Block Compressed Sensing Images using Curvelet Transform

Nasser Eslahi, Ali Aghagolzadeh, Seyed Mehdi Hosseini Andargoli
Faculty of Electrical and Computer Engineering
Babol University of Technology, Babol, Iran
Email: nasser.eslahi@stu.nit.ac.ir, aghagol@nit.ac.ir, smh_andargoli@nit.ac.ir

Abstract—Due to the optimal sparse representation of objects with edges by the multiscale and directional Curvelet Transform, its application have been increasingly interests over the past years. In this paper, we investigate how the block-based compressed sensing (BCS) can be improved to an efficient recovery algorithm, by employing the iterative Curvelet thresholding (ICT). Also, we consider two accelerated iterative shrinkage thresholding (IST) methods, including the following: 1) Beck and Teboulle’s fast iterative shrinkage thresholding algorithm (FISTA); 2) Bioucas-Dias and Figueiredo’s two-step iterative shrinkage thresholding (TwIST) algorithm, to increase the execution speed of the proposed methods rather than simple ICT. To compare our experimental results with the results of some other methods, we employ pick signal to noise ratio (PSNR) and structural similarity (SSIM) index as the quality assessor. Numerical results show good performance of the new proposed BCS using accelerated ICT methods, in terms of these two quality assessments.

Index Terms—Compressed Sensing, Sparsity, landweber iteration, Accelerated Iterative Shrinkage Thresholding, Iterative Curvelet Thresholding

I. INTRODUCTION

The recently introduced theory of compressed sensing (CS) [1]-[3] suggests a new framework for simultaneous sampling and compression of signals at a rate significantly below the Nyquist rate. It also permits that under certain conditions, the original signal can be reconstructed exactly from a small set of measurements. Applying CS for 2D signals (still images and video sequences) involves several challenges, such as a huge memory is needed to store the random sampling operator. Also the reconstruction algorithms are often time consuming and an expensive computationally process.

In [4, 5] a paradigm of block-based compressive sampling (BCS) for 2D images is addressed which the original image is divided into smaller blocks and each block is sampled and processed independently, using the same measurement operator. This method has some advantages including: (I) As we have the same size blocks, we can use a measurement operator as the same size, which can be easily stored and implemented through a random sampled filter bank; (II) Since each block is processed independently, the initial solution can be easily obtained and the reconstruction process can substantially speed up; (III) Since the encoder does not need to wait until the entire image is measured and then send the sampled data, BCS is suitable for real time applications. In [5] an extension of the BCS is presented that uses the directional transforms (e.g., dual tree discrete Wavelet transform (DDWT) [6] and Contourlet transform (CT) [7]) as the transform matrix.

In this paper, we particularly consider discrete curvelet transform (DCuT) [8, 9] as the transform matrix, because of its highly directional geometry/features representation rather than DDWT and CT. Also, we employ iterative curvelet thresholding (ICT) [10, 11] based on projected Landweber [12], that adapts hard shrinkage function to provide sparsity-enforcing thresholding while preserving the edges and features. The other motivation of applying the curvelet thresholding is that most natural images are compressible by the curvelet transform [13]. In the beginning of each iteration, we apply a spatial smoothing filter to smooth the image and reduce the element like and block artifacts, efficiently. We abbreviate the new proposed image recovery base on block compressed sensing using iterative Curvelet thresholding as BCS-ICT. Moreover, we combine two recently presented IST techniques by Bioucas-Dias and Figueirodo’s two-step IST (TwIST) [14] and fast IST algorithm (FISTA) by Beck and Teboulle [15] within the framework of the proposed BCS-ICT to accelerate the execution time. We refer to deploy of both TwIST and FISTA within the BCS-ICT framework as BCS-TwIST and BCS-FICTA, respectively. The FICTA and TwICT are not new [16], but it is the first time that they are applied for BCS.

The contribution of our paper includes: 1) We employ the ICT within the framework of methods of [4, 5] contribute with an adaptive smoothing filter, to produce a new BCS method base on discrete curvelet transform and ICT. 2) We apply the FICTA and the TwICT in the framework of the new proposed BSC-ICT method to reduce its execution time and increase the visual quality.

The rest of this paper is organized as follows. In section II, we introduce an abstract framework for compressed sensing and also we review two well-known paradigms of the BCS techniques, briefly. Section III provides a brief review of the basics of discrete curvelet transform and two accelerated IST techniques. Furthermore, the new BCS techniques based on accelerated ICT methods are presented. Numerical results and comparisons for respective methods are given in section VI and finally, section V concludes the paper.

II. BASIC THEORY OF COMPRESSED SENSING

Suppose we wish to recover a real value finite length signal $u \in \mathbb{R}^n$ from a finite length observation $f \in \mathbb{R}^m$; so that
\( m \ll n \) and there is a linear projection between them
\[
f_{m \times 1} = \Phi_{m \times n} u_{n \times 1} \tag{1}
\]
where \( \Phi \in \mathbb{R}^{m \times n} \) is a \( m \times n \) measurement (sensing) matrix (so this recovery problem is a linear inverse problem). Since the number of unknowns is much more than the observations, clearly we are not able to recover every \( u \) from \( f \) and it is generally considered as an ill-posed problem. However, if \( u \) be sufficiently sparse in the sense that \( u \) can be written as a superposition of a small number of vectors taken from a known (sparsifying) transform domain basis \((t = n)\) or frame \((t > n)\) \( \Psi \in \mathbb{R}^{t \times n} \) or even adaptive learned sparsifying basis (i.e., see [17]), then exact recovery of \( u \) is possible. So sparsity plays a key role in recovering of \( u \) from observation vector \( f \). Also, \( u \) would be called \( s \)-sparse if only its \( s \) coefficients in the set of transform domain \((\vartheta = \Psi u)\) are nonzero and other \((n - s)\) coefficients are zero. In other word, the transform domain signal \( \vartheta \) can be well approximated using only \( s < m \ll n \) nonzero entries. The simple method for recovery is finding \( \vartheta \) with the smallest \( \ell_0 \) pseudo-norm \((\|\cdot\|_{\ell_0})\) is not truly a norm, but it is a pseudo-norm; it is merely the number of nonzero coefficients) consists with the observation \( f \), like following optimization problem
\[
\arg \min_{\vartheta} \|\vartheta\|_{\ell_0} \text{ s.t. } f = \Phi \Psi^{-1} \vartheta \tag{2}
\]
and then recovering \( \hat{u} \) using \( \hat{u} = \Psi^{-1} \vartheta \). Unfortunately this problem is \( \text{NP-Hard} \) and cannot be solved for large \( n \), but there have been a large number of alternative optimization methods which proposed for finding a solution to this problem. It is shown that if the sensing matrix satisfies restricted isometry property (RIP) [2] and the signal is sparse enough, then \( f = \Phi \Psi^{-1} \vartheta \) has a unique solution and \( u \) can be reconstructed with high accuracy from the incomplete measurements \( f \). It seems logical to replace \( \ell_0 \) pseudo-norm with \( \ell_1 \) norm because it is the closest convex norm to non-convex \( \ell_0 \) pseudo-norm and minimizing the \( \ell_1 \) norm instead. We can find a solution to \( f = \Phi \Psi^{-1} \vartheta \) by solving convex \( \ell_1 \) optimization problem alternative to (2); i.e.,
\[
\arg \min_{\vartheta} \|\vartheta\|_{\ell_1} \text{ s.t. } f = \Phi \Psi^{-1} \vartheta \tag{3}
\]
This convex optimization program is often known as basis pursuit (BP) [18]. BP takes much more time to run and its computational complexity is often high; so other faster reconstruction algorithms have been proposed recently (e.g., gradient projected methods [19], greedy pursuits [20] and iterative methods [21]).

In practice CS measurements are corrupted with some additive noise. Thus (1) becomes
\[
f_{m \times 1} = \Phi_{m \times n} u_{n \times 1} + n_{m \times 1} \tag{4}
\]
where \( n \) represents the measurement error. When the observation \( f \) is contaminated by noise, the equality constraint in (3) must be relaxed to a noise-aware variant. To obtain an estimate with a reasonable accuracy and robustness to the noise, the estimation of \( u \) is formulated as an optimization problem which incorporates the prior information about the original signal. Hence (3) becomes, typically, basis pursuit denoising (BPDN) [22] in its unconstrained Lagrangian form:
\[
\min_{\vartheta} \left\{ \| f - \Phi \Psi^{-1} \vartheta \|_{\ell_2}^2 + \lambda \| \vartheta \|_{\ell_1} \right\} \tag{5}
\]
here the first term is a penalty that represents the closeness of the solution to the observed scene and quantifies the prediction error with respect to the measurements. The second term is a regularization term that represents a priori sparse information of the original scene and also it is designed to penalize an estimate that would not exhibit the expected properties. Here \( \lambda \) is a regularization parameter that balances the contribution of both terms. This minimizing problem can be solved easily by iterative shrinkage/thresholding (IST) methods (see, e.g., [10]), Bregman iterative algorithms (see, e.g., [17] and [23]), projected gradient methods and etc.

The iterative shrinkage methods are quite universal, robust, and simple to implement. Another advantage is that the rich existing sparse transforms can be incorporated easily into the IST framework. Therefore, it became one of the most popular tools for solving linear inverse problems. However, a drawback of the IST methods is its slow convergence. Many accelerated techniques for IST methods have been proposed. The most popular of them are two-step IST (TwIST) [14] and fast IST algorithm (FISTA) [15].

A. Basic BCS

A prototype of BCS technique is proposed in [4] that first an image, of size \( I_r \times I_c \), is divided into small non-overlapping blocks (i.e., size \( B \times B \)) and then the same sensing matrix \( \Phi_B \) (i.e., size \( m_B \times B^2 \)), where \( m_B = \lceil \frac{m \times B^2}{n} \rceil, n = I_r I_c \) is applied for sampling of each block. In this case, we have:
\[
f_i = \Phi_B u_i \tag{6}
\]
where \( u_i \) is a vector representing of block \( i \) of the input image, \( f_i \) is its corresponding measurement vector and \( \Phi_B \) is an orthonormalized i.i.d Gaussian matrix. Using this technique has several benefits comparing to use of a random sampling operator to the entire image; i.e., the encoder does not need to wait until the entire image is measured, but may send each block after its linear projection. At the decoder side, each block is processed independently; therefore we have increasing in speed of encoding and decoding. Also in this case, we just need to store a \( m_B \times B^2 \) measurement matrix instead of a \( m \times n \) measurement matrix. Usually, when the subsampling rate \( m_B / B^2 \) (or measurement rate; \( m/n \)) is low, this method creates blocking artifacts which reduce the visual quality. To reduce this blocking artifact and smooth the image, a \( 3 \times 3 \) Wiener filter is applied in the spatial domain in the beginning of each iteration. The method addressed in [4] introduces a paradigm of CS but it does not lead to an approach for high quality image reconstruction.

B. BCS Using Directional Transforms

A combination of two processing steps for recovery algorithm is addressed in [5] to improve the visual quality of
reconstructed image. This BCS recovery method is based on
consecutive projection onto convex sets (POCS) [24] and
iterative hard thresholding (IHT) [21]. Also, it uses a 2D Wiener
filter into the PL iteration that incorporates a smoothing opera-
tion intended to simultaneously achieving sparsity, smoothness
and reducing the blocking artifacts. Furthermore, the iterative
projection-based BCS recovery adapts bivariate shrinkage [25]
to directional decomposition structure of images to provide
sparsity-enforcing thresholding. The method in [5] is called
BCS-SPL. The BCS-SPL gains a better reconstructed image
compared with the BCS in [4], but both of them do not
preserve edges well.

III. BCS-ACCELERATED ICT

In this section, first, we briefly introduce the basics of dis-
crete curvelet transform (DCuT) and iterative curvelet thresh-
olding (ICT), and then we propose two new BCS methods
using accelerated ICT.

A. Basics of DCuT and ICT

Traditional wavelets perform well only at the represent-
ing point singularities, since they ignore geometric properties
of structures and do not exploit the regularity of edges. Thus,
wavelet-based compression, denoising or structure extraction
become computationally inefficient for geometric features with
line and surface singularities.

The curvelet transform represents edges and singularities
along curves more precisely with the anisotropic needle-
like features. Unlike wavelets, curvelets are indexed by three
parameters: 1) a scale $2^{-j}, j \in \mathbb{N}_0;$ 2) an equispaced sequence
of rotation angles $\theta_{j,l} = 2\pi l 2^{-\frac{j}{2}}, l = 0, 1, \ldots, 2^{\frac{j}{2}} - 1$
such that $0 \leq \theta_{j,l} \leq 2\pi;$ and 3) a position $x_k^{(j,l)} =
R_{\theta_{j,l}}(k_1 2^{-j}, k_2 2^{-\frac{j}{2}} T); k = (k_1, k_2) \in \mathbb{Z}^2,$
where $R_{\theta_{j,l}}$ denotes the rotation matrix with angle $\theta_{j,l}$
and $k$ is the sequence of translation parameters. The curvelet elements are defined by

$$
\Psi_{j,l,k}(x) := \Psi_j(R_{\theta_{j,l}}(x - x_k^{(j,l)}), x = (x_1, x_2) \in \mathbb{R}^2 \quad (7)
$$

where $\Psi_j$ are smooth functions with compact support on
wedges in Fourier domain. The family of curvelet functions
forms a tight frame, and thus, each function $f$ has a representa-
tion as:

$$
f = \sum_{j,l,k} \langle f, \Psi_{j,l,k} \rangle \Psi_{j,l,k} \quad (8)
$$

where $\langle f, \Psi_{j,l,k} \rangle$ denotes the scalar product of $f$ and $\Psi_{j,l,k}.$
The coefficients $\psi_{j,l,k} = \langle f, \Psi_{j,l,k} \rangle$ are called curvelet coeffi-
cients of function $f.$

The basic ICT is similar to IST (see, e.g., [11]); actually ICT
is an extension of IST when the transform matrix is DCuT.
The basic ICT can be written as:

$$
u^{(k+1)} = S_{r,T}(u^{(k)} + \alpha^{(k)} \Phi^T (f - \Phi u^{(k)})) \quad (9)
$$

where $\alpha^{(k)}$ is a step size of line search, and $S_{r,T}(\cdot)$ is the
shrinkage operator given by

$$
S_{r,T}(f) = \sum_{j,l,k} T_r((f, \Psi_{j,l,k}) \Psi_{j,l,k} \quad (10)
$$

where $T_r$ can be taken as a soft or hard-shrinkage function.
Soft-shrinkage function can be defined by a fixed threshold
$\tau > 0$ as:

$$
T_{s,\tau}(x) = \begin{cases}
-x & x \geq 0 \\
0 & |x| < \tau \\
-x & x \leq 0
\end{cases} \quad (11)
$$

and hard-shrinkage function is considered as:

$$
T_{h,\tau}(x) = \begin{cases}
x & |x| \geq \tau \\
0 & o.w.
\end{cases} \quad (12)
$$

The ICT can be also considered in coefficient domain as:

$$
g^{(k+1)} = T_{s,r}(g^{(k)} + \alpha^{(k)} \Phi^T (f - \Phi \Psi^{-1} g^{(k)})) \quad (13)
$$

where $g^{(k)} = \Psi u^{(k)}, u^{(k)} = \Psi^{-1} g^{(k)}$ and $T = \{h, s\}.$

B. Proposed BCS Method

IST (and also ICT) is slow convergence algorithm [16],
but many accelerated techniques for IST methods have been
proposed. The most popular of them are TwIST [14] and
FISTA [15].

The TwIST has a general formulation

$$
u^{(1)} = \beta S_{r,T}(u^{(0)} + \gamma \Phi^T (f - \Phi u^{(0)}))
$$

$$
u^{(k+1)} = (1 - \alpha) u^{(k-1)} + (\alpha - \beta) u^{(k)} + \beta S_{r,T}(u^{(k)} + \gamma \Phi^T (f - \Phi u^{(k)})), k \geq 1 \quad (14)
$$

where $\alpha$ and $\beta$ are the relax parameters and can be taken as:

$$
\alpha = \rho^2 + 1, \quad \beta = 2\alpha/(1 + \epsilon) \quad (15)
$$

where $\epsilon$ is a small value, e.g., $10^{-1}$ or $10^{-3},$ and

$$
\rho = (1 - \sqrt{\kappa})/(1 + \sqrt{\kappa}) < 1, \quad \kappa = \zeta_1/\max(1, \zeta_m). \quad (16)
$$

Also, $\zeta_1$ and $\zeta_m$ are two real numbers such that

$$
0 < \zeta_1 \leq \lambda_i(\Phi^T \Phi) \leq \zeta_m, \quad (17)
$$

where $\lambda_i(\cdot)$ is the $i$th eigenvalue of its argument.

FISTA is another accelerated IST algorithm by general form:

$$
\tilde{u}^{(k+1)} = S_{r,T}(\tilde{u}^{(k)} + \gamma \Phi^T (f - \Phi \tilde{u}^{(k)}))
$$

$$
u^{(k+1)} = u^{(k)} + (k - 1)/t_k u^{(k)} - u^{(k-1)}, \quad (18)
$$

here, $t_k^{(k+1)} = (1/2)(1 + \sqrt{1 + 4(t_k^{(k)})^2})$ starting with $t_0 = 1$
and $\gamma$ is a scaling factor. These accelerated algorithms have
been extended to use ICT for BCS image recovery by applying
DCuT as the transform matrix $\Psi.$ We rename these two
methods as TwICT and FICTA, respectively.

By adapting TwICT and FICTA in the framework of meth-
ods [4], we accomplish a BCS method based on accelerated
ICT. We refer to the overall of these two techniques as BCS-TwICT and BCS-FICTA, respectively. In the first step of processing, an adaptive smoothing filter is applied in the spatial domain to reduce the blocking and element like artifacts. For low measurement rates (below 30%) we employ a $3 \times 3$ median filter, otherwise a Wiener filter the same size is used. Moreover, we employ the hard shrinkage function (12) as the threshold criterion in hard thresholding $S_{\tau, \sigma}^\circ(\cdot)$ step. We remove some insignificant curvelet coefficients by using hard shrinkage function in curvelet domain. Table I shows the framework of BCS-FICTA as a typical case of BCS-ICT method. Also, BCS-TwIST framework is similar to the last one.

### IV. Numerical Experiments

In this section, we evaluate the performance of the proposed BCS-FICTA and BCS-TwIST as two new BCS techniques. To evaluate our simulation results, we use two applicable quality assessors, pick signal to noise ratio (PSNR) in dB and structural similarity (SSIM). For an 8 bits gray scale $n \times n$ image, PSNR is calculated as

$$\text{PSNR} = 20 \log_{10} \frac{n255}{|u - \tilde{u}|},$$

(19)

where $u$ and $\tilde{u}$ denote the original image and the reconstructed image, respectively.

SSIM [26] is another assessor for measuring the similarity between two images, and it is near to human eye perception. In this paper, the following definition for SSIM index is used:

$$\text{SSIM}(u, \tilde{u}) = \frac{(2\mu_u\mu_{\tilde{u}} + c_1)(2\sigma_{u\tilde{u}} + c_2)}{(\mu_u^2 + \mu_{\tilde{u}}^2 + c_1)(\sigma_u^2 + \sigma_{\tilde{u}}^2 + c_2)}$$

(20)

where $\mu_u$ and $\sigma_u$ denote the average and variance of $u$, respectively (and similarly for $\tilde{u}$), $\sigma_{u\tilde{u}}$ is the cross covariance of $u$ and $\tilde{u}$, $c_1$ and $c_2$ are two variables to stabilize the division with weak denominator so that $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ and $k_1, k_2 \ll 1$. $L$ is the dynamic range of pixel values (255 for 8bits gray scale images). For our experiments, we use $k_1 = 0.01$ and $k_2 = 0.03$. In this paper, we use the mean of the SSIM (MSSIM) of all trails. All experiments were performed using MATLAB 2012a, on a laptop computer equipped with Intel® core™ M i5, 2.4 GHz processor, with 6 GB of RAM, and running on Windows 7. The performances of our experiments are evaluated on two well-known standard test images of size $512 \times 512$, “Barbara” and “Lenna”. In all experiments, we use block size of $32 \times 32$ which is used in [4, 5], to compare with them. Also, to evaluate the effectiveness of the proposed BCS-ICT (BCS-FICTA and BCS-TwICT), the obtained results are compared with those of some well-known CS recovery algorithms such as BCS-SPL [5], sparsity adaptive matching pursuits (SAMP) [20], split augmented Lagrangian shrinkage algorithm (SALSA) [27] and gradient projection for sparse reconstruction (GPSR) [19]. We used the implementation codes obtained from authors for BCS-SPL, GPSR, SALSA and SAMP. In BCS-SPL case, the common parameters are $k_{\text{max}} = 200$ (maximum number of iterations) and $TOL = 10^{-4}$.

To evaluate our proposed methods, we use the second generation of DCuT [9] as the transform matrix $\Psi$ with linear decay threshold value $\sigma(k) = \sigma_0(1 - k/k_{\text{max}})$ in each iteration within the hard shrinkage function applied in the DCuT domain to enforce sparsity. In this case, $k_{\text{max}} = 50$, $TOL = 10^{-4}$, the initial threshold value $\sigma_0 = 0.2$ and the size of smoothing filter is $3 \times 3$ (see table 1). Also, we used $\alpha = 1$ and $\beta = 1.8182$ for BCS-TwIST. We use functions “vision.MedianFilter” and “wiener2” in MATLAB for median filter and Wiener filter as the smoothing filter, respectively. All values are obtained by averaging on 4 independent trails.

Fig. 1 shows the results of experiment 1 with 20% of measurements, which is performed on “Barbara”. Fig. 1(a) is the result of direct block-based recovery by $u_j = \Phi_B^T f_j$. Fig. 1(b)-(d) are obtained by GPSR [19], SALSA [27] and SAMP[20], respectively. It is obvious that these methods produce low visual quality reconstructed images and do not perform well in preserving edges. Fig. 1(e)-(g) show the results of the BCS-SPL method using DCT, DWT (popular biorthogonal 9-7 DWT) and DDWT as the transform matrices, respectively. It can be seen that BCS-SPL methods present a better image recovery compared with the methods used in Fig.1 (b)-(d) but, they do not perform well in preserving curves and link-like objects, too (i.e., see the stripped lines of pants and scarf). Fig. 1(h)-(i) are obtained by BCS-FICTA and BCS-TwICT with the same decay threshold value. It is obvious that they perform well in preserving edges and produce images with higher quality compared to the mentioned methods, but with the higher computational complexity than the methods of Fig. 1(b)-(g). The most of computational complexity is used by the Curvelet transform in each iteration.

### TABLE I: BCS-FICTA ALGORITHM

<table>
<thead>
<tr>
<th>Input: $f, \Phi_B, \Psi, \sigma_0, B$; Block size, $k_{\text{max}}$: maximum iteration number, $\tau$: TOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output: $u$: Reconstructed image;</td>
</tr>
<tr>
<td>Initialization: Set $k = 0$; $u(0^-) = u(0)$; $t(0^+) = 1$; $u_j^0 = \Phi_B^T f_j$</td>
</tr>
<tr>
<td>repeat</td>
</tr>
<tr>
<td>$u(k) = \text{smoothing filter}(u(k))$;</td>
</tr>
<tr>
<td>for each block $j$ do</td>
</tr>
<tr>
<td>$\hat{u}(k) \leftarrow u_j(k) + (\frac{1}{1 + t^+}) (u_j(k) - u(k))$</td>
</tr>
<tr>
<td>$\hat{u}(k) \leftarrow \hat{u}(j) + \Phi_B^T (f_j - \Phi_B \hat{u}(j))$</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>$\hat{u}(k) = \Psi \hat{u}(k)$</td>
</tr>
<tr>
<td>$\sigma(k) \leftarrow \sigma_0 (1 - k/k_{\text{max}})$</td>
</tr>
<tr>
<td>$\hat{u}(k) = T_{h, \sigma}(\hat{u}(k))$</td>
</tr>
<tr>
<td>$\hat{u}(k) = \Psi^{-1} \hat{u}(k)$</td>
</tr>
<tr>
<td>for each block $j$ do</td>
</tr>
<tr>
<td>$\hat{u}_j(k + 1) \leftarrow \hat{u}_j(k) + \Phi_B^T (f_j - \Phi_B \hat{u}_j(k))$</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>Compute $D(k + 1) = \frac{1}{\sqrt{2}} | \hat{u}(k + 1) - \hat{u}(k) |_2$</td>
</tr>
<tr>
<td>Update $u(k) \leftarrow u(k + 1)$</td>
</tr>
<tr>
<td>$k = k + 1$</td>
</tr>
<tr>
<td>until $</td>
</tr>
</tbody>
</table>

Note: $u_j^{(k)}$ denotes $j$th block of $u^{(k)}$, by size $B \times B$. |
If we do not use accelerated ICT, the recovery time would increase significantly, due to slow convergence of IST and ICT methods. The BCS-FICTA is slightly faster than BCS-TwICT, but it loses an insignificant value in PSNR and SSIM. Anyway, both of them are faster than BCS using simple ICT. Fig. 1(h) shows a part of the original Barbara image. Fig. 2 has the same scheme as Fig. 1, but performed on “Lenna” image. Fig. 3(a) and Fig. 3(b) show the PSNR performance of the proposed methods in various measurement ratios, compared with the other mentioned recovery algorithms for “Barbara” and “Lenna”, respectively. The numerical results show the adequacy of our proposed methods in BCS.

V. CONCLUSION

The motivation of this paper is to investigate how we can improve the block-based compressed sensing recovery algorithm. We adopted the general paradigm of BCS using a projection-based reconstruction iterative method, contributes with an adaptive smoothing filter and using DCuT as the transform matrix. Also, we adapted two accelerated ICT methods to reduce the computational complexity (compared to the simple ICT) and achieve higher visual quality. Our methods have advantages and drawbacks. For instance the BCS-FICTA and BCS-TwICT are much slower than the BCS-SPL, but they achieve higher PSNR and SSIM values and also better visual quality, particularly for curve-like and textural images.

REFERENCES

Fig. 2: Recovery of Lenna with 20% of measurements by methods: (a) Direct BCS-based Recovery by $u_j = \Phi_T^B f_j$, PSNR=6.79 dB, SSIM=0.022. (b) GPSR, PSNR=28.65 dB, MSSIM=0.787, $t_g$=13 sec. (c) SALSA, PSNR=28.5, MSSIM=0.782, $t_g$=12.7 sec. (d) SAMP, PSNR=28.48 dB, MSSIM=0.782, $t_d$=12.7 sec. (e) BCS-SPL-DCT, PSNR=30.41 dB, MSSIM=0.844, $t_e$=18.26 sec. (f) BCS-SPL-DWT, PSNR=30.7 dB, MSSIM=0.854, $t_f$=16.9 sec. (g) BCS-FICTA, PSNR=32.68, MSSIM=0.89, $t_{BCS-FICTA}$=114 sec. (h) BCS-TwICT (and BCS-ICT, $t_{BCS-ICT}$=140.5 sec), PSNR=32.73 dB, MSSIM=0.891, $t_{BCS-TwICT}$=129 sec. (i) The original image (a part of Lenna image).


Fig. 3: PSNR Performance of the proposed method compared with some well-known recovery methods in various measurement rates. (a) Barbara. (b) Lenna.