	Exe.	2266.11	2440.49	2503.80	2742.34
(Wavelet tree)	CPU	2267.32	9235.26	9450.31	10289.34
TS-BCS-MCMC [24]	Exe.	2550.03	2737.74	2892.09	2982.26
(DCT tree)	CPU	2548.93	10270.53	10819.22	11109.16
Our method (1D sensing:	Exe.	0.02	0.02	0.02	0.02
Haar wavelet-based)	CPU	0.06	0.06	0.06	0.12
Our method (1D sensing:	Exe.	0.03	0.02	0.02	0.02
DCT-based)	CPU	0.06	0.05	0.06	0.06
Our method (2D sensing:	Exe.	0.06	0.07	0.07	0.07
DCT-based)	CPU	0.25	0.31	0.31	0.36

DCT or wavelet) domain. Nevertheless, when SSIM is adopted as the image quality evaluation metric, the obtained results indeed indicate that Cameraman is better reconstructed than Barbara for our method and those CS methods exploiting sparsity patterns.

5.2. Reconstruction speed

The reconstruction speed is measured in terms of execution time (abbreviated as Exe.) and CPU time. In Tables A.3 and A.4, we only provide the comparison of reconstruction speed under different measurement rates (MRs) for the Cameraman and Barbara images, respectively. Nevertheless, similar results can also be observed for other images with different sparsity levels. As the results indicate, our CIS method finds its usefulness in real-time image sensing and recovery due to its extremely fast CS recovery.

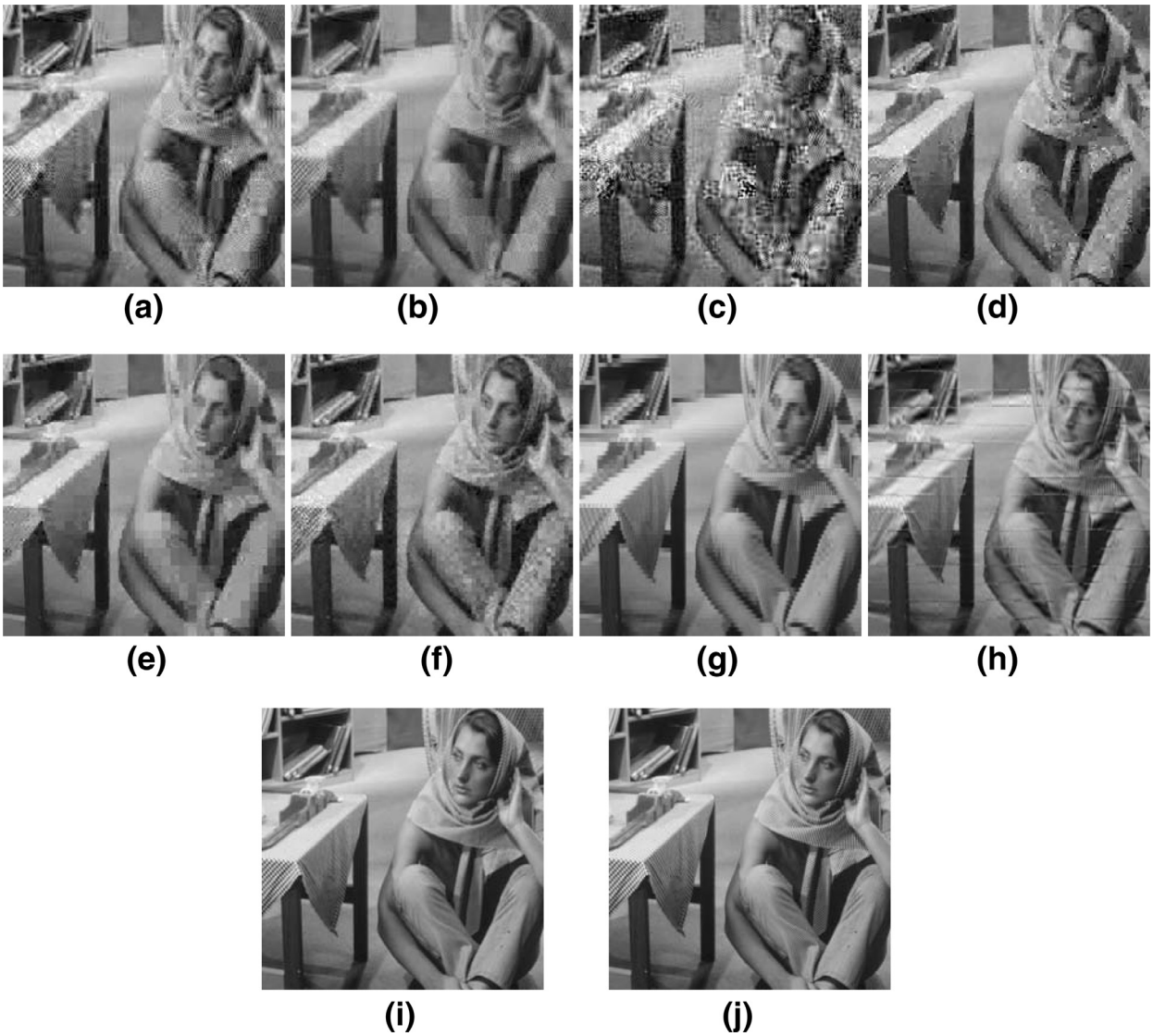

Fig. A.2. CS recovered Barbara images with MR 12.5% using (a) OMP; (b) Lasso; (c) MCS [3] with block sparsity; (d) TS-BCS-VB [25]; (e) TS-BCS-MCMC [24] with wavelet tree; (f) TS-BCS-MCMC [24] with DCT tree; (g) our method with Haar wavelet-based 1D sensing; (h) our method with DCT-based 1D sensing; (i) our result with DCT-based 2D separate (whole image) sensing; and (j) original Barbara. All of the obtained values of PSNR and SSIM can be found in Table A.2.

5.3. Fast compressed sensing and recovery of large images

Fast compressed sensing and recovery of large-scale images is still challenging for the existing CS algorithms that employ a 1D sensing strategy. The feasibility of our 2D separate (whole image) sensing algorithm was demonstrated using a Shepp–Logan image of size 2048×2048 and an airport image of size 1024×1024. The reconstruction qualities obtained using our 1D sensing and 2D separate sensing strategies for the Shepp–Logan image are shown in Fig. A.3 under the extremely low measurement rate 1.5625%. Since Shepp–Logan is rather sparse, low measurement rates are sufficient to yield good recovery results. In particular, Fig. A.3(d) exhibits non-blocky effects for 2D separate sensing.

On the other hand, for a non-sparse image like Airport, as shown in Fig. A.4(a), under the measurement rate 1.5625%, only the reconstruction result obtained using 2D separate sensing is acceptable. Therefore, we further show the recovery results in Fig. A.4(c)–(h) under the measurement rates 6.25 and 12.5%. It is not surprising to observe that more measurements are required for non-sparse signals to yield acceptable reconstruction results.

The execution time for the three proposed sensing strategies in recovering the large images shown here is between 2 and 3 s for measurement rates ranging from 1.5625 to 50%.

Table A.4
Recovery speed (in seconds) comparison of CS algorithms under different measurement rates (MRs) for Barbara image.

Methods	Time	MR (1.56%)	MR (6.25%)	MR (12.5%)	MR (25.0%)
OMP	Exe.	0.32	0.70	1.36	3.56
(Sparsify toolbox)	CPU	1.25	2.81	5.43	14.20
Lasso	Exe.	1.88	8.97	24.22	85.18
(Sparsify lab)	CPU	7.60	35.83	96.86	339.83
Model-based CS [3]	Exe.	1.02	1.46	2.66	6.54
(Block sparsity)	CPU	4.13	5.82	10.73	26.18
TS-BCS-VB [25]	Exe.	219.94	244.47	247.23	267.77
	CPU	220.66	937.05	949.77	1022.48
TS-BCS-MCMC [24]	Exe.	2243.21	2492.97	2580.08	2742.51
(Wavelet tree)	CPU	2241.52	9411.93	9213.51	10279.05
TS-BCS-MCMC [24]	Exe.	2585.22	2842.79	2848.40	2978.40
(DCT tree)	CPU	2579.59	10650.77	10670.41	11109.13
Our method (1D sensing:	Exe.	0.02	0.02	0.02	0.03
Haar wavelet-based)	CPU	0.06	0.06	0.06	0.12
Our method (1D sensing:	Exe.	0.02	0.02	0.02	0.02
DCT-based)	CPU	0.06	0.05	0.06	0.06
Our method (2D sensing:	Exe.	0.06	0.06	0.06	0.07
DCT-based)	CPU	0.25	0.27	0.25	0.31

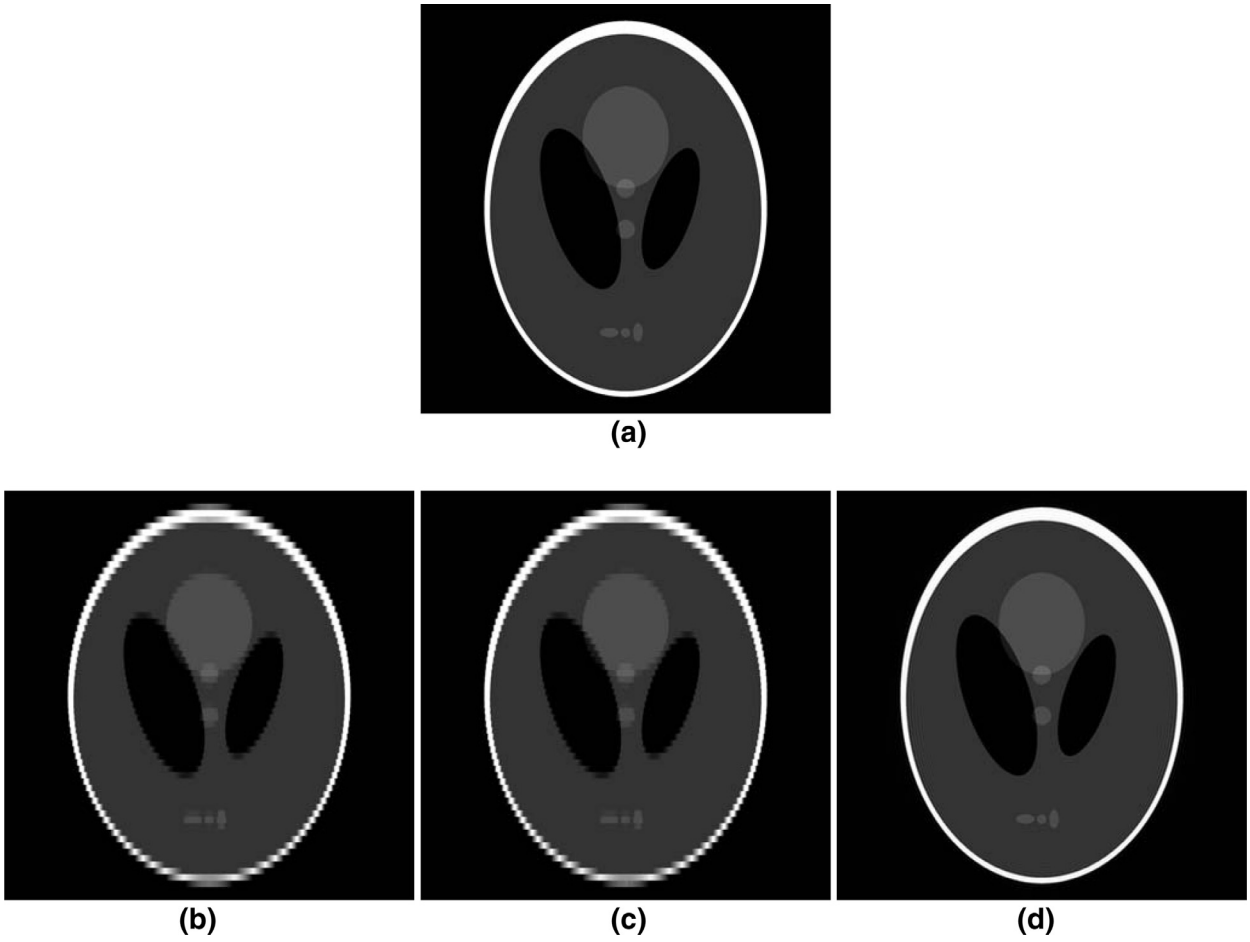

Fig. A.3. CS recovered Shepp–Logan images with MR 1.5625%: (a) original image of size 2048 × 2048; (b) our result with Haar wavelet-based 1D sensing (PSNR: 22.18dB; SSIM: 0.95); (c) our result with DCT-based 1D sensing (PSNR: 22.22dB; SSIM: 0.94); and (d) our result with DCT-based 2D separate sensing (PSNR: 28.69dB; SSIM: 0.92).

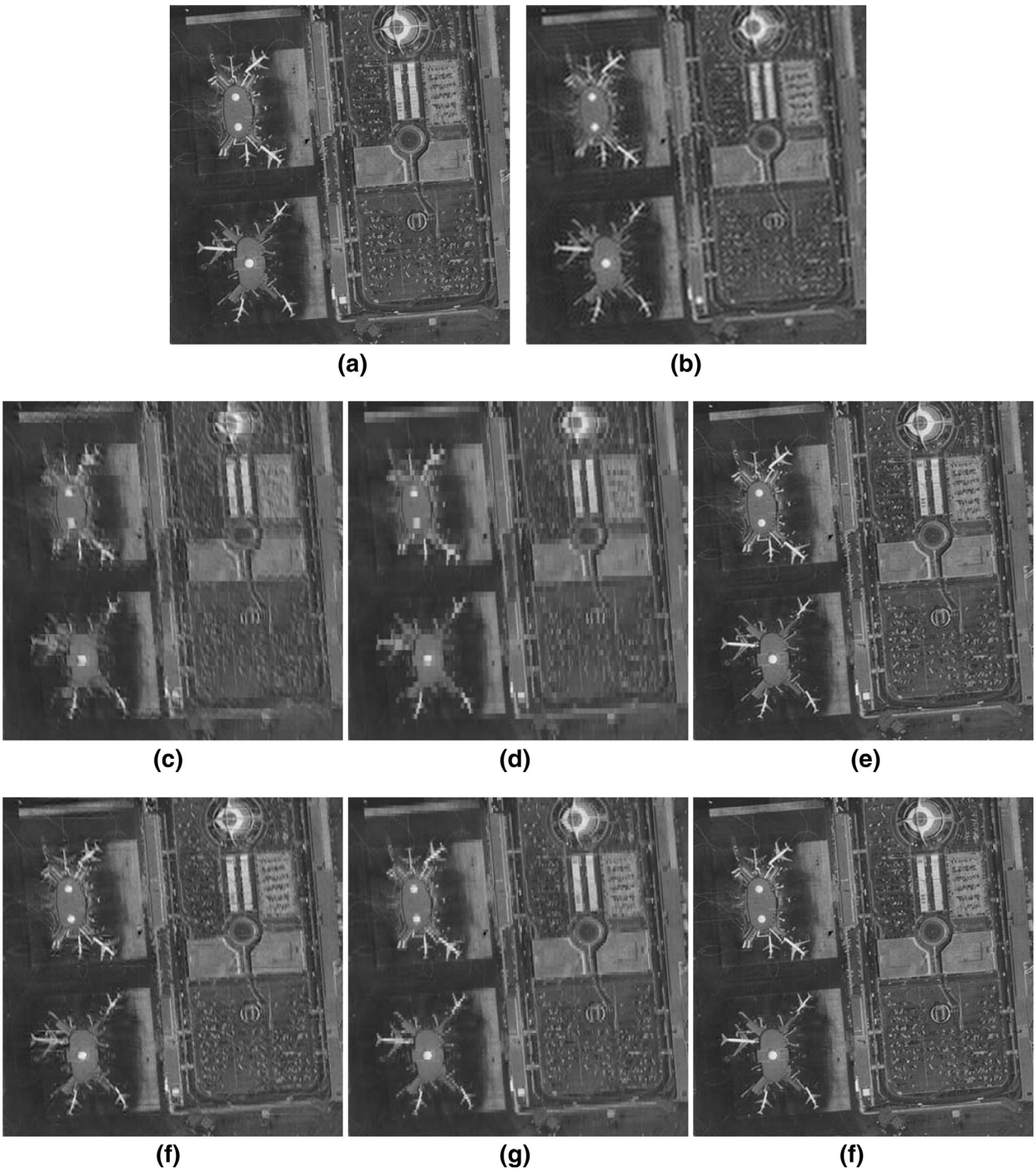

Fig. A.4. CS recovered Airport images: (a) original image of size 1024×1024; (b) recovered result under MR = 1.5625% via 2D separate sensing; (c)–(e) recovered results under MR = 6.25% via Haar wavelet-based 1D sensing, and DCT-based 1D sensing and 2D sensing, respectively; (f)–(h) recovered results under MR = 12.5% via Haar wavelet-based 1D sensing, and DCT-based 1D sensing and 2D sensing, respectively.

6. Conclusions and future work

Fast and accurate compressed sensing recovery is still a challenging issue, and has received considerable attention in the literature. In this paper, we do not follow the tradition of imposing certain sparsity patterns on a compressed sensing recovery algorithm. On the contrary, we propose a novel sampling matrix for the purpose of preserving important measurements in compressed sensing of images. Under this circumstance, extremely fast compressed image sensing recovery with a closed-form

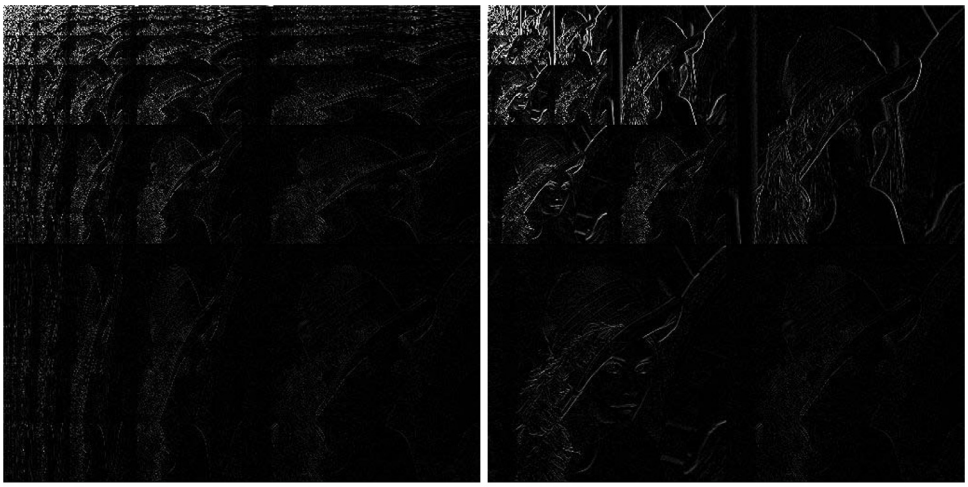

Fig. A.5. Comparison of Haar Transforms: (a) Haar matrix decomposition; (b) conventional Haar decomposition.

solution of computational complexity $O(m^2)$ can be achieved. The reconstruction accuracy of our method also is verified to be comparable to or outperform the state-of-the-art CS algorithms.

We have also studied 1D sensing and 2D separate sensing strategies for 1D signals and 2D images, respectively. Compared to 1D sensing, 2D separate sensing is found to be particularly feasible in compressed sensing of large-scale images in terms of storage and computation overhead reduction and reconstruction quality improvement.

The issues, including the impact of noisy measurements on our methods and the exploration of CS characteristics for our method, deserve further study.

Acknowledgment

This work was supported by National Science Council, Taiwan, under grants NSC 97-2628-E-001-011-MY3 and NSC 100-2628-E-001-005-MY2.

Appendix A. Linear transforms on random projection

The linear operator introduced in [31] is employed to derive the linear transform relationship among x, y, and ϕ. Let S be a linear orthogonal transform operator, and let x denote a 1D signal with dimensionality $m \times 1$. The vector X of transform coefficients corresponding to x is represented as: $X = Sx$, where S is a matrix with dimensionality $m \times m$. The original signal x can be reconstructed as:

$$\hat{x} = S^t X = S^t S x, \tag{A.1}$$

where S^t is the transpose of S, and $S^t S = S S^t = I$. Similarly, if the 2D case, i.e., x of size $m \times m$, is considered, then we have:

$$\begin{aligned} &X = SxS^t, \\ &\hat{x} = S^t X S = S^t S x S^t S. \end{aligned} \tag{A.2}$$

According to the above characteristics, the linear transform relationship among x, y, and ϕ can be derived under the cases of DCT and discrete Haar wavelet transform (DHWT), respectively. For the DCT case, let S_m and S_n denote $m \times m$ and $n \times n$ DCT matrices, respectively. Also, let $T[\cdot]$ and $T^2[\cdot]$ denote the 1D-DCT and 2D-DCT operations, respectively. Starting from $y = \phi x$, we can derive:

$$\begin{aligned} &S_m y = S_m \phi x = (S_m \phi S_n^t)(S_n x) \Rightarrow \text{corresponding to} \\ &T[y] = T[\phi x] = T^2[\phi] T[x], \end{aligned} \tag{A.3}$$

where for notational simplicity, the size of T can be found depending on its argument. Thus, Eq. (A.3) explains the rationality of Eq. (4).

For the wavelet case, the simple structure inherent in the Haar wavelet is adopted. Let H denote a 2D Haar wavelet transform. Then, the Haar transform of x can be derived, similar to Eq. (A.2), as:

$$X = HxH^t. \tag{A.4}$$

It is, however, not straightforward to use the conventional Haar wavelet to achieve Eq. (A.4) because, for example, if the 8×8 Haar wavelet, as shown in Eq. (A.5), is used, only one wavelet decomposition is allowed.

$$H = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}. \tag{A.5}$$

To handle this problem, another type of Haar transform suitable for Eq. (A.4) is designed as:

$$H = \begin{bmatrix} c^3 & c^3 & c^3 & c^3 & c^3 & c^3 & c^3 & c^3 \\ c^3 & c^3 & c^3 & c^3 & -c^3 & -c^3 & -c^3 & -c^3 \\ c^2 & c^2 & -c^2 & -c^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c^2 & c^2 & -c^2 & -c^2 \\ c & -c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & -c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c & -c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c & -c \end{bmatrix}, \tag{A.6}$$

where $c = \frac{\sqrt{2}}{2}$.

Different from conventional wavelet transforms, it is worth noting that the use of the Haar matrix, shown in Eq. (A.6), allows multi-scale wavelet decomposition to be finished within one matrix operation. Fig. A.5(a) and (b), respectively, illustrate the results of Haar wavelet decomposition using the designed Haar matrix (via Eqs. (A.4) and (A.6)) and the conventional Haar filter. It can be observed that the Haar matrix will continue to decompose the LH-band and HL-band at each scale, while the conventional Haar wavelet will not. Furthermore, we want to clarify that Fig. A.5 merely is used to illustrate the difference between the Haar matrix and conventional Haar wavelet. In fact, the Haar matrix is used in our method to decompose the sampling matrix, as indicated in Eq. (4).

References

[1] S.D. Babacan, R. Molina, A.K. Katsaggelos, Bayesian compressive sensing using laplace priors, IEEE Trans. Image Process. 19 (1) (2010) 53–63.
[2] R. Baraniuk, V. Cevher, M.B. Wakin, Low-dimensional models for dimensionality reduction and signal recovery: a geometric perspective, Proc. IEEE 98 (6) (2010) 959–971.
[3] R. Baraniuk, V. Cevher, M.F. Duarte, C. Hedge, Model-based compressive sensing, IEEE Trans. Inf. Theory 56 (4) (2010) 1982–2001.
[4] T. Blumensath, M.E. Davies, Iterative hard thresholding for compressed sensing, Appl. Comput. Harmonic Anal. 27 (3) (2009) 265–274.
[5] C.F. Caiafa, A. Cichocki, Computing sparse representations of multidimensional signals using Kronecker bases, Neural Comput. J. 25 (1) (2013) 186–220.
[6] C.F. Caiafa, A. Cichocki, Stable, robust and super fast reconstruction of tensors using multi-way projections, IEEE Trans. Signal Process. 63 (3) (2015) 780–793.
[7] E. Candes, J. Romberg, Sparsity and incoherence in compressive sampling, Inverse Probl. 23 (3) (2007) 969–985.
[8] P.J. Carrgues, Ph. D. dissertation, 2009.
[9] R.E. Carrillo, L.F. Polania, K.E. Barner, Iterative algorithms for compressed sensing with partially known support, Proceedings of the IEEE ICASSP, 2010.
[10] R.E. Carrillo, L.F. Polania, K.E. Barner, Iterative hard thresholding for compressed sensing with partially known support, Proceedings of the IEEE ICASSP, 2011, pp. 4028–4031.
[11] V. Cevher, C. Hedge, M.F. Duarte, R.G. Baraniuk, Sparse signal recovery using Markov random fields, in: Proceedings of the Workshop on Neural Information Processing Systems, 2008.
[12] V. Cevher, P. Indyk, C. Hedge, R.G. Baraniuk, Recovery of clustered sparse signals from compressive measurements, in: Proceedings on Sampling Theory and Applications (SAMPTA), 2009.
[13] H.-C. Chen, H.-T. Kung, D. Vlah, D. Hague, M. Muccio, B. Poland, Collaborative compressive spectrum sensing in a UAV environment, in: Proceedings of the Military Communications Conference (MILCOM), November 2011.
[14] H.-W. Chen, C.-S. Lu, S.-C. Pei, Fast compressive sensing recovery with transform-based sampling, in: Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations, June 27–30, Edinburgh, UK, 2011, p. 110.
[15] Compressive sensing resources: "http://dsp.rice.edu/cs".
[16] S. Dekel, Adaptive Compressed Image Sensing Based on Wavelet-Trees, 2008, Report.
[17] S. Deutsch, A. Averbuch, S. Dekel, Adaptive compressed image sensing based on wavelet modeling and direct sampling, in: Proceedings of the SAMPTA, 2009.
[18] T.T. Do, Y. Chen, D.T. Nguyen, N. Nguyen, L. Gan, T.D. Tran, Distributed compressed video sensing, in: Proceedings of the IEEE International Conference on Image Processing, 2009, pp. 1393–1396.
[19] M.F. Duarte, M.A. Davenport, D. Takhar, J.N. Laska, T. Sun, K.F. Kelly, R.G. Baraniuk, Single-pixel imaging via compressive sampling, IEEE Signal Process. Mag. 25 (2008) 83–91.
[20] Y.C. Eldar, M. Mishali, Robust recovery of signals from a structured union of subspaces, IEEE Trans. Inf. Theory 55 (11) (2009) 5302–5316.
[21] Y.C. Eldar, P. Kuppinger, H. Bolcskei, Block-sparse signals: uncertainty relations and efficient recovery, IEEE Trans. Signal Process. 58 (6) (2010) 3042–3054.
[22] L. Gan, Block compressed sensing of natural images, in: Proceedings of the Conference on Digital Signal Processing (DSP), July 2007.Cardiff, UK
[23] L. Gan, T.T. Do, T.D. Tran, Fast compressive imaging using scrambled block Hadamard ensemble, in: Proceedings of the 16th European Signal Processing Conference (EUSIPCO), Lausanne, Switzerland, August 2008.
[24] L. He, L. Carin, Exploiting structure in wavelet-based Bayesian compressed sensing, IEEE Trans. Signal Process. 57 (9) (2009) 3488–3497.
[25] L. He, H. Chen, L. Carin, Tree-structured compressive sensing with variational Bayesian analysis, IEEE Signal Process. Lett. 17 (3) (2010) 233–236.
[26] J. Huang, T. Zhang, D. Metaxas, Learning with structured sparsity, in: Proceedings of the International Conference on Machine Learning (ICML), 2009.
[27] L.W. Kang, C.S. Lu, Distributed compressive video sensing, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Taipei, Taiwan, 2009, pp. 1169–1172.

[28] H.-T. Kung, C.-K. Lin, D. Vlah, Cloudsense: continuous fine-grain cloud monitoring with compressive sensing, in: Proceedings of the 3rd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud), June 2011.
[29] C.-S. Lu, W.-J. Liang, Fast compressive sensing of high-dimensional signals with tree-structure sparsity pattern, in: Proceedings of the IEEE ChinaSIP, 2014, pp. 738–742.
[30] F. Magalhaes, F.M. Araujo, M.V. Correia, M. Abolbashari, F. Farahi, Active illumination single-pixel camera based on compressive sensing, J. Appl. Opt. 50 (4) (2011) 405–414.
[31] N. Merhav, V. Bhaskaran, A transform domain approach to spatial domain image, in: Proceedings of the HPL-94-116, Technion City, Haifa 32000, Israel, 1994.
[32] S. Mun, J.E. Fowler, Block compressed sensing of images using directional transforms, in: Proceedings of the IEEE International Conference on Image Processing, 2009, pp. 3021–3024.
[33] S. Mun, J.E. Fowler, Residual reconstruction for block-based compressed sensing of video, in: Proc. Data Compression Conference (DCC), 2011, pp. 183–192.
[34] D. Needell, J. Tropp, Cosamp: iterative signal recovery from incomplete and inaccurate samples, Appl. Comput. Harmonic Anal. 26 (3) (2009) 301–321.
[35] J. Prades-Nebot, Y. Ma, T. Huang, Distributed video coding using compressive sampling, in: Proceedings of the Picture Coding Symposium, 2009.
[36] Y. Rivenson, A. Stern, Compressed imaging with a separable sensing operator, IEEE Signal Process. Lett. 16 (6) (2009) 449–452.
[37] Y. Rivenson, A. Stern, Practical compressive sensing of large images, in: Proceedings of the 16th International Conference on Digital Signal Processing, July 2009.
[38] R. Robucci, L.K. Chiu, J. Gray, J. Romberg, P. Hasler, D. Anderson, Compressive sensing on a cmos separable transform image sensor, Proc. IEEE 98 (Jun. 2010) 1089–1101.
[39] P. Schniter, Turbo reconstruction of structured sparse signals, in: Proceedings of the 44th Annual Conference on Information Sciences and Systems (CISS), 2010.
[40] M. Stojnic, F. Parvaresh, B. Hassibi, On the reconstruction of block-sparse signals with an optimal number of measurements, IEEE Trans. Signal Process. 57 (8) (2009) 3075–3085.
[41] J.A. Tropp, A.C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inf. Theory 53 (12) (2007) 4655–4666.
[42] N. Vaswani, W. Lu, Modified-CS: modifying compressive sensing for problems with partially known support, in: Proceedings of the IEEE ISIT, 2009.
[43] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[44] X. Wu, X. Zhang, J. Wang, Model-guided adaptive recovery of compressive sensing, Proceedings of the Conference on Data Compression, 2009, pp. 123–132.
[45] E.Y. Yam, J.W. Goodman, A methematical analysis of the DCT coefficient distributions for images, IEEE Trans. Image Process. 9 (10) (2000) 1661–1666.
[46] L. Yu, J.P. Barbot, G. Zheng, H. Sun, Bayesian compressive sensing for cluster structured sparse signals, Signal Process. 92 (1) (2012) 259–269.
[47] G. Yu, G. Sapiro, S. Mallat, Image modeling and enhancement via structured sparse model selection, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), Hong Kong, 2010.
[48] G. Yu, G. Sapiro, S. Mallat, Solving inverse problems with piecewise linear estimators: from Gaussian mixture models to structured sparsity, Submitted, arxiv.org/abs/1006.3056, 2010.
[49] G. Yu, G. Sapiro, Statistical compressive sensing of Gaussian mixture models, Submitted, arxiv.org/abs/1101.5785, 2011.