

FAST COMPRESSED IMAGE SENSING BASED ON SAMPLING MATRIX DESIGN

Chun-Shien Lu, Hung-Wei Chen, and Sung-Hsien Hsieh*

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

ABSTRACT

We study a fast compressive image sensing (CIS) paradigm, with computational complexity $O(m^2)$, as an alternative to compressive sensing, where m denotes the length of a measurement vector $y = \phi x$ that is sampled from the signal x of length n via the sampling matrix ϕ with dimensionality $m \times n$. In order to balance between reconstruction quality and speed, a new sampling matrix ϕ is designed. The characteristics of our method are: (i) recovery 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 . Comparisons with state-of-the-art compressive sensing methodologies are provided to demonstrate the feasibility of our method in terms of reconstruction quality and computational complexity.

Index Terms— Compressive sensing, Measurement, Recovery, Sparsity, Transform

1. INTRODUCTION

Compressive sensing (CS) has received considerable attention recently due to its revolutionary development in simultaneously sensing and compressing (sparse) signals. Inspired by the development of compressive sensing and single-pixel cameras, it is possible to sense and recover an image with as few measurements as possible if the image to be sensed is sufficiently sparse. Sparsity pattern and sensing strategy, discussed below, are the hinges on which compressed image sensing turns.

1.1. Sparsity Pattern

Researchers have explored the structure or correlation inherent in the transformed coefficients 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 popularly exploited.

In [4], instead of capturing non-adaptive or universal measurements, the authors propose attaining adaptive transform coefficients by exploiting the tree-structure of the Haar wavelet. The so-called adaptive CS framework demonstrates its superiority over the non-adaptive counterparts. In [6], a tree-structured Bayesian compressive sensing framework is proposed, wherein the hierarchical statistical models of wavelet and DCT are adopted, and Markov chain Monte Carlo (MCMC) inference is employed. The computationally inefficient MCMC mechanism is later replaced with variational analysis in [7] to speed up recovery.

In addition, the concept of clustered sparsity is also popular in compressive sensing. As summarized in [2], many existing CS algorithms exploiting clustered sparsity need to know some pre-defined

information, such as numbers, sizes, and positions of clusters, along with the degree of sparsity.

1.2. Sensing Strategy

As an illustrative example, consider the size of a sampling matrix required to sense an image of size 128×128 is as huge as 16384×16384 (assuming the use of 1D sensing), which occupies more than 12Gb for 32-bit single precision floating point. Most current desktop machines cannot afford the storage overhead of storing such a sensing matrix. In addition, large sensing matrix will incur computational overhead during the process of sampling.

There are two solutions to this problem, despite almost all CS algorithms being developed for 1D sensing. For purpose of compressive image sensing (CIS), one common strategy adopted is to divide an image into several patches/blocks with 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.

Although block-based image sensing seems to be feasible, it still incurs the sensor calibration problem. The other solution is to employ 2D separable sensing [10]. That is, separable sensing is conducted along the row and column directions such that the 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.

1.3. (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 relationship between m and the sparsity k of a signal x satisfies $m = O(k \cdot \log(\frac{n}{k}))$. 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; thereby leading to a larger measurement rate.

On the contrary, if m is still chosen to keep 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.

*Contact author: lcs@iis.sinica.edu.tw

2. PROPOSED METHOD: FAST COMPRESSED IMAGE SENSING

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

The first thing we notice is that the inference for exploiting some specific sparsity patterns is time-consuming. In order not to spend time in tracing larger transform coefficients, we propose to sample only those transform coefficients that are situated at lower frequencies. That is, we do not seek ‘‘perfect reconstruction,’’ which is difficult to achieve, due to natural images usually not being sparse.

The second concern is that CS conventionally relies on the assumption of sparsity to reconstruct the original signal from (far) fewer measurements. However, many natural signals inherently containing textured components are a kind of non-sparse signals.

In sum, the assumption of sparsity and the exploitation of structured sparsity do not conform to the property of less-sparse signals. In this paper, we study a new compressive image sensing algorithm via an elaborate design of sampling matrices. We start from the random projection, $y = \phi x$, to describe the proposed 1D sensing of an image patch/block and 2D separate sensing of a whole image, respectively.

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:

$$Ty = T(\phi x), \quad (1)$$

where x is regarded as a small image or an image patch of reasonable size. Eq. (1) is further derived based on the principle of linear operations [3] as:

$$Ty = T(\phi x) = (T^2\phi)(Tx), \quad (2)$$

where T^2 denotes 2D linear operator (see Sec. 5 for details).

Eq. (2) 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 up 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:

$$\Phi = (T^2)^{-1}(T^2\phi)^z, \quad (3)$$

where our designed sampling matrix Φ involves a random matrix ϕ and 2D linear operator T^2 .

Now, Φ is stored in the sensors for the purpose of compressive sensing. According to Eq. (2) and Eq. (3), we have the following derivations:

$$y = \Phi x \Rightarrow Ty = (T^2\Phi)(Tx) = (T^2\phi)^z(Tx). \quad (4)$$

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. (4) to obtain:

$$\begin{aligned} Ty &= \Phi^s(Tx)^s \Rightarrow \\ (\Phi^s)^{-1}Ty &= (\Phi^s)^{-1}\Phi^s(Tx)^s = (Tx)^s. \end{aligned} \quad (5)$$

It is now evident that the signal x can be approximately and quickly recovered. For the purpose of sensing, the measurement y is available at the encoder via random projection in Eq. (4). For the purpose of recovery, y is first processed at the decoder via Eq. (5), and then $(\Phi^s)^{-1}Ty$ in Eq. (5) 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 and is fixed for use at the decoder. Hence, the computational complexity of our CS recovery algorithm merely comes from processing y via $(\Phi^s)^{-1}Ty$.

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. (10) of Sec. 5, 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:

$$y_{m \times m} = \phi_{m \times n} x_{n \times n} \phi_{n \times m}^t, \quad (6)$$

where $\phi_{n \times m}^t$ denotes the transpose of $\phi_{m \times n}$. Eq. (6) also reveals the measurement rate of $\frac{m^2}{n^2}$ in 2D sensing.

Similar to Eq. (10), let $S_{m \times m}$ and $S_{n \times n}$ stand for two 1D linear operators with respective transpose represented as $S_{m \times m}^t$ and $S_{n \times n}^t$. We have $S_{n \times n} S_{n \times n}^t = I_{n \times n} = S_{n \times n}^t S_{n \times n}$, where $I_{n \times n}$ denotes an identity matrix of dimensionality $n \times n$. Then, we can further derive by imposing 2D transform on the measurement matrix $y_{m \times m}$ of Eq. (6) to attain:

$$\begin{aligned} T^2(y) &= S_{m \times m} y_{m \times m} S_{m \times m}^t \\ &= S_{m \times m} \phi_{m \times n} x_{n \times n} \phi_{n \times m}^t S_{m \times m}^t \\ &= S_{m \times m} \phi_{m \times n} (S_{n \times n}^t S_{n \times n}) x_{n \times n} \\ &\quad (S_{n \times n}^t S_{n \times n}) \phi_{n \times m}^t S_{m \times m}^t \\ &= (S_{m \times m} \phi_{m \times n} S_{n \times n}^t) (S_{n \times n} x_{n \times n} S_{n \times n}^t) \\ &\quad (S_{n \times n} \phi_{n \times m}^t S_{m \times m}^t) \\ &= T^2(\phi) T^2(x) T^2(\phi^t). \end{aligned} \quad (7)$$

Then, the sampling matrix can be derived in a way similar to Eq. (3) for 2D separate sensing. Another merit of 2D sensing is that it enables compressive sensing of large images without resorting to block-based sensing.

3. EXPERIMENTAL RESULTS

Several experiments were conducted to verify the performance of the proposed turbo fast CIS method along with different sensing strategies in terms of reconstruction quality and speed. State-of-the-art CS algorithms, including orthogonal matching pursuit (OMP), TS-BCS-MCMC [6], TS-BCS-VB [7], model-based CS (MCS) [1], and structurally random matrix (SRM) [5], were chosen for comparisons

under different measurement rates (MRs). The default settings of all source codes were employed in our experiments to better guarantee the good performances of the aforementioned methods for fair comparisons.

All experiments were conducted in Matlab 7.11 (R2010b) with 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.

3.1. Reconstruction Quality

The recovery quality is measured in terms of PSNR (in dB) and structural similarity (SSIM, $0 \leq \text{SSIM} \leq 1$) indexing, respectively, where bigger is better. It is surprising to find (Table 1 is merely an example) that our method significantly outperforms or is comparable to all of the algorithms used for comparisons 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. Please refer to [8] for illustrations of image reconstruction results.

3.2. Reconstruction Speed

The reconstruction speed is measured in terms of execution time (abbreviated as Exe.) and CPU time. We only provide in Table 2 the comparison of reconstruction speed under different measurement rates (MRs) for the Barbara image. 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 recovery. For the CS algorithms used for comparison, only the model-based CS (with CoSaMP+block sparsity) [1] is as fast as ours. It can be, however, found from Table 1 that our method obtains far better reconstruction quality than [1], in particular, for measurement rates smaller than 12.5%.

3.3. Fast CS and Recovery of Large Images

Fast compressive sensing and recovery of large-scale images is still challenging for the existing CS algorithms. The feasibility of our 2D separable sensing algorithm was demonstrated using a Shepp-Logan image of size 2048×2048 . Again, our method can obtain acceptable reconstruction results [8]. The execution time for all strategies is between 2 and 3 seconds for measurement rate ranging from 1.5625% to 50%.

4. CONCLUSIONS

Fast and accurate compressive sensing recovery is still a challenging issue. Instead of following the tradition of imposing certain sparsity patterns on a CS recovery algorithm, we propose to design a new and novel sampling matrix for the purpose of preserving important measurements. Under this circumstance, extremely fast CIS recovery with a closed-form solution of computational complexity $O(m^2)$ can be achieved. We have also studied 1D sensing and 2D separable sensing strategies, and shown that 2D separable sensing is particularly feasible in compressive sensing of large-scale images in terms of storage and computation overhead reduction and reconstruction quality improvement.

5. APPENDIX: LINEAR TRANSFORM OF RANDOM PROJECTION

The linear operator introduced in [9] is employed to derive the 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: $X=Scx$, where S is a matrix with dimensionality $m \times m$. The original signal x can be reconstructed as:

$$\hat{x} = S^t X = S^t Sx, \quad (8)$$

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:

$$\begin{aligned} X &= SxS^t, \\ \hat{x} &= S^t X S = S^t SxS^t S. \end{aligned} \quad (9)$$

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

$$\begin{aligned} S_m y &= S_m \phi x = (S_m \phi S_n^t)(S_n x) \Rightarrow \\ T[y] &= T[\phi x] = T^2[\phi]T[x]. \end{aligned} \quad (10)$$

Thus, Eq. (10) explains the rationality of Eq. (2).

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. (9), as:

$$X = HxH^t. \quad (11)$$

However, it is not straightforward to use the conventional Haar wavelet to achieve Eq. (11) because only one wavelet decomposition is allowed [8].

To handle this problem, another type of Haar transform suitable for Eq. (11) is designed with $c = \frac{\sqrt{2}}{2}$ as:

$$H = \begin{bmatrix} c^3 & c^3 & c^3 & c^3 & c^3 & c^3 & c^3 & c^3 \\ c^3 & c^3 & c^3 & c^3 & -c^3 & -c^3 & -c^3 & -c^3 \\ c^2 & c^2 & -c^2 & -c^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c^2 & c^2 & -c^2 & -c^2 \\ c & -c & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & -c & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c & -c & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c & -c \end{bmatrix}. \quad (12)$$

Different from conventional wavelet transforms, it is worth noting that the use of the Haar matrix, shown in Eq. (12), allows multi-scale wavelet decomposition finished within one matrix operation. Please refer to [8] for the illustrations of Haar wavelet decomposition using the designed Haar matrix (via Eq. (11) and Eq. (12)) and the conventional Haar filter. In fact, the Haar matrix is used in our method to decompose the sampling matrix, as indicated in Eq. (2).

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.

Table 1. Recovery Quality Comparison of CS Algorithms under Different Measurement Rates (MRs) for Barbara image.

Methods	Metrics	MR (1.56%)	MR (3.13%)	MR (6.25%)	MR (12.5%)	MR (25.0%)
OMP (Sparsify toolbox)	PSNR(dB)	15.84	16.55	18.00	20.26	23.12
	SSIM	0.19	0.22	0.31	0.47	0.64
Model-based CS (CoSaMP+block sparsity) [1]	PSNR(dB)	7.13	7.82	9.81	16.32	23.82
	SSIM	0.04	0.05	0.06	0.31	0.69
TS-BCS-VB [7]	PSNR(dB)	18.46	19.30	20.31	21.37	22.35
	SSIM	0.37	0.41	0.46	0.54	0.62
TS-BCS-MCMC (DCT tree) [6]	PSNR(dB)	8.81	11.41	18.07	22.49	24.57
	SSIM	0.06	0.20	0.43	0.59	0.73
Structurally Random Matrix (SRM) [5]	PSNR(dB)	15.52	16.68	18.50	20.51	23.38
	SSIM	0.25	0.25	0.30	0.38	0.54
Our Method (1D sensing: Haar wavelet-based)	PSNR(dB)	19.25	19.32	21.18	22.98	25.49
	SSIM	0.48	0.50	0.56	0.65	0.79
Our Method (1D sensing: DCT-based)	PSNR(dB)	19.31	19.38	20.27	22.60	25.41
	SSIM	0.48	0.49	0.52	0.61	0.77
Our Method (2D sensing: DCT-based)	PSNR(dB)	22.47	23.24	23.92	24.54	25.70
	SSIM	0.56	0.61	0.67	0.73	0.82

Table 2. Recovery Speed (in seconds) Comparison of CS Algorithms under Different MRs for Barbara image.

Methods	time	MR (1.56%)	MR (3.13%)	MR (6.25%)	MR (12.5%)	MR (25.0%)
OMP (Sparsify toolbox)	Exe.	0.32	0.42	0.70	1.36	3.56
	CPU	1.25	1.68	2.81	5.43	14.20
Model-based CS [1] (CoSaMP+block sparsity)	Exe.	1.13	1.25	1.49	2.60	5.75
	CPU	4.49	4.99	5.91	10.42	22.84
TS-BCS-VB [7]	Exe.	219.94	236.28	244.47	247.23	267.77
	CPU	220.66	907.33	937.05	949.77	1022.48
TS-BCS-MCMC [6] (DCT tree)	Exe.	2585.22	2748.26	2842.79	2848.40	2978.40
	CPU	2579.59	10305.24	10650.77	10670.41	11109.13
Our Method (1D sensing: Haar wavelet-based)	Exe.	0.02	0.02	0.02	0.02	0.03
	CPU	0.06	0.06	0.06	0.06	0.12
Our Method (1D sensing: DCT-based)	Exe.	0.02	0.02	0.02	0.02	0.02
	CPU	0.06	0.06	0.05	0.06	0.06
Our Method (2D sensing: DCT-based)	Exe.	0.06	0.06	0.06	0.06	0.07
	CPU	0.25	0.16	0.27	0.25	0.31

6. REFERENCES

- [1] R. Baraniuk, V. Cevher, M. F. Duarte, and C. Hedge, "Model-based Compressive Sensing," *IEEE Trans. on Information Theory*, vol. 56, no. 4, pp. 1982-2001, 2010.
- [2] R. Baraniuk, V. Cevher, and M. B. Wakin, "Low-Dimensional Models for Dimensionality Reduction and signal Recovery: A Geometric Perspective," *Proceedings of The IEEE*, vol. 98, no. 6, pp. 959-971, 2010.
- [3] H. W. Chen, C. S. Lu, and S. C. Pei, "Fast Compressive Sensing Recovery with Transform-based Sampling," *Proc. Workshop on Signal Processing with Adaptive Sparse Structured Representations*, pp. 110, June 27-30, Edinburgh, UK, 2011.
- [4] S. Deutsch, A. Averbuch, and S. Dekel, "Adaptive compressed image sensing based on wavelet modeling and direct sampling," *Proc. SAMPTA*, 2009.
- [5] T. T. Do, L. Gan, N. H. Nguyen, and T. D. Tran, "Fast and Efficient Compressive Sensing Using Structurally Random Matrices," *IEEE Trans. on Signal Processing*, vol. 60, no. 1, pp. 139-153, 2012.
- [6] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressed sensing," *IEEE Transactions on Signal Processing*, vol. 57, no. 9, pp. 3488-3497, 2009.
- [7] L. He, H. Chen, and L. Carin, "Tree-Structured Compressive Sensing with Variational Bayesian Analysis," *IEEE Signal Processing Letters*, vol. 17, no. 3, pp. 233-236, 2010.
- [8] C. S. Lu, H. W. Chen, and S. C. Pei, "Compressive Image Sensing: Turbo Fast Recovery with Lower-Frequency Measurement Sampling," Technical Report, <http://www.iis.sinica.edu.tw/page/library/TechReport/tr2011/tr11005.pdf>.
- [9] N. Merhav and V. Bhaskaran, "A transform domain approach to spatial domain image," *HPL-94-116*, 1994.
- [10] Y. Rivenson and A. Stern, "Compressed Imaging with a Separable Sensing Operator," *IEEE Signal Processing Letters*, vol. 16, no. 6, pp. 449-452, 2009.