Adaptive Compressive Imaging using Non-Random Dictionaries

Joseph Zammit, Student Member, IEEE, and Ian J. Wassell, Member, IEEE

Abstract—Practical, real-time compressive sensing (CS) of images requires that an image is first partitioned into blocks to reduce the sensing matrix size. However, the blocks are found to have differing sparsity levels which makes reconstruction problematic. Adaptive CS attempts to detect blocks with low sparsity and assign more measurements to these blocks to increase reconstruction quality. This paper proposes two algorithms to estimate sparsity based on Block Boundary Variation (BBV) and 2D Discrete Cosine Transform (2D-DCT) Domain (DD) measurements. The former uses the variation in pixel values around the boundary of image blocks to modify the measurement rate whereas the latter uses 2D DCT measurements. The estimation measurements are sampled at the encoder side, from the overall budget of CS measurements available to the algorithm. The two schemes are combined with a non-random CS algorithm that uses linear transform measurements directly from the sparsifying domain, for example the Linear 2D DCT (L-DCT) to create two Adaptive L-DCT (AL-DCT) algorithms, AL-DCT-BBV and AL-DCT-DD, that can be reconstructed with very low computational complexity at the decoder, in real-time, with very good quality. The quality can be increased further by reconstructing the images using conventional CS techniques, for example the Denoising Approximate Message Passing (D-AMP) algorithm as a post-processing operation, but incurring significant time penalties. D-AMP was modified to produce two algorithms, DAMP-D and IDA, that produce better results on AL-DCT-BBV and AL-DCT-DD. The adaptive, block-based algorithms achieve state-of-the-art performance and can be reconstructed from images sensed with Single-Pixel Cameras, Multi-Pixel Cameras and Focal Plane Processing Image Sensors.

I. INTRODUCTION

Compressive sensing (CS) has emerged as an alternative analogue signal processing technique which capitalises on the sparsity of the sensed signals, in some domain, to perform joint sampling and source coding, requiring only $O(S)$ compressive sensing measurements for an $S$-sparse signal. When extended to images and video, CS has several other advantages including lens-less cameras, multi-wavelength imaging, and reduced capture time.

Real-time Video Compressive Sensing has emerged as an interesting field of study for energy efficient capture and transmission of visual sensory data, for example in the field of Wireless Sensor Networks (WSNs). The classical digital signal processing paradigm of sampling at the Nyquist frequency, source coding, channel coding, followed by transmission, is often found not to be energy efficient enough to allow wireless sensor nodes to operate autonomously for as long as is required.

The theory of compressive sensing developed by Candes, Romberg and Tao [1], and Donoho [2] independently, is based on the Johnson-Lindenstrauss Lemma [3] which states that a small number of points in a high-dimensional space can be mapped, with low distortion, into a significantly-lower dimensional space. The theory gives bounds on the number of compressive samples required to reconstruct sparse signals together with statistical guarantees.

The application of interest, real-time image and video sensing, requires CS analogue sensors that can perform the required sensing energy-efficiently, and in a short period of time. In certain applications, for example magnetic resonance imaging (MRI), the current sensing scanners inherently collect measurements in the form required by CS. In other applications, researchers have had to design compressive sensors matched to the application, for example cochlea-inspired harmonic analysers for vibrational signals [4], or the single-pixel camera (SPC) for image and video signals [5].

Whereas the compressive sensing itself is straightforward, represented by the multiplication of the sensed signal $f \in \mathbb{R}^N$ with a measurement matrix $\Phi \in \mathbb{R}^{M \times N}$, to collect $M$ sparse samples $y \in \mathbb{R}^M$, the reconstruction is challenging. The solution of the underdetermined linear system $y = \Phi \Psi x = Ax$ where $\Psi$ is some transform such that $f = \Psi x$, $x$ is sparse, and $A$ is the effective sensing matrix, requires the computation of the $L_0$ minimization of $x$, which is known to be NP hard. $L_1$ and Total Variation (TV) minimization have been found to be alternative feasible solutions, since the problems can be cast as Linear Programming (LP) problems, and solved using state-of-the-art LP solvers.

$L_1$ minimization itself is however time consuming when the order $N$ of the sensed signal is large. Therefore, a significant number of reconstruction techniques have been proposed such as Matching Pursuit [6], Bayesian [7], Approximate Message Passing [8], and recently Denoising [9] and Neural Networks [10], which deliver accurate reconstructions in a fraction of the time required by $L_1$ and TV minimization.

When compressive sensing is applied to images, the vectorised signal has dimension $N$ given by $N = H \times W$ where $H$ and $W$ are the vertical and horizontal pixel resolutions. The measurement matrix at the sensor side is then of size $M \times N$ or $r(H \times W)^2$ where $r$ is the sensing sub-rate $M/N$. The matrix thus becomes too large to handle effectively and energy efficiently at the energy-constrained sensor. Furthermore, the reconstruction exercise is very time-consuming at the decoder side.

Block-based compressive image sensing has thus been proposed [11], [12], [13], inspired by the success of block-based...
image and video compression schemes such as JPEG [14], JPEG2000 [15], H.264 [16] and H.265 [17]. The image or frame is broken down into, often non-overlapping, \( B \times B \) pixel blocks, which are compressively sensed and reconstructed independently, and then reformed into the decoded image or frame.

The measurement and sensing matrices are reduced greatly in size and become manageable both in terms of storage requirements and computational effort. Unfortunately, Block-based Compressive Sensing (BCS) is afflicted by blockiness, and usually delivers Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) values below those of full-image CS. A significant body of literature has focused on BCS, extending it to incorporate Differential Pulse Code Modulation (DPCM) and motion compensation techniques found in current image and video compression schemes, tackling image blockiness and increasing PSNR and SSIM.

BCS is essential for energy-efficient, real-time image and video sensing applications. If we contemplate a Multi-Pixel Camera (MPC) instead of the SPC [5] [19], then the \( m \) compressive samples in each block, can be collected in parallel, and the sensing time of the image or frame is reduced by the number of \( B \times B \) pixel blocks \( n_B = H \times W / B^2 \) the image or frame has been broken down into.

Compressive sensing can be carried out using non-random, deterministic measurements [20]. It has been noted that the SPC or the MPC, that can be implemented using Digital Micro-Mirror Devices (DMDs) for example [5], or using Focal Plane Processing Image Sensors (FPPIS) [21], can collect both random and non-random, deterministic measurements. Thus, if the measurement matrix consists of transform coefficients instead of random coefficients, then it can compute transform coefficients directly, with the same energy efficiency as for the random projections. Reconstruction at the decoder would be trivially implemented as the inverse transform of the forward transform implemented on the DMD sensor. Deterministic measurement matrices allow storage space reduction through compact representation. It has been shown that objective and subjective quality of the image can be improved by using reconstruction techniques developed for random CS. The main disadvantage is that the reconstruction quality achieved when using non-random CS reconstruction techniques is lower than that achieved with random RIP-compliant (restricted isometry property) sensing matrices.

When an image is divided into blocks, the activity in each block varies and hence the sparsity changes as well. If the measurement rate in each block is the same, some blocks will be over sampled and others under sampled. In the latter case, the reconstruction fails with very noticeable errored blocks. A number of authors have therefore proposed and studied adaptive techniques which aim to match the measurement rate in BCS to the underlying sparsity in the block. This paper extends this work and combines it with non-random sensing regimes.

### A. Contributions

The main contribution in this paper is the development of two adaptive algorithms to estimate sparsity - Block Boundary Variation (BBV) and DCT Domain (DD) - that can be applied to compressive image sensing. In this paper, the techniques are combined with a 2D-linear-transform-based CS technique which senses the image with measurements taken directly from the sparsifying domain, acquired in zigzag order, even though the sensing matrix is not RIP-compliant. The advantage of this CS technique is that reconstruction is accomplished in real-time on CPU-equipped decoders, using low complexity inverse 2D linear transforms. Extensive simulation has shown that the two algorithms, AL-DCT-BBV and AL-DCT-DD, achieve state-of-the-art performance amongst CPU-reconstructed, real-time algorithms that can be reconstructed on SPCs, MPCs and FPPIS.

The paper also demonstrates that classical CS reconstruction techniques, such as Denoising Approximate Message Passing (D-AMP) [9], can be used to improve the quality of reconstruction albeit with a large decoding time penalty, in post-processing applications. D-AMP was modified in two ways to improve the reconstruction of the non-random, adaptive algorithms. First the Onsager term in D-AMP was damped to produce the D-AMP Damped (DAMP-D) algorithm and secondly it was simplified to produce the Iterative Denoising Algorithm (IDA).

A minor contribution is the development of two deblocking techniques that can deblock L-DCT-ZZ, AL-DCT-BBV and AL-DCT-BBV reconstructions, namely the Zigzag scan Filter (ZZF) and Threshold Filter (THF), that have a lower complexity than BM3D [22].

### B. Structure of the paper

Section [I] discusses related work in the field of adaptive block compressive sensing (ABCS). Section [III] introduces CS and reconstruction theory. This is extended to better reconstruct non-random CS images in section [VII]. Non-adaptive deterministic measurement algorithms are introduced and benchmarked in section [V] prior to a description of the two adaptive deterministic algorithms. The BBV and DD algorithms are introduced in section [V], analysed in section [VI] and investigated empirically in section [VIII] first comparing them with non-adaptive CS algorithms, then comparing them with other adaptive CS algorithms proposed in the literature. Section [IX] draws conclusions from the study.

### C. Notation used in this paper

The following notation is used to represent the algorithms and variants proposed in this paper. Only one of the options separated by | in curly brackets must be included in the name. Any, or none, of the options in square brackets can be included. \{BBV|DD\} make the algorithms adaptive and require the initial [A] option. Thus \{A|L-\{DCT\|DWT\}-\{ZZ|THB|THI\}|{BBV|DD}\}-\{BM3D|ZZF|THF\}|{D-AMP|DAMP-D}|{IDA}\} succinctly represents all possible algorithm variants discussed in this paper.
Matrices are designated by bold capital letters, with vectors in bold lower case. Scalars are represented by normal letters, both upper and lower case, sometimes accompanied by subscripts. The superscript $i$ designates the value of vectors or scalars in the $i$th iteration. Subscripts also designate the scalar components of vectors and matrices.

II. RELATED WORK

Several authors include the measurements required to classify blocks at the encoder, prior to allocating the subrate to each block, as part of the overall compressive measurements $M$. An alternative technique is to first transmit a reduced resolution image using normal BCS techniques, reconstructing it at the decoder, performing the analysis there, and feeding back the subrate information to the encoder.

Another group of papers assume that a full set of image pixels are already available to the encoder prior to compressive sensing and use these to perform the classification task. We include these full sensing prior techniques here, because they help shed light on any additional improvements possible.

A. Encoder side Adaptive CS techniques

Averbuch et al. proposed a departure from using random dictionaries that are incoherent with the sparse basis as used in classical CS [23]. Their dictionaries, instead, are non-random transform coefficient measurements from the sparsifying basis itself for example wavelets. They showed that these non-random measurements can still be captured by a DMD. Furthermore, they noted that starting from the base subband in a sparse wavelet representation of the image, coefficients in the coarser subbands are a very good indication of the location of non-zero coefficients in the finer subbands. They then propose an adaptive technique which first collects the wavelet coefficients in the lowest subband, and then to increasingly measure other wavelets in the finer sub-bands, based on statistical modelling in the wavelet domain, until the measurement allocation is exhausted. Statistical Adaptive CS (Stat-ACS) achieves 2.83 dB, 1.47 dB and 1.75 dB improvement over classical, non-adaptive CS algorithms, for $256 \times 256$ Lena, Cameraman and Peppers images respectively. To improve quality, the authors then suggest a cartoon/pixel model of an image with the texture measured by computing the 2D Discrete Cosine Transform (2D-DCT) in $16 \times 16$-pixel blocks, that are then used to adapt wavelet measurements based on activity in the block. The performance of Texture-ACS was shown to increase by 0.56 dB over Stat-ACS for the $512 \times 512$ Barbara image. The only overhead is in the classification DCT measurements collected from $16 \times 16$ blocks.

Nguyen et al. varied the rate in each BCS block based on the number of edge pixels in each $16 \times 16$ pixel block [24]. To eliminate overhead, the mean of each block is obtained as the mean of normal block measurements since the mean is preserved by the linear sensing measurement operations. The low-resolution image formed by the means is then upscaled using bicubic interpolation and used to estimate the number of edge pixels in each high-resolution Nguyen pixel block. Experimental results on $512 \times 512$, Lena, Barbara, Boat and Cameraman test images, at 0.2, 0.4 and 0.6 subrates, and using three standard CS reconstruction techniques, show that adaptivity allows PSNR and SSIM performance to increase significantly over non-adaptive reconstruction, from between 0.64 dB to 3.29 dB respectively.

Liu and Ling proposed Intra-Scale Variable Density Sampling (InVDS) to capture more measurements where it is more likely to find sparsifying basis coefficients [25]. They observed that in wavelet decomposition, the large coefficients at finer scales cluster around the large coefficients at coarser scales. Thus, starting from all the coefficients at the coarsest scales, Latin Hypercube Sampling (LHS) is used to sample the finer scales where it is most likely to find the largest coefficients. The largest sampled coefficients are added to the set of recovered coefficients until the allocated measurement budget is exhausted. The image can be retrieved directly from the measurements, or a CS reconstruction technique can be used to enhance the quality. D-AMP [9] was proposed for this purpose. Simulation on $512 \times 512$ Barbara, Lena, Cameraman and Pirate (aka Man) images was used to show the effectiveness of the technique compared with wavelet schemes that adopt classical search techniques instead of LHS. Excellent results were also reported for DCT and Hadamard schemes even though the transforms decorrelate the coefficients besides sparsifying them. D-AMP reconstruction was shown to achieve remarkable improvements on the directly reconstructed images, reaching, for example, 11.23 dB in the case of Cameraman using DCT measurements at a subrate of 0.30. It is intimated that all the measurements are accounted for in the search and measurement scheme, i.e., there is no overhead, and it is not assumed that the M largest coefficients are known a priori.

B. Decoder Side Adaptive CS techniques

Zhang et al. proposed a four-stage, adaptive BCS (ABCS) scheme that first transmits a block compressive sensed image measured at a low fixed rate [26]. At the decoder, the image is reconstructed, and a block classification is performed based on the values of transform coefficients from non-overlapping $8 \times 8$ DCT operations. The DCT transform coefficients are grouped into DC, Low-Frequency, Edge and High-Frequency coefficients and depending on the total values of these four groups, the block is classified as plain, texture, or edge. This information is fed back to the encoder which dedicates more measurements to plain blocks, so that these can be better resolved into texture or edge blocks at the decoder during the second stage. The decoder repeats the classification and when this information is again fed back to the encoder, all measurements are now allocated to texture and edge blocks, in the third stage. In the fourth and final stage, after decoder-side classification information is fed back to the encoder, more measurements are allocated to texture than edge blocks. During the four stages the average number of random CS samples ($R_{BCS}$) collected from each $8 \times 8$ block increases from 4 to 10, to 16 and finally to 22. However, the number of measurements is unevenly assigned, with more measurements in Texture blocks than in Edge blocks and Plain blocks.
respectively. The increases in PSNR for 256 × 256 Lena and Cameraman and 512 × 512 Barbara, over non-adaptive BCS, is 1.41 dB, 3.89 dB and 2.83 dB respectively. This scheme avoids classification measurement overheads since classification is performed at the decoder and uses feedback to adapt the block measurements.

Zhu et al. first proposed a new non-adaptive jointly reweighted sampling scheme JRW-BCS [27]. They then propose allocating bit rate to blocks depending on (i) entropy of the DCT coefficients in a block (ii) number of variations of DCT coefficients about a threshold in a block (iii) number of DCT coefficients in a block exceeding a threshold. They then propose two solutions to implement adaptive sampling. The first does the statistical analysis at the encoder side and must transmit information about the number of measurements per block to the decoder. The second, uses two phases. The first is a low-rate non-adaptive CS, which allows a low-quality per block to the decoder. The second, uses two phases. The first does the statistical analysis at the encoder side and

Zheng and Zhu vary the sampling rate in each block of texture blocks sampled the most frequently and smooth blocks the least. When the base layer was coded at r = 0.4 the PSNR for 256 × 256 images Lena, Boat and Cameraman and Barbara increased by 0.12 dB, 0.21 dB, 1.50 dB and 0.56 dB respectively over that for normal BCS using the same effective measurement rate as for the combined base and enhancement layer measurements in ABCS. The classification measurements assume that all the pixel values are available, and that classification is carried out at the encoder side.

Guicquero et al. adapt the number of measurements in block CS, based on low-cost block variance measurements [32]. The image is first partitioned into non-overlapping 8 × 8 sub-blocks and the variance measured in these sub-blocks. The variances from 4 × 4 sub-blocks are then averaged to obtain the variance of 32 × 32 pixel blocks and used to adapt the number of measurements on each of the 32 × 32 pixel block with the measurement ratio calculated from a sigmoid rate-variance function. The 8 × 8 pixel block variances calculated are also used as additional measurements to improve reconstruction quality. The increase in PSNR for 256 × 256 Cameraman, Darkhair, Blonde, Pirate and Lena images was found to be 2.9 dB, 0.5 dB, 0.6 dB, 0.2 dB and 0.8 dB respectively, over the non-adaptive scheme. The adaptivity determining calculations require the availability of pixel values at the encoder as in

Akbari et al. proposed first using non-adaptive BCS to capture and transmit a low resolution image [30]. Following reconstruction at the decoder, a saliency map is computed using the Graph Based Visual Saliency model (reference [11] in [30]), from which block rates are computed for each image block, and fed back to the encoder for a second adaptive sensing round. Experimental results on 512 × 512 Lena, Plane, Mandrill, Boat and Living Room images indicate that the proposed scheme achieves PSNR improvements of between 0.5 dB to 1.43 dB. This scheme avoids adaptation overheads at the encoder by computing allocation rates at the decoder and using a feedback channel to transmit them back to encoder. The effectiveness of such schemes comes at the cost of additional sensing resources.

C. Full sensing techniques

Wang et al. proposed an ABCS scheme consisting of a base and enhancement layer [51]. The base layer uses BCS to compress each block independently using a low fixed measurement rate. The blocks are then classified into smooth, texture and other blocks depending on whether the average size of the pixel variances in each block is below threshold T1 (smooth), between T1 and T2 (other) or above T2 (texture). Additional measurements are taken from each block, with
decode $512 \times 512$ images in 30 to 70 ms. For $512 \times 512$ Lena, it is reported that AS combined with separable MMSE reconstruction, improves PSNR by 5.03 dB over a non-AS technique. The technique requires that all pixels are available at the encoder side to compute the block variance.

Canh varies the block measurement rate based on the number of edge pixels (ABCS-Edge), the $L_1$ norm of the horizontal and vertical pixel gradients (ABCS-GradL1), the standard deviation (ABCS-Std), or the preferred proposal based on an empirically derived look-up table (ABCS-Proposed) [37]. The proposed method was compared with non-adaptive BCS and the other three adaptive schemes, on 12 test images of size $512 \times 512$ at $r = 0.2, 0.3$ and 0.4. All adaptive schemes performed better than BCS at all sub-rates with the best scheme, ABCS-Proposed some 2 dB to 4 dB better than BCS. The techniques proposed and investigated compute the statistics from pixels available at the encoder. However, the author alludes to the possibility of using statistics collected from the previous video frame.

Wang et al. defined a spatial frequency (SF) measure, which is a scalar number for each block, representing the root mean square of the average horizontal and vertical pixel differences in a block, that is used to allocate rate unequally in textural and other blocks [38]. Textural blocks are defined as those whose SF measure exceeds a threshold. The authors compare their ABCS-SF-D technique with multiscale block compressive sensing with smooth projecting Landweber (MS-BCS-SPL) [39] at rates between 0.1 and 0.9, on the compressive sensing with smooth projecting Landweber (MS-BCS-SPL) [39] at rates between 0.1 and 0.9, on the $512 \times 512$ images Lena, Barbara, Peppers, Mandrill, Goldhill and Boat. Their proposed technique performs better on all images at and above $r = 0.3$. MS-BCS-SPL is usually better below 0.3 except on Mandrill. This scheme assumes that the original pixels are available to compute the SF allocation factor.

Wu et al. proposed a scheme which first partitions an image into blocks, computes the DCT on the blocks, uses a Just Noticeable Distortion (JND) model based on the Human Vision System to set DCT coefficients below threshold to zero [40]. The coefficients are then permuted randomly within the whole image such that each block ends up with roughly the same number of non-zero Transform Coefficients. The permuted TCs are then measured employing a standard CS technique using a Gaussian measurement matrix and reconstructed into an image using TV minimization. Results based on $256 \times 256$ images Lena, Cameraman, House, Mandrill, Boat, Airplane, Barbara and Peppers show that the proposed JND adaptive scheme is better than BCS-SPL [3 in [40]], LiRan [5 in [40]] and CRP [33] proposed in the literature when $r$ is at or above 0.3. Below 0.3, BCS-SPL performs better. This scheme requires the computation of a full set of 2D DCT coefficients in the first instance.

III. COMPRESSION IMAGE SENSING AND RECONSTRUCTION

In sensing applications, it is often the case that information needs to be extracted from a set of measured data points. Taking $f \in \mathbb{R}^N$ to be the signal of interest and $y \in \mathbb{R}^M$ the measured signal, then the relationship between the two can be written as:

$$y = \Phi f$$ (1)

where $\Phi \in \mathbb{R}^{M \times N}$ is the measurement matrix sampling signal $f$ to give $y$. If linear measurements of $f$ are taken, then the reconstruction problem is reduced to solving a set of linear equations. The classical Nyquist-Shannon sampling theorem sets the sampling rate to twice the highest frequency for $f$ to be reconstructed.

In a CS framework, the signal of interest $f$ has a discrete representation $x \in \mathbb{R}^N$ in some transform domain with basis $\Psi \in \mathbb{R}^{N \times N}$ such that $f = \Psi x$. Vector $x$ is defined as sparse if it contains $S$ non-zero elements such that $S \ll N$.

The measured signal $y$ is therefore given as:

$$y = \Phi \Psi x$$ (2)

Grouping the measurement matrix and sparsifying basis matrix gives:

$$y = Ax$$ (3)

where $A = \Phi \Psi \in \mathbb{R}^{M \times N}$ is defined as the sensing matrix. The goal in CS is thus to represent the original signal using $M < N$ samples, substantially below the Nyquist rate. As the locations of the $S$ sparse non-zero values of $x$ are not known in advance, reconstruction involves solving the undetermined set of equations given in equation (3) [41].

Two main questions arise from equation (3): (i) how should the measurement, and hence sensing matrix be chosen?; and (ii) how does one solve for $x$? As the solution of (3) involves locating the non-zero components in vector $x$, the most natural approach is an $L_0$ minimization which can be written as:

$$\min_x ||x||_0 \quad s.t. \ Ax = y$$ (4)

As $L_0$ minimization is NP-hard and thus computationally intractable, a common approach is re-write the problem as an $L_1$ minimization:

$$\min_x ||x||_1 \quad s.t. \ Ax = y$$ (5)

Solving equation (5) can now be achieved via a linear programme (LP) which is tractable [41]. This is commonly referred to as basis pursuit (BP) [41]. Other reconstruction methods are introduced in section III.

In real world applications noise is always present and re-writing (3) to take this into account gives:

$$y = Ax + n$$ (6)

where $n \in \mathbb{R}^M$. The convex relation in (5) does not hold in this instance. However, it can be replaced by BP denoising (BPDN) [41] giving:

$$\min_x ||x||_1 \quad s.t. \ ||Ax - y||_2 < \epsilon$$ (7)

where $\epsilon$ depends on $n$.

An important feature of the sensing matrix is that it should retain the information pertinent to the sparse signal $x$. There
are several conditions that can shed some light as to whether a measurement matrix possess this feature. Three well known properties are: (i). The restricted isometry property (RIP); (ii). The null space property; and (iii). Mutual coherence. The sensing matrix that satisfies any of these three properties possess the required information preservation feature [41].

It is found that when $A$ is as a random Gaussian matrix, it satisfies the RIP property [41] and is often used as the measurement matrix in image CS. Simpler matrices, such as random Welsh-Hadamard matrix are also possible, and have added implementation advantages, especially on mechanical DMD cameras [5].

A. Reconstruction Algorithms

Many compressive sensing algorithms exist to recover the sparse signals from compressive samples. Recently, Pilastri and Tavares presented a taxonomy of reconstruction algorithms in compressive sensing [42]. In this taxonomy, the algorithms are grouped into six clusters: Convex Relaxation; Greedy; Non-convex; Iterative; Bregman Iterative; and Combinatorial.

Reconstruction algorithms are characterized by their inherent complexity, and the minimum number of compressive samples from which they can reliably recover the sparse coefficients, and hence reconstruct the signal.

In image and video coding, we encounter signals whose transform in some domain is only approximately sparse, and hence we need reconstruction algorithms which can recover the predominant $S$ sparse components in the presence of noise consisting of a long tail of small, non-zero components. If the sorted coefficients $x_i$ decay with a power law, such that:

$$|x_i| \leq c \cdot i^{-p}$$

where $c$ is a constant and $p \geq 1$, then $x_i$ is called p-compressible and it can be well approximated by an exact sparse signal. The transform coefficients of 2D image and compressible and it can be well approximated by an exact sparse signal. The transform coefficients of 2D image and video data fit this model.

The sensing matrix that satisfies any of these three properties are: (i). The restricted isometry property (RIP); (ii). The null space property; and (iii). Mutual coherence. It is shown that adding the Onsager correction term $1/\delta z_t^{-1}(\eta_t - \eta_{t-1})$ to (9), derived from belief propagation in graphical models, improves the convergence of the ISTA solution. The operator $\langle u \rangle$ performs the component-wise mean of the vector $u$, that is $\langle u \rangle \equiv \sum_{i=1}^{N} u_i/N$, $\eta_t$ is the derivative of $\eta_t$ and $\delta = M/N$ (equivalent to $C_R$ in this paper).

The AMP algorithm then finds $x$ iteratively by solving (11) and (12) below:

$$x^{t+1} = \eta_t(A^*z^t + x^t)$$

$$z^t = y - Ax^t + \frac{1}{\delta} z_t^{-1}(\eta_{t-1} - \eta_t) (A^*z^{t-1} + x^t)$$

The authors claim that “D-AMP offers state-of-the-art CS recovery performance while operating tens of times faster than competing methods”. Indeed, AMP combined with a BM3D denoiser recovers a $128 \times 128$ pixel image, with close to state-of-the-art PSNR, in tens of seconds at a sampling rate of 10%.

IV. NON-ADAPTIVE LINEAR 2D TRANSFORM SENSING

In this paper we focus on the use of non-random, linear 2D Transform sensing matrices such as the 2D-DCT or the 2D Discrete Walsh-Hadamard Transform (2D-DWHT). To accomplish compressive sensing, capitalizing on the energy compaction property of the 2D transform, $M$ TCs are sensed, sorted in increasing order of sequency using the zigzag scanning procedure present in JPEG [14] for example. If the objective is not compressive sensing, but compression, all the TCs may be collected and then only those exceeding some threshold are retained.
The advantage of non-random sensing is that the signals can be decoded very rapidly at the decoder, such that the collected images or frames can be used in real-time image and video transmission for example, or to decrease the measurement time in MRI. However, it is possible to use techniques developed for reconstruction in compressive image sensing to deblock an image and increase the quality measured, for example, using PSNR or the more subjectively matched SSIM [18].

A. The compressive sensing L-\{DCT\|DWHT\}-ZZ algorithm

The compressive sensing, Linear 2D-DCT or 2D-DWHT with zigzag scan (L-\{DCT\|DWHT\}-ZZ) algorithm is described in Algorithm 1. This technique has been proposed and studied by a number of authors [44], [45]. The block-based algorithm includes sensing the whole image if the block is the same size as the image.

B. The L-\{DCT\|DWHT\}-\{THB\|THI\} algorithms

When using DCT or DWHT basis for reconstruction, optimal Mean Square Error (MSE) results are obtained by sensing all TCs and then retaining only those TCs which exceed a threshold. Two full sensing (FS) algorithms will now be described.

The Threshold over Blocks (THB) algorithm (Algorithm 2) selects the \( m \) largest TCs in a block, whereas the Threshold over Image (THI) (Algorithm 3) selects the \( M \) largest coefficients in an image and results in \( m_i \) TCs per block. The algorithms generalize to sensing the whole image if the block size is the same size as the image.

Both the threshold-based algorithms are inherently adaptive. Although the THB algorithm selects the same number of measurements per block, it selects the \( m \) largest ones, and not in zigzag order, and hence it adapts the coefficient position block-by-block. The THI algorithm collects a different number of TCs per block since these are selected from amongst all the blocks. Since 2D transforms concentrate the energy into the low spatial frequency coefficients, the DC and low frequency components will always be present in each block, and it is also possible to enforce a minimum number of TCs to be included in every block.

C. Deblocking Block-Based L-\{DCT\|DWHT\}-\{ZZ\|THB\|THI\}

BCS [13] of images exhibits blocking artefacts when block-based reconstruction is used, and the compression factor \( C_F := N/M \geq 1 \) is high. Deblocking improves the subjective quality considerably and can also lead to an improvement of objective quality measures. We propose to deblock block-based L-\{DCT\|DWHT\}-\{ZZ\|THB\|THI\} using BM3D [22], which is used frequently in the literature, or two faster algorithms proposed below, which use the whole-image forward and reverse 2D DCT transform pair on the decoded image.

\[ C_R := M/N \leq 1 \]

---

### Algorithm 1: L-\{DCT\|DWHT\}-\{ZZ\|BM3D\|ZZF\|THF\}

**Input:** Image \( I \), compression ratio \( C_R = M/N \), block size \( B \), desired 2D DCT or 2D DWHT transform, optional \{BM3D\|ZZF\|THF\} filter selection

**Output:** Image \( O \) sensed using L-\{DCT\|DWHT\}, reconstructed using the relative inverse transform and optionally deblocked using \{BM3D\|ZZF\|THF\}

1. Crop and partition \( I \) into \( n_B = \lceil H/B \cdot \lceil W/B \rceil \rceil, B \times B \) non-overlapping blocks;
2. Calculate the number of non-adaptive measurements per block \( m = \lceil C_R \cdot B^2 \rceil \);
3. Collect \( m \) DCT or DWHT coefficients in zigzag order from each of the \( n_B \) blocks;
4. Transmit \( M = m \times n_B \) measurements;
5. Reconstruct the Image from the \( m \) TCs per block using 2D IDCT or 2D IDWHT;
6. Optionally, deblock the image by performing BM3D, ZZF or THF filtering.

### Algorithm 2: L-\{DCT\|DWHT\}-THB

**Input:** Image \( I \), compression factor \( C_R = M/N \), block size \( B \), desired 2D DCT or 2D DWHT transform

**Output:** Image \( O \) sensed using L-\{DCT\|DWHT\}-THB, reconstructed using the relative inverse transform

1. Crop and partition \( I \) into \( n_B = \lceil H/B \cdot \lceil W/B \rceil \rceil, B \times B \) non-overlapping blocks;
2. Collect all \( B \times B \) L-\{DCT\|DWHT\}-ZZ coefficients from each of the \( n_B \) blocks;
3. Select the \( m = \lceil C_R \cdot B^2 \rceil \) largest coefficients in each block;
4. Transmit \( M = m \times n_B \) measurements;
5. Reconstruct the Image from the \( m \) TCs per block using the 2D IDCT or 2D IDWHT.
TABLE I: Comparison of ZZF, THF and BM3D deblocking on the 256 × 256 Image Set, with best PSNR and SSIM in bold.

| Algorithm: | L-[DCT/DWHT]-THI |
|------------|------------------|
| **Input:** | Image \( I \), compression ratio \( C_R = M/N \), block size \( B \), desired DCT or DWHT basis |
| **Output:** | Image \( O \) sensed using L-[DCT/DWHT]-THI, reconstructed using the relative inverse transform |

1. Crop and partition \( I \) into \( n_B = \lfloor H/B \rfloor \cdot \lfloor W/B \rfloor \), \( B \times B \) non-overlapping blocks;  
2. Collect all \([\text{DCT/DWHT}]\) coefficients from all blocks;  
3. Select the \( M \) largest coefficients from all blocks;  
4. Transmit the \( M \) measurements;  
5. Reconstruct the Image on a block by block basis using 2D IDCT or 2D IDWHT.

The Zigzag scan Filter (ZZF) deblocks an L-[DCT/DWHT]-[ZZ/THF/THI] sensed image which has been reconstructed block-by-block, by performing the L-DCT-ZZ algorithm on the whole image and retaining only the \( M \) largest components where \( M = \lfloor (N/C_R) \rfloor \), where \( \lfloor x \rfloor \) is the integer part of \( x \). The Thresholding filter (THF) first computes a 2D-DCT of the whole image and retains the \( M \) largest TC components before re-inverting the filtered transform.

If the image is non-square, then it is first split into two overlapping square sub-images, which are deblocked independently. The two sub-images are then re-formed into the rectangular image by overlapping them and averaging the pixels in the overlap region.

Tables I and II show the PSNR, SSIM and execution time for ZZF, THF and Fast BM3D (FBM3D) (22), (46) (BM3D with the fast ‘Lc’ option set, and \( C_R \) set to \( C_F \)) for the two image sets. PSNR is calculated using the equation \( \text{PSNR}=10 \log_{10} (255^2 / \text{MSE}) \) where MSE is the mean square error. The SSIM values were calculated using the ssim_index.m Matlab script, downloaded from https://www.cns.nyu.edu/~lcv/ssim/ssim_index.m, with default settings. Execution time was estimated using the Matlab cputime function.

Deblocking is mostly required when \( C_R \leq 0.1 \). With both Image Sets, BM3D improved the reconstruction most for \( C_R \) between 0.1 and 0.01, followed very closely by THF. Since the latter executes 4 to 22 times faster, it is the preferred solution.

The THF deblocking technique generates better images, subjectively. Although BM3D returns better PSNR and SSIM results on the 256 × 256 image set (fig. I), blocking is not...
### TABLE III: Benchmarking block-based algorithms.

| Block-based Algorithm | Block Size | 256 x 256 | 512 x 512 |
|-----------------------|------------|-----------|-----------|
|                       |            | PSNR      | Time      | PSNR      | Time      |
|                       |            | dB        | seconds   | dB        | seconds   |
| L-DCT-ZZ              | 32         | 31.81     | 0.0136    | 33.61     | 0.047     |
| L-DCT-THI             | 32         | 40.88     | 0.0149    | 41.87     | 0.058     |
| L-DCT-THB             | 32         | 37.92     | 0.0144    | 39.63     | 0.051     |
| MS [39]               | 16 x 32 x 64 | 30.97   | 4.63      | 32.25     | 12.40     |
| MH [47]               | 32         | 30.50     | 14.18     | 31.95     | 58.15     |
| MH-MS [47]            | 16 x 32 x 64 | 31.70   | 17.64     | 32.87     | 68.98     |
| MBTV-NLLM [48]        | 32         | 30.66     | 23.00     | 30.92     | 28.00     |
| MBTV-NLLM-CST [48]    | 32         | 34.89     | 10min+    | 34.37     | 10min+    |
| BM3D-AMP (30 iterations) [9] | 32       | 29.98     | 120.04    | 30.72     | 429.03    |

... (30 Iterations) [9] 
32
29.98
120.04
30.72
429.03

D. Benchmarking non-adaptive L-{DCT|DWHT}-\{ZZ|THB|THI\}

This section compares block-based L-{DCT|DWHT}-ZZ with state-of-the-art random CS techniques, JPEG, and the full sensing L-DCT-{THB|THI} algorithms. Full sensing allows the transform coefficients to be computed directly in the analogue domain during sensing and may still offer power advantages by avoiding digital domain computation.

The following algorithms were benchmarked by executing the code made available by their authors on the 256 x 256 (fig. 1) and 512 x 512 (fig. 2) image sets: MS [39], MH [47], MH-MS [47], MBTV-NLLM [48], MBTV-NLLM-CST [48] and D-AMP [9]. The results are presented in table III. The PSNR and SSIM values are reported, averaged over compression ratios $C_R = \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

When the block size is 32 x 32 pixels, L-DCT-ZZ performs, on average, 1.83 dB and 2.89 dB better than BM3D-AMP on 256 x 256 and 512 x 512 pixel images respectively. The average SSIM is 0.095 and 0.155 better as well. The algorithms were executed on the same workstation such that execution times can be compared. L-DCT-ZZ is reconstructed in 13.6ms and 47ms for 256 x 256 and 512 x 512 pixel images respectively.

The striking result in table III is that L-DCT-ZZ is only surpassed by MBTV-NLLM-CST in terms of PSNR and SSIM, on both 256 x 256 and 512 x 512 image sets. The reconstruction time for L-DCT-ZZ is less than 14ms and 50ms for these resolutions respectively, whereas MBTV-NLLM-CST takes tens of minutes to reach its superior results. Note that the non-compressive sensing L-DCT-THB and L-DCT-THI outperform MBTV-NLLM-CST by large margins on both image sets but are reconstructed in milliseconds rather than minutes.

V. Adaptive Compressive Image Sensing Methods

The non-adaptive, block-based, deterministic L-DCT-ZZ algorithm is described in section IV. The image $I_o$ is partitioned into non-overlapping $B \times B$ pixel blocks that are then 2D-DCT transformed. $m$ transform coefficients (TCs) are retained from each block achieving a compression ratio $C_R = m/B^2$. In the adaptive scheme, the number of TCs collected varies according to the sparsity of each block. Fig. 3 shows a block diagram of the proposed adaptive system.

Two techniques for adapting the number of linear 2D-DCT transform coefficients in block compressive sensing are proposed, one in the spatial domain and the other in the 2D-DCT transform domain. Both techniques use two phases to encode an image adaptively. In the first phase, $m_1$ measurements are collected from each block $i$ and used to determine the number of additional transform coefficients, $m_2$, to be collected from all blocks in the second phase.

In the case of the DD technique, the $m_1$ measurements are collected as the DC and low-frequency components as shown in fig. 4 and determine the next $m_2$ measurements to be collected in phase 2.

All the measurements collected from both phases are considered for reconstruction purposes and hence the total number of measurements collected $M = \sum_{i=1}^{n} (m_1 + m_2)$.

The methods used to estimate $m_2$ must not be computationally intensive because these will be implemented at the encoder side, and a target application for the methods described in this paper is for autonomously powered wireless image and video sensors. However, it is also possible to transmit the $m_1$ measurements to a decoder and use more elaborate algorithms at the decoder to estimate $m_2$ per block which information is then fed back to the encoder as has been proposed in the literature [26, 27, 30]. In power-constrained applications, it is necessary to consider whether the power required to operate a receiver module to receive the $m_2$ feedback information is greater than the power required to just compute $m_2$ at the encoder. The requirement to wait for the reception of feedback information will also add to the reconstruction time of the image, although the increase may be contained if the sensing node is close to the receiver and the data rate of the link is sufficiently high.

Fig. 3 shows the block AL-DCT measurements reconstructed in real-time at the decoder, using an inverse 2D-DCT transform and reconstituted into the output image $I_o$. The number of phase-two TCs, $m_2$, is transmitted as side information. If better reconstruction quality is required, full-image CS reconstruction is invoked to output image $I_o$, albeit with a large reconstruction time penalty.
The measurement of the BBV can either be treated as a necessary overhead, or it is accounted for by reducing the number of transform coefficients that can be collected. This paper takes the latter view and tries to minimize the number of TCs that can be collected. This is accomplished in three ways. First, only the TV in the top row and left-hand column borders are measured in each block. Since the blocks stack to form an image, the right-hand TV is obtained from the next block to the right, and the bottom row TV from the block below the current row. This reduces the number of pixels that need to be measured per block, by one half to \((4B - 2)\). Secondly, rather than collect pixel values, it is possible to collect pixel difference measurements, which involves only two adjacent pixels at a time. Thus, it is possible to reduce the measurements to around \(B\) per block. Thirdly, it is possible that not all adjacent pixel measurements are necessary and that the BBV can be itself estimated from fewer measurements. It is thus possible to collect a pixel-difference measurement every \(L\) pixels around the block border.

The right-hand blocks and the bottom blocks do not have right-hand and bottom neighbours that can contribute the missing edges of the boundary. In these cases, the right-hand and bottom TVs can be measured from the current block.

### Algorithm 4: Adaptive AL-{DCT|DWHT}-BBV

**Input:** Image \(I\), compression factor \(C_F\), block size \(B\), reconstruction algorithm (RA) = \{IDCT2|D-AMP|DAMP-D|IDA\}.  
**Output:** Image \(O\) sensed using AL-DCT-BBV and reconstructed using a RA.

1. Crop and partition image, \(I\) into \(n_B = \lfloor H/B \rfloor \cdot \lfloor W/B \rfloor\), \(B \times B\) blocks;  
2. Collect \(m_1^1 = 2 \cdot n_S\) samples per block where \(n_S = \lfloor B/C_R \rfloor\) measurements are equally spaced in the top and left-hand block border;  
3. Estimate the number \(m_2^1\) of additional samples to collect from each block \(i\) where \(m_2^1\) is given by equation (21):  
4. Collect the \(m_2^1\) TCs from each block \(i\);  
5. Transmit \(M = \sum_{i=1}^{n_B} (m_1^1 + m_2^1)\) measurements;  
6. Reconstruct \(O\) from the \(M\) received samples using the selected RA.

Fig. 5 shows BBV measurements in a \(32 \times 32\) pixel block. The variation measurements in the current block, which consist of the absolute differences \(|P_1 - P_2|\) of two adjacent pixels \(P_1\) and \(P_2\), are outlined in black. \(X_0\) is the pixel offset of the first variation measurement in a row, \(Y_0\) is the pixel offset of the first measurement in a column, and \(L\) is the stride between pixel variation measurements. In the figure, \(X_0 = 1\), \(Y_0 = 1\) and \(L = 4\). We acquire Phase 1 measurements that decrease as \(C_F\) increases. Hence we set:

\[
L = \lfloor C_F \rfloor 
\]

and

\[
X_0 = Y_0 = \lfloor L/2 \rfloor
\]
The number of measurements $n_s$ per block side is given by:

$$n_s = \left\lceil \frac{B}{L} \right\rceil$$  \hspace{1cm} (18)

and the number of measurements used to calculate the BBV is $4 \cdot n_s$, with half of them in the current block, and the other half in adjacent blocks, shaded in grey in fig. [5].

If $C_F > B$, $n_s$ will be zero according to equation (18). In this case the algorithm reverts to being non-adaptive.

The number of blocks $n_B$ in an image of $H$ by $W$ pixels is given by:

$$n_B = \frac{H \cdot W}{B^2}$$  \hspace{1cm} (19)

Then the number of BBV measurements $M_{BBV}$ required in phase one is given by:

$$M_{BBV} = 2 \cdot n_s \cdot n_B + \frac{(H+W)}{L}$$  \hspace{1cm} (20)

These BBV measurements, besides being used to calculate the number of measurements $m_i^2$ required in phase two, can also be transmitted to the decoder to serve as additional reconstruction measurements. Therefore, the number of measurements used in the reconstruction will be equal to $M$. The number of measurements in block $i$ during phase 2, $m_i^2$, is given by:

$$m_i^2 = \left( M - M_{BBV} \right) \cdot \frac{BBV_i}{\sum_{i=1}^{N} BBV_i}$$  \hspace{1cm} (21)

where $BBV_i$ is the BBV of block $i$, that is the sum of the absolute difference of two adjacent pixels at each measurement point, around the boundary of the block. Half the measurements are collected from block $i$, and the other half from the adjacent blocks to the right and below, as shown in fig. [5]. $m_i^2$ has to be capped at $B^2 - m_i^2$ and any remaining measurements can be collected from other non-fully-measured blocks.

### B. Adaptive L-DCT-ZZ in the DCT domain (DD)

The Adaptive L-DCT-ZZ algorithm in the DCT domain, AL-DCT-DD, described in Algorithm [5] uses $m_i^2$ linear 2D DCT measurements per block $i$, in zigzag scan order, to compute the number of measurements $m_i^2$ to collect in phase 2. Several authors have considered first capturing a full set of measurements ($M = N$) to compute the value of all the pixels at the encoder and then adapting the number of measurements to transmit per block, as reported in Section [4]. These Full Sensing techniques are feasible if the power required to sense the transform coefficients directly in the optical domain, is...
However, if compressive sensing is used, the strategy is to assign less sensing power than Full Sensing techniques.

Using digital computations. Compressive Sensing techniques block. The shaded measurements are imported from adjacent images. The iterative denoising-based reconstruction algorithm investigates why the techniques improve performance of the L-DCT-ZZ algorithm.

Using the 2D DCT as the sparsifying domain, the number of TCs in each block, $m_i$, that maximize PSNR, can be found using the L-DCT-Thi algorithm, that is measure all TCs in all blocks, then select the M largest ones, irrespective of the blocks in which they occur. The higher the number of TCs in a block i, the less the sparsity in the block, and the more samples that are required from that block.

It is required to show that the $\hat{m}_i$ BBV and DD measurements are correlated with $m_i$. The correlation coefficient $r$ and the confidence level, $(1 - p)$, where $p$ is the probability of the null hypothesis (that there is no correlation, i.e., $r = 0$), were measured for the two image sets for $C_R = \{0.1...0.5\}$. In all cases the probability of the null hypothesis was always below 0.0001, indicating strong confidence in the correlation results. The correlation results $r = \text{Corr}(m_i, \hat{m}_i)$ for the $256 \times 256$ and $512 \times 512$ image sets are tabulated in Table IV.

In both cases, the best correlation is obtained with the DD algorithm. However, BBV returns better correlation for the Barbara image, which is usually a challenging image to compress using L-DCT-ZZ. The Peppers and Girl images are significantly better estimated by DD.

The correlation is strongest at $C_R = 0.5$. The lowest correlation occurs for BBV at $C_R = 0.1$ and may be due to the low level of measurements acquired to compute $m_i$, since the number of block boundary measurements is proportional to $C_R$. This indicates that BBV should probably have a minimum number of boundary measurements from a correlation perspective, although this could impact PSNR and SSIM, because each phase one measurement reduces the number of DCT measurements.

## VI. Analysis of the BBV and DD Adaptive Techniques

The BBV and DD algorithms use block boundary variation and the number of L-DCT-ZZ transform coefficients that exceed a threshold $T$, respectively, to estimate the number of adaptive measurements $\hat{m}_i$ in each block $i$. This section investigates why the techniques improve performance of the L-DCT-ZZ algorithm.

It is required to show that the $\hat{m}_i$ BBV and DD measurements are correlated with $m_i$. The correlation coefficient $r$ and the confidence level, $(1 - p)$, where $p$ is the probability of the null hypothesis (that there is no correlation, i.e., $r = 0$), were measured for the two image sets for $C_R = \{0.1...0.5\}$. In all cases the probability of the null hypothesis was always below 0.0001, indicating strong confidence in the correlation results. The correlation results $r = \text{Corr}(m_i, \hat{m}_i)$ for the $256 \times 256$ and $512 \times 512$ image sets are tabulated in Table IV.

In both cases, the best correlation is obtained with the DD algorithm. However, BBV returns better correlation for the Barbara image, which is usually a challenging image to compress using L-DCT-ZZ. The Peppers and Girl images are significantly better estimated by DD.

The correlation is strongest at $C_R = 0.5$. The lowest correlation occurs for BBV at $C_R = 0.1$ and may be due to the low level of measurements acquired to compute $m_i$, since the number of block boundary measurements is proportional to $C_R$. This indicates that BBV should probably have a minimum number of boundary measurements from a correlation perspective, although this could impact PSNR and SSIM, because each phase one measurement reduces the number of DCT measurements.

## VII. Reconstructing Non-Random CS Using Iterative Denoising-Based Algorithms

In this section, we build on the work of [9] to design iterative algorithms that better reconstruct non-random CS images. The iterative denoising-based reconstruction algorithm is represented in block diagram form in fig. 6. In D-AMP:

$$m_i^2 = \left\lfloor n_B \cdot m_i^1 \right\rfloor$$
Fig. 6: Generic Block Diagram for Iterative Denoising Reconstruction Algorithms: D-AMP, DAMP-D and IDA.

(a)  

(b)  

Fig. 7: Reconstructing L-DCT-ZZ using D-AMP (when $D_F = 1.00$) and DAMP-D for $D_F > 1.00$. (a) Full PSNR range (b) Zoomed PSNR range showing detail at high PSNR. Results for the Barbara image sensed at $C_F = 16^1$.

Fig. 8: Reconstructing AL-DCT-ZZ using D-AMP (when $D_F = 1.00$) and IDA for $D_F > 1.00$. (a) Full PSNR range (b) Zoomed PSNR range showing detail at high PSNR. Results for the Barbara image sensed at $C_F = 10^1$.

\[
\alpha^t = \frac{1}{\delta} \text{div}[D_{\sigma^{t-1}}(x^{t-1} + A^*z^{t-1})] \\
\]  

so that $\alpha^t z^{t-1}$ is the Onsager term. The $\Delta$ block introduces a delay of one iteration, $D_{\sigma_t}$ is the denoiser block, and $r^t$ is given by:

\[
r^t = y - A x^t
\]
The other variables are as defined for D-AMP above.

To use D-AMP on non-random, block-based CS, the non-random forward transform is cast as a sensing matrix \( A \) which has a sparse representation:

\[
A = \begin{bmatrix}
B_{1,1} & \cdots & B_{1,i} \\
\vdots & \ddots & \vdots \\
B_{n_B,1} & \cdots & B_{n_B,n_B}
\end{bmatrix}
\] (25)

where \( B_{1,i} \) are \( m \times n \) basis matrices with \( m = \lfloor M \times C_R \rfloor \) where \( C_R \) is the compression ratio \( M/N \), \( n = B \times B \), \( B \) is the size of one side of the image block and \( n_B \) is the number of blocks in the image. In non-adaptive CS, \( m \) and \( n \) are constant since there are equal measurements per block. In adaptive CS, \( m \) varies per block. \( A^* \) is the transpose of \( A \).

When \( B \) is a random Gaussian block matrix, D-AMP frequently fails to recover some of the blocks correctly. This never happens when the row vectors of \( B \) are taken from the 2D-DCT basis functions, in zigzag order, with the first basis function being that for the DC component.

1) Modified D-AMP - DAMP-D: BM3D-AMP was derived based on a Gaussian distribution of the residual reconstruction error at each iteration [9]. When the measurement matrix is non-random 2D-DCT, rather than random Gaussian, this assumption no longer holds and reconstruction quality deteriorates. Fig. [9] shows that the Onsager term implements an adaptive integration of the reconstruction error, akin to integral control in a closed-loop feedback system. This inspires us to vary the integral control loop gain by introducing a damping
TABLE VI: L-DCT-ZZ, AL-DCT-BBV and AL-DCT-DD with IDCT, D-AMP, DAMP-D and IDA reconstruction on the 512×512 image set. Maximum PSNR and SSIM values are in bold.

| \( C_R \) | L-DCT-ZZ | L-DCT-DD-AMP | L-DCT-DD-D-AMP | L-DCT-IDA |
|---|---|---|---|---|
| | PSNR dB | SSIM | Time s | PSNR dB | SSIM | Time s | PSNR dB | SSIM | Time s |
| 0.01 | 22.85 | 0.5343 | 0.145 | 23.31 | 0.5241 | 0.147 | 23.64 | 0.5175 | 0.15 | 23.25 | 0.5223 |
| 0.02 | 22.33 | 0.5925 | 0.139 | 22.35 | 0.6185 | 0.145 | 22.48 | 0.6309 | 0.129 | 22.49 | 0.6316 |
| 0.04 | 25.70 | 0.6685 | 0.139 | 25.25 | 0.7004 | 0.117 | 26.49 | 0.7069 | 0.115 | 26.60 | 0.7084 |
| 0.10 | 28.63 | 0.7940 | 0.133 | 29.22 | 0.8128 | 0.122 | 29.45 | 0.8175 | 0.165 | 29.63 | 0.8200 |
| 0.20 | 31.62 | 0.8767 | 0.133 | 32.15 | 0.8845 | 0.128 | 32.35 | 0.8857 | 0.183 | 32.61 | 0.8908 |
| 0.30 | 33.82 | 0.9164 | 0.132 | 34.09 | 0.9177 | 0.154 | 33.40 | 0.9023 | 0.218 | 34.68 | 0.9242 |
| 0.40 | 35.90 | 0.9411 | 0.134 | 35.61 | 0.9367 | 0.209 | 36.27 | 0.9430 | 0.252 | 36.68 | 0.9455 |
| 0.50 | 38.06 | 0.9593 | 0.138 | 37.04 | 0.9494 | 0.248 | 38.44 | 0.9608 | 0.283 | 38.78 | 0.9614 |

2) Simplified D-AMP - IDA: Since the Onsager term requires the residual error to be Gaussian, it was hypothesized that a simplified version of the iterative denoising algorithm might provide better performance. Indeed, simplifying \( \alpha' \) to:

\[
\alpha' = \frac{1}{D_F} 
\]

was found to provide better performance than DAMP-D. The ensuing algorithm is called the IDA with damping factor \( D_F \).

Fig. 8 shows that IDA without damping, i.e. \( D_F = 1 \), does not reconstruct the Barbara image in image set 256×256 satisfactorily. The PSNR decreases on the first iteration and then oscillates around a level which is 10 dB lower. However, when \( D_F \) increases above 1, following the initial dip, the PSNR tends to increase. At above a damping value of 1.4, IDA increases the initial L-DCT-ZZ PSNR by around...
2 dB, the maximum value after 20 iterations being reached with \( D_F = 2 \).

3) **Tuning \( D_F \):** To find the optimal value of \( D_F \), the PSNR was plotted for \( D_F \) between 1.0 and 3.0 for various algorithms, image sets and compression ratios. For example, fig. 9 shows the PSNR versus \( D_F \) plotted for [A]L-DCT-{ZZ,BBV,DD}-IDA, for the 256 x 256 image set, with \( C_R = 0.10 \). \( D_F = 2.0 \) was found, empirically, to be a good compromise value across the compression ratios of interest.

![Fig. 9: PSNR versus \( D_F \) for [A]L-DCT-{ZZ,BBV,DD}-IDA for the 256 x 256 image set, with compression ratio \( C_R = 0.10^3 \).](image)

TABLE VII: Comparison of L-DCT-ZZ, AL-DCT-DD and AL-DCT-BBV with results in the literature. Best results are highlighted in bold. The underlined result is the only inferior [A]L-DCT-{ZZ,BBV,DD} result.

| Algorithm                  | Encoder Side | L-DCT-ZZ | AL-DCT-DD | AL-DCT-BBV |
|----------------------------|--------------|----------|-----------|------------|
| ABCS-TVAL3 [24]            | 30.82        | 37.38    | 38.68     | 39.35      |
| ABCS-Zhang [26]            | 29.21        | 30.06    | 31.21     | 33.73      |
| Proposed in [30]           | 31.08        | 32.88    | 34.44     | 34.16      |
| Decoder Side               |              |          |           |            |
| ABCS-Wang [31]             | 28.36        | 35.22    | 36.54     | 36.85      |
| Var-reg OP3 [32]           | 29.18        | 30.39    | 31.64     | 31.02      |
| TABLE I [35]               | 27.42        | 31.48    | 32.41     | 32.93      |
| ABCS-Canh [37]             | 32.16        | 34.37    | 35.62     | 35.59      |
| ABCS-SF-D [38]             | 32.64        | 32.26    | 33.19     | 33.84      |
| JND [40]                   | 28.35        | 32.13    | 33.36     | 33.34      |

VIII. **Empirical Investigation**

In this section, the two adaptive algorithms are evaluated empirically and compared with other published encoder-side and decoder-side adaptive CS algorithms, as well as with full sensing techniques. None of the techniques published in the literature were accompanied by source code. So, the method used to compare with these published results was to repeat the simulation on the same image test sets, and at the same compression ratios \( C_R = M/N \). The algorithms proposed in this paper were tested on a server equipped with an Intel Xeon CPU E5-160 v3 clocked at 3.50 GHz, with 32.00GB of RAM, running Matlab version 2019a on Windows 10. The D-AMP algorithm was downloaded together with the D-AMP toolbox from [49].

A. **Comparing the non-adaptive and adaptive algorithms when reconstructed using IDCT, D-AMP, DAMP-D and IDA**

The results in tables [V] and [VI] show that AL-DCT-DD improves AL-DCT-ZZ PSNR and SSIM at all resolutions and compression ratios studied. AL-DCT-BBV improves performance at both resolutions when \( C_R = 0.1\ldots0.5 \) but suffers marginal degradation when \( C_R = 0.01\ldots0.04 \) on the 256x256 image set. AL-DCT-DD is better in PSNR and SSIM terms than AL-DCT-BBV on the 256x256 image set. AL-DCT-BBV PSNR is better for 512x512 images when \( C_R = 0.1\ldots0.5 \).

Reconstructing [A]L-DCT-{ZZ,BBV,DD} with D-AMP improves PSNR when \( C_R = 0.1\ldots0.5 \) on both image sets, except for AL-DCT-BBV on the 512x512 image set. D-AMP leads to considerable degradation on both image sets when \( C_R = 0.01\ldots0.04 \).

Reconstructing [A]L-DCT-{ZZ,BBV,DD} with DAMP-D and IDA with \( D_F = 0 \) always improves PSNR and SSIM considerably for both resolutions and all compression ratios. IDA reconstruction is always superior to DAMP-D. The reconstruction time using IDA is also considerably lower than that using DAMP-D because of the simpler \( \alpha^t \) term (fig. 3).

The improvement in performance with CS reconstruction comes at the expense of reconstruction time with [A]L-DCT-{ZZ,BBV,DD} reconstructed in tens of milliseconds whereas D-AMP, DAMP-D and IDA require tens to hundreds of seconds.

The best PSNR results for both image sets, averaged over \( C_R = 0.1\ldots0.5 \), is achieved by AL-DCT-BBV-IDA, which at 35.02 dB and 35.74 dB are better than the respective MBTV-NLLM-CST results in table [III]. However, AL-DCT-DD-IDA performs better at lower compression ratios, i.e., \( C_R = 0.01\ldots0.04 \).

B. **Comparison with results in the literature**

L-DCT-ZZ and the two adaptive algorithms AL-DCT-{BBV,DD} achieve state-of-the-art performance, in PSNR and SSIM terms, across many image test sets used in the literature. Table VII summarizes PSNR results where [A]L-DCT-{ZZ,BBV,DD} performs better. Only ABCS-SF-D [38] achieves better results than L-DCT-ZZ. Besides performing better, the proposed algorithms can be decoded in tens of milliseconds on CPUs as shown in tables [III] [V] and [VI].

Sections VIII-C, VIII-D and VIII-E below discuss these Encoder Side, Decoder Side and Full Sensing results where our algorithms did not produce the best PSNR results or where performance is close.
marginally better than Stat-ACS [23]. Stat-ACS performs

C. Comparison with Encoder-Side ACS algorithms

Table VIII shows that AL-DCT-{BBV|DD}-IDA perform marginally better than Stat-ACS [23]. Stat-ACS performs 1.5 dB better than AL-DCT-{BBV|DD} without IDA reconstruction. However, Stat-ACS collects wavelet measurements over the whole Image rather than in image blocks. The
algorithm assumes a predictive model which is matched to the test images and the behaviour with generic images is not reported.

L-DCT-ZZ, AL-DCT-BBV and AL-DCT-DD perform significantly better than the ABCS algorithm in [24] as shown in table VII. AL-DCT-BBV averages 39.35 dB over four 512 × 512 test images at \( C_R = 0.2, 0.4, 0.6 \) as against 30.82 dB achieved by ABCS with TVAL3 reconstruction.

Liu et al. studied both directly and D-AMP reconstructed adaptive InVDS schemes [23]. As shown in table X both the non-adaptive and adaptive L-DCT based algorithms perform better than directly reconstructed InVDS algorithms, even without IDA reconstruction.

AL-DCT-{BBV|DD} can be better improved by IDA reconstruction than InVDS using WT, DCT, DWHT and D-AMP as shown in table X AL-DCT-BBV-IDA with \( D_F = 2 \) achieves the best results with InVDS WT D-AMP second. This is a significant result since InVDS is not block-based and has been tuned to the image set [25].

### D. Comparison with Decoder-Side ACS algorithms

AL-DCT-BBV and AL-DCT-DD perform better than Zhangs [26] and Akbari et al.’s [30] decoder-side proposed methods, as shown in table VII.

Table XI compares the [AL-DCT-{DD|BBV}]|IDA] algorithms with Zhu et al.’s Full-Sensing and Feedback (Decoder-Side) algorithms. Both AL-DCT-{BBV|DD} perform better than the EB and VB variants of Zhu et al.’s Solution 2 (feedback) algorithms and can be decoded in real-time without IDA reconstruction. The Solution 2, SB variant performs marginally better than the Solution 2, EB variant. Both these algorithms can be decoded without requiring a feedback iteration.

### E. Comparison with Encoder-Side Adaptive FS algorithms

Table VII contains simulation results which show that both AL-DCT-BBV and AL-DCT-DD perform better than the Full Sensing algorithms presented in [31], [32], [33], [37], [38] and [40].

The two algorithms perform better than AS-NESTA presented in [23]. AS-CRP-NESTA performs some 0.5 dB better than AL-DCT-{BBV|DD}-BM3D. However, CRP first requires computation of all 2D DCT coefficients and then performs CS on the permuted coefficients. It is not clear how this can be implemented physically on an SPC or MPC.

As shown in table XI the VB and SB Sol-1 variants presented in [27] are better than the AL-DCT-{BBV|DD}-IDA algorithms with the SB variant performing best. AL-DCT-{BBV|DD}-IDA perform marginally better than the EB variant. However, the three Sol-1 algorithms assume that all the image pixels are available at the encoder to classify the blocks whereas our schemes do not.
IX. CONCLUSIONS AND FURTHER WORK

Two adaptive algorithms, AL-DCT-BBV and AL-DCT-DD, have been described that adapt the number of non-random, linear transform coefficients collected from image blocks, in block-based compressive image sensors. Both the Block Boundary Variation and the DCT Domain techniques to estimate the number of TCs per block have been shown to correlate well the number of TCs as estimated by the full 2D DCT analysis. The schemes can be implemented on Single Pixel Cameras, Multi Pixel Cameras and Focal Plane Processing Image Sensors. Reconstruction is possible in tens of milliseconds on CPUs, although GPU acceleration is possible.

A block diagram representation of the D-AMP algorithm has been presented and used to describe the original D-AMP algorithm, a modified version, DAMP-D, which damps the Onsager term with a factor $D_F$ and a simplified version termed the Iterative Denoising Algorithm. The IDA simplifies the Onsager term to $z^t/D_F$ and achieves better results, reconstructing $\{A|L-DCT-\{ZZ\}|BBV|DD\}$ algorithms, than D-AMP and DAMP-D. Both modified algorithms can be used as a post-processing technique on the AL-DCT-$\{BBV|DD\}$ algorithms. Setting $D_F = 2.0$ has been empirically found to be close to optimal for $256 \times 256$ and $512 \times 512$ image sets.

$\{A|L-DCT-\{BBV|DD\}\}$ achieve state-of-the-art performance in Adaptive Block CS of images. They can be used in non-GPU-assisted, real-time applications, such as video capture with SPCs, MPCs or FPPIS. DAMP-D and IDA reconstruction improves performance further, but not in real time.

The improvement in performance by damping the Onsager term has been examined empirically and can benefit from further theoretical analysis.

APPENDIX A

EMPIRICAL OPTIMIZATION OF T FOR AL-DCT-DD WITH $B = 32$

For both $256 \times 256$ and $512 \times 512$ image sets, the value of $T$ was varied to obtain the best PSNR results. PSNR values when $T = \{15, 30, 60\}$ are shown in table XII. The bold PSNR values show the best empirical values for a given $C_R$ (or $C_F = 1/C_R$). The conditions shown in table XII were included in the DD algorithm and given the resolution of the image $H = 256$ or $512$ and the value of $C_F$, determine the value of $T$ from the set $\{15, 30, 60\}$.

ACKNOWLEDGMENTS

The authors would like to thank and acknowledge funding from the Cambridge Trust, the Engineering and Physical Sciences Research Council for Doctoral Training in Sensor Technologies and Applications (EP/L015889/1), and the Endeavour (Malta) Scholarships Scheme.

REFERENCES

[1] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 489–509, Feb 2006.

[2] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, April 2006.

[3] W. B. Johnson and J. Lindenstrauss, “Extensions of Lipschitz mappings into a Hilbert space,” Contemporary mathematics, vol. 26, no. 1, 1984.

[4] C. Peckens and J. P. Lynch, “Utilizing the cochlea as a bio-inspired compressive sensing technique,” Smart Materials and Structures, vol. 22, no. 10, p. 105027, 2013.

[5] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, “Single-pixel imaging via compressive sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 83–91, March 2008.

[6] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” IEEE Transactions on signal processing, vol. 41, no. 12, pp. 3397–3415, 1993.

[7] D. Baron, S. Sarvotham, and R. G. Baraniuk, “Bayesian compressive sensing via belief propagation,” IEEE Transactions on Signal Processing, vol. 58, no. 1, pp. 269–280, 2009.

[8] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” Proceedings of the National Academy of Sciences, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[9] C. A. Metzler, A. Maleki, and R. G. Baraniuk, “From denoising to
compressed sensing,” IEEE Transactions on Information Theory, vol. 62, no. 9, pp. 5117–5144, 2016.
[10] A. Mousavi, A. B. Patel, and R. G. Baraniuk, “A deep learning approach to structured signal recovery,” in 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 2015, pp. 1336–1343.
[11] L. Gan, “Block compressed sensing of natural images,” in 2007 15th International Conference on Digital Signal Processing, July 2007, pp. 403–406.
[12] S. Mun and J. E. Fowler, “Block compressed sensing of images using directional transforms,” in 2010 Data Compression Conference, March 2010, pp. 547–547.
[13] J. E. Fowler, S. Mun, E. W. Tramel et al., “Block-based compressed sensing of images and video,” Foundations and Trends in Signal Processing, vol. 4, no. 4, pp. 297–416, 2012.
[14] I. IEC, “Information technology-digital compression and coding of continuous-tone still images: Requirements and guidelines,” Standard, ISO IEC, pp. 10.918–1, 1994.
[15] I. ISO, I. JTC et al., “Information technology-jpeg 2000 image coding system part 1: Core coding system,” ISO/IEC 15444-1, 2001.
[16] H. Series, “Audiovisual and multimedia systems; infrastructure of audiovisual services and coding of moving video. H.264, advanced video coding for generic audiovisual services,” International Telecommunication Union. Version, vol. 12, 2007.
[17] I. Rec., “H.265 and ISO/IEC 23008-2: High efficiency video coding (HEVC),” 2013.
[18] Zhigang Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600–612, April 2004.
[19] X. Yuan, G. Huang, H. Jiang, and P. A. Wilford, “Block-wise lessnless compressive camera,” in 2017 IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 31–35.
[20] S. Jafarpour, “Deterministic compressed sensing,” PRINCETON UNIV NJ, Tech. Rep., 2011.
[21] M. Robucci, J. D. Gray, L. K. Chiu, J. Romberg, and P. Hasler, “Compressive sensing on a CMOS separable-transform image sensor,” Proceedings of the IEEE, vol. 98, no. 6, pp. 1089–1101, 2010.
[22] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising with block-matching and 3d filtering,” in Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, vol. 6064. International Society for Optics and Photonics, 2006, p. 606414.
[23] A. Averbuch, S. Dekel, and S. Deutsch, “Adaptive compressed image sensing using dictionaries,” SIAM Journal on Imaging Sciences, vol. 5, no. 1, pp. 57–89, 2012.
[24] Nguyen et al., “Rate allocation method for block-based compression sensing,” Journal of Broadcasting Engineering, vol. 20, no. 3, pp. 398–407, 2015.
[25] J. Liu and C. Ling, “Adaptive compressed sensing using intra-scale variable density sampling,” IEEE Sensors Journal, vol. 18, no. 2, pp. 547–558, Jan 2018.
[26] X. Zhang, A. Wang, B. Zeng, L. Liu, and Z. Liu, “Adaptive block-wise compressive image sensing based on visual perception,” IEEETRANSAC TIONS ON INFORMATION AND SYSTEMS, vol. 96, no. 2, pp. 383–386, 2013.
[27] S. Zhu, B. Zeng, and M. Gabbouj, “Adaptive sampling for compressed sensing based image compression,” Journal of Visual Communication and Image Representation, vol. 30, pp. 94–105, 2015.
[28] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in Proceedings of 27th Asilomar conference on signals, systems and computers. IEEE, 1993, pp. 40–44.
[29] S. Becker, J. Bobin, and E. J. Candès, “Nesta: A fast and accurate first-order method for sparse recovery,” SIAM Journal on Imaging Sciences, vol. 4, no. 1, pp. 1–39, 2011.
[30] A. Akbari, D. Mandache, M. Trocan, and B. Granado, “Adaptive saliency-based compressive sensing image reconstruction,” in 2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW), July 2016, pp. 1–6.
[31] A. Wang, L. Liu, B. Zeng, and H. Bai, “Progressive image coding based on an adaptive block compressed sensing,” IEICE Electronics Express, vol. 8, no. 8, pp. 575–581, 2011.
[32] W. Guicquero, A. Dupret, and P. Vandergeynst, “An adaptive compressive sensing with side information,” in 2013 Asilomar Conference on Signals, Systems and Computers, Nov 2013, pp. 138–142.
[33] Z. Gao, C. Xiong, L. Ding, and C. Zhou, “Image representation using block compressive sensing for compression applications,” Journal of Visual Communication and Image Representation, vol. 24, no. 7, pp. 885–894, 2013.
[34] H. Zheng and X. Zhu, “Sampling adaptive block compressed sensing reconstruction algorithm for images based on edge detection,” The Journal of China Universities of Posts and Telecommunications, vol. 20, no. 3, pp. 97–103, 2013.
[35] J. Luo, Q. Huang, S. Chang, and H. Wang, “Fast reconstruction with adaptive sampling in block compressed imaging,” IEEE Electronics Express, vol. 11, no. 6, pp. 2014056–2014056, 2014.
[36] ——, “Hardware efficient architecture for compressive imaging,” IEEE Electronics Express, vol. 11–20, 2014.
[37] T. Nguyen Canh, “Adaptive block compressive sensing: Towards a complete approach,” 10.13140/RG.2.1.1807.4728, 2015.
[38] W. Wang, W. Yang, and J. Li, “An adaptive sampling method of compressed sensing based on texture feature,” Optik-International Journal for Light and Electron Optics, vol. 127, no. 2, pp. 648–654, 2016.
[39] J. E. Fowler, S. Mun, and E. W. Tramel, “Multiscale block compressed sensing with smoothed projected landweber reconstruction,” in 2011 19th European Signal Processing Conference, Aug 2011, pp. 564–568.
[40] J. Wu, Y. Wang, K. Zhu, and Y. Zhu, “Perceptual sparse representation for compressed sensing of image,” in 2016 Visual Communications and Image Processing (VCIP). IEEE, 2016, pp. 1–4.
[41] S. Foscarini and H. Rauhut, “A mathematical introduction to compressive sensing,” Bull. Am. Math. Soc., vol. 54, pp. 151–165, 2017.
[42] A. L. Pilastri and J. M. R. Tavares, “Reconstruction algorithms in compressive sensing: an overview,” in 11th edition of the Doctoral Symposium in Informatics Engineering (DSIE-16), 2016.
[43] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences, vol. 57, no. 11, pp. 1413–1457, 2004.
[44] J. Romberg, “Imaging via compressive sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 14–20, 2008.
[45] X. Yuan and R. Haimi-Cohen, “Image compression based on compressive sensing: End-to-end compression with jpeg,” arXiv preprint arXiv:1706.01000, 2017.
[46] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Bm3d image denoising with shape-adaptive principal component analysis,” in SPARS’09-Signal Processing with Adaptive Sparse Structured Representations, 2009.
[47] C. Chen, E. W. Tramel, and J. E. Fowler, “Compressed-sensing recovery of images and video using multihypothesis predictions,” in 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), Nov 2011, pp. 1193–1198.
[48] T. Van Chien, K. Q. Dinh, B. Jeon, and M. Burger, “Block compressive sensing of image and video with nonlocal lagrangian multiplier and patch-based sparse representation,” Signal Processing: Image Communication, vol. 54, pp. 93–106, 2017.
[49] “D-amp toolbox: This package contains the code to run the bm3d, bm3dabsca, bls-gsm, and nlm variants of the d-amp, d-vamp, and d-prgmp algorithms,” Apr. 2017. original-date: 2017-02-09T18:27:40Z, available at https://github.com/ricedsp/D-AMP_Toolbox.
Ian J. Wassell received the B.Sc. and B.Eng. degrees from the University of Loughborough, Loughborough, U.K., in 1983, and the Ph.D. degree from the University of Southampton, Southampton, U.K., in 1990. He is a Senior Lecturer in the Computer Laboratory, University of Cambridge, Cambridge, U.K. He has in excess of 15 years experience in the simulation and design of radio communication systems gained via a number of positions in industry and higher education. He has published more than 180 papers concerning wireless communication systems. His research interests include fixed wireless access, sensor networks, cooperative networks, propagation modeling, compressive sensing, and cognitive radio. He is a member of the IET and a Chartered Engineer.