Compressive Video Sampling with Approximate Message Passing Decoding

JIANWEI MA[†♯], GERLIND PLONKA[‡], M. YOUSUFF HUSSAINI [†]

[†] DEPARTMENT OF MATHEMATICS, FLORIDA STATE UNIVERSITY, TALLAHASSEE, FLORIDA 32306-4120, USA

[‡] INSTITUTE FOR NUMERICAL AND APPLIED MATHEMATICS, UNIVERSITY OF GÖTTINGEN, LOTZESTRASSE 16-18, 37083 GÖTTINGEN, GERMANY

[♯] SCHOOL OF AEROSPACE, TSINGHUA UNIVERSITY, BEIJING 100084, CHINA

Abstract

In this paper, we apply compressed sensing to video compression. Compressed sensing (CS) techniques exploit the observation that one needs much fewer random measurements than given by the Shannon-Nyquist sampling theory to recover an object if this object is compressible (i.e., sparse in the spatial domain or in a transform domain). In the CS framework, we can achieve sensing, compression and denoising simultaneously. We propose a fast and simple online encoding by application of pseudo-random downsampling of the two-dimensional fast Fourier transform to video frames. For off-line decoding, we apply a modification of the recently proposed approximate message passing (AMP) algorithm. The AMP method has been derived using the statistical concept of 'state evolution', and it has been shown to considerably accelerate the convergence rate in special CS-decoding applications. We shall prove that the AMP method can be rewritten as a forward-backward splitting algorithm. This new representation enables us to give conditions that ensure convergence of the AMP method and to modify the algorithm in order to achieve higher robustness. The success of reconstruction methods for video decoding also essentially depends on the chosen transform, where sparsity of the video signals is assumed. We propose to incorporate the 3D dual-tree complex wavelet transform that possesses sufficiently good properties of directional selectivity and shift invariance while being computationally less expensive and less redundant than other directional 3D wavelet transforms.

Keywords: Compressed sensing, video online compression, approximate message passing algorithm, high-speed jet flow, 3D directional wavelets, forward-backward splitting algorithm

I. INTRODUCTION

The large amount of raw data acquired in a high-speed camera needs to be compressed immediately before storage and transmission. In this paper, we consider the problem of how one can implement simple and fast online compression of real-time high-speed camera images and how one can effectively recover/decode the original images by an offline algorithm.

Conventional approaches to image/video compression are usually computationally expensive in encoding and they remain relatively simple in decoding processing. Popular compression

methods are based on decorrelation transforms. Applying quantization and entropy encoding to the obtained transform coefficients, one can compress the images. Frequently used tools are JPEG or JPEG2000 with the discrete cosine transform or a wavelet transform as kernel algorithm. In the last few years, directional wavelets such as ridgelets, contourlets, shearlets, and curvelets have been proposed to explore the geometric and structural sparsity of images. Adaptive wavelet transforms e.g. bandlets, tetrolets, and easy path wavelet transform have also been shown to obtain sparse representations of images. However, these sophisticated transforms require high computational burden, they are therefore not suitable for real-time compression of high-speed camera images. For example, the 2D wavelet forward transform of a 1024×1024 image takes 0.54 seconds, and the 2D curvelet forward transform [5] takes 6.8 seconds (test in a laptop with 2.1 GHz Intel Core 2 Duo CPU and 2.0 GB memory). This is not enough efficient for a high-speed camera with 2500 fps (frames per second) considered in our project.

By contrast with the usual approach to acquire large data sets followed by compression, the theory of compressed sensing/compressive sampling (CS) [6], [12] suggests that a compressible unknown signal can be recovered by a small number of random measurements using sparsity-promoting nonlinear recovery algorithms. The CS-based data acquisition depends on the sparsity of data in a certain basis (or frame) rather than on its bandwidth limited by Shannon-Nyquist sampling theorem. Naturally, the image recovery results for a sparsely sampled signal essentially depend on the chosen reconstruction method and the chosen transform domain, where the signal is assumed to be sparse. In this paper, we will consider how one can benefit from the CS theory for the online compression of acquired sequent images by a high-speed camera, instead of improving the camera's imaging instrumentation.

A. Related work

Applications of CS to video processing are still in an infant stage. Wakin et al. [31] first applied the compressive imaging to representation and encoding of videos acquired by a special single-pixel camera model. The results show that a 3D video reconstruction (joint frames) using a 3D wavelet transform is better than the 2D frame-by-frame reconstruction using a 2D wavelet transform. In [11], the problem of signal reconstruction from its quantized signal vector is viewed as a compressive sensing recovery problem where the quantized coefficients are subsampled measurements. Baig et al. [2] considered Gaussian quantization effects on CS videos.Park and Wakin [23] proposed a multiscale CS video processing algorithm, where the CS measurements are taken independently for each frame, and where motion estimation is also applied at the decoding step. The crucial idea of this method is that the motion estimation and CS sparsity-promoting reconstruction is carried out alternately in the multiscale framework. Cossalter et al. [9] also considered the motion estimation in CS video analysis. Stankovic [30] and Prades-Nebot et al. [25] divided each frame into non-overlapping blocks and approximate each block by a linear combination of blocks in previously transmitted frames the CS decoding process. Zheng and Jacobs [36] proposed a video compressive sensing method using spatial domain sparsity, where key frames or reference frames are fully sampled and CS measurements are applied to the difference between the successive frames and the other frames. Xu et al. [33]

incorporated a user attention model with visual rhythm analysis in CS video processing which can automatically determine regions of interest. For every group of successive video frames, the reference frames are fully sampled and build a user attention model to automatically determine regions of interest for non-reference frames that are sampled by CS. Marcia and Willett [22] use CS for increasing the resolution of digital videos. Liu et al. [19] considered maximum frame rate video acquisition by applying a block-based adaptive framework for compressed sensing. Kang et al. [18] applied distributed compressive video sensing, where in the decoding step, compressed video data can be efficiently reconstructed using a modified gradient projection for the reconstruction algorithm.

B. Contribution of this paper

In this paper, we apply the CS technique to compression of videos and of a high-speed jet flow. We use 10% pseudo-random downsampling of 2D FFT as CS measurement matrices for each frame of high-speed camera data. The online compression only involves a 2D FFT that takes 0.08 seconds for the compression of a 1024×1024 image.

In the decoding step, we apply the new approximate message passing (AMP) reconstruction method that has recently been introduced by Donoho, Maleki and Montanari [13]. The AMP algorithm is based on a heuristic formalism in statistics called 'state evolution' [3], [13], [14]. In this paper, we show that the AMP method can be interpreted as a forward-backward splitting algorithm. This new representation enables us to give a proof of convergence of this iterative reconstruction algorithm under suitable restrictions. Further, we derive some modifications of the AMP algorithm in order to improve its robustness.

For application in 3D video compression, the AMP reconstruction method is connected with a sparsity condition based on the dual-tree complex wavelet transform (DTCWT) [28], i.e., we assume that the considered data can be well decorrelated and sparsely represented by the wavelet coefficients of the DTCWT. The DTCWT has a reasonable arithmetical complexity, good directional sensitivity and shift invariance. Therefore this transform is an efficient method to represent edges and surfaces sparsely.

Observe that the online compression and denoising of video data is achieved by a sensing step using the 2D FFT, without any pre-analysis and complicated adaptive transforms of the video source. The method has low complexity and is easily implementable for real-time work.

Extensive numerical experiments for different video data will show the good performance of the AMP method in combination with the 3D DTCWT, where we particularly compare the AMP with the common iterative soft thresholding (IST), and where (besides 3D DTCWT) the 3D tensor product wavelet transform is applied as sparsity constraint.

The paper is organized as follows. In Section 2, we summarize the idea of compressed sensing and introduce the iterative soft thresholding as well as the AMP algorithm for reconstruction, where the latter can be seen as a generalization of IST. Section 3 focusses on the AMP method. We show its representation as a forward-backward splitting algorithm and give several remarks on its heuristic background and on possible modifications that enhance its robustness. In Section

4, we shortly explain, how the 3D dual-tree complex wavelet transform can be incorporated into the AMP reconstruction. Finally, Section 5 is devoted to extensive numerical experiments with video data and a real high-speed jet flow. Section 6 contains some concluding remarks.

II. COMPRESSED SENSING AND AMP RECONSTRUCTION

The theory of compressed sampling studies the problem how the original signal can be recovered from highly incomplete measurements. Let y be a signal that is sampled and let f be a small number of measured samples. The size of y is $N \times 1$ and the size of f is $n \times 1$, where $n \ll N$. Further, let $\tilde{A}$ be the $n \times N$ measurement matrix such that f can be written as

$$f = \tilde{A}y + \epsilon,$$

where ϵ denotes noise. The recovery of the signal $y \in \mathbb{R}^N$ from the observation $f \in \mathbb{R}^n$ is an underdetermined linear system that leads to a seriously ill-posed problem because there are much fewer rows than columns of $\tilde{A}$. However, let us assume that the signal y possesses a sparse representation in a certain basis or a frame, i.e., there is a transform matrix $\Psi \in \mathbb{R}^{M \times N}$ (with $M = N$ for a basis and $M > N$ for a frame), such that Ψy contains only a small set of significant entries (e.g., $K < n$ nonzero coefficients). Further, let the measurement matrix $\tilde{A}$ be not correlated with Ψ. Usually, one assumes that $\tilde{A}$ satisfies the so-called Restricted Isometry Property (RIP) [6]. Then y can be reconstructed with high accuracy from the incomplete measurements f, see [6], [12], [16].

Applying a transform Ψ to y, we obtain in coefficient domain

$$f = \tilde{A}\Psi^{-1}\Psi y + \epsilon = Ax + \epsilon$$

with $x := \Psi y$ and $A = \tilde{A}\Psi^{-1}$, where the Ψ is a forward transform and Ψ^{-1} is its inverse transform. Throughout the paper, we will assume that the transform matrix Ψ represents an orthonormal basis or a Parseval frame, i.e., we have $\Psi^*\Psi = I_N$.

In order to solve the reconstruction problem we consider now the unconstrained optimization problem

$$\min_x \left(\frac{1}{2}\|f - Ax\|_2 + \lambda\|x\|_0 \right),$$

where the first term forces x to be a reasonable (approximate) solution of the linear system $Ax = f$, and the second term forces x to be a sparse vector. Here, the quasi-norm $\|x\|_0$ denotes the number of nonzero components in the vector x, and hence a small value $\|x\|_0$ means sparsity of y in the basis/frame Ψ. However, the above functional is not convex, and its solution is NP-hard. A convex relaxation leads to the usually considered optimization problem

$$\min_x \left(\frac{1}{2}\|f - Ax\|_2 + \lambda\|x\|_1 \right). \tag{1}$$

The regularization parameter λ needs to be tuned suitably. Equation (1) is a so-called synthesis formulation, where one seeks the sparse coefficient vector x in the transform domain, while the minimization problem

$$\min_y \left(\frac{1}{2}\|f - \tilde{A}y\|_2 + \lambda\|\Psi y\|_1 \right)$$

is an analysis formulation, where one directly seeks the image y whose coefficient vector Ψy is sparse.

For simplicity, let us assume here that again $x \in \mathbb{R}^N$ and that $A \in \mathbb{R}^{n \times N}$. (In the case that Ψ is a frame with $M > N$, the dimension N can be simply replaced by M.)

Iterative soft shrinkage/thresholding (IST) is one of the most popular methods to solve the ℓ_1-norm CS decoding problem (1) because of its simplicity and efficient performance. Specifically, the iteration step of IST is given by

$$
\begin{align}
x_{k+1} &= \mathcal{T}_{\lambda t_k}(x_k + t_k A^* z_k), & (2) \\
z_{k+1} &= f - A x_{k+1} & (3)
\end{align}
$$

with $x_0 = 0$, $z_0 = f$, and where t_k is an appropriate step size at each iteration step. Here, the operator $\mathcal{T}_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ is the soft shrinkage operator, defined for each component by

$$
\mathcal{T}_\alpha(x)_i := (|x_i| - \alpha)_+ \operatorname{sgn}(x_i).
$$

In the last two years, many convergence accelerating improvements of the IST have been proposed. The main objective is to devise a faster method, keeping the simplicity of the IST method and significantly improving the global convergence rate. Here, we just mention a few ideas including Beck and Teboulle's first-order fast IST method (FISTA) [4], Daubechies et al's accelerated projected gradient method [10], Wright et al's SpaRSA (sparse reconstruction via separable approximation) [32], and Bregman iteration methods [15], [24], [35], [37].

In this paper, we focus on the AMP algorithm proposed in [3], [13], [20] that has been proved to be extremely effective in reconstructing sparse signals from a small number of incoherent linear measurements. The AMP method offers the low complexity of IST and the reconstruction power of basis pursuit.

Compared to the conventional IST (2)-(3), the AMP only adds one additional term, see [21]. The AMP algorithm reads

$$
\begin{align}
x_{k+1} &= \mathcal{T}_{t_k}(x_k + A^* z_k), & (4) \\
z_{k+1} &= f - A x_{k+1} + \frac{N}{n} \langle \mathcal{T}'_{t_k}(x_k + A^* z_k) \rangle z_k, & (5)
\end{align}
$$

with initial conditions $x_0 = 0$, $z_0 = f$. Here, for $x = (x_1, \ldots, x_N)^T$ we write $\langle x \rangle := \sum_{i=1}^N x_i / N$, and $\mathcal{T}'_{t_k}$ is the first derivative of the thresholding operator (applied separately to each component).

As in [13], [3], we assume that the measurement matrix A satisfies $AA^* = I_n$ (resp. $\tilde{A}\tilde{A}^* = I_n$).

Remark.

Message passing methods are used in statistics/machine learning and for computing of inferences in graphical models and graph-based error correcting codes. Within the last years, these belief propagation methods and their relation to compressed sensing methods and linear programming have been extensively studied, see e.g. [1], [17], [20], [26], [27], [34]. The basic variables of these special iterative algorithms are messages being associated to directed edges in the graph that encode the structure of the statistical model.

III. AMP AS A FORWARD-BACKWARD SPLITTING ALGORITHM

In this section, we show that the AMP iteration can be understood as a special forward-backward splitting algorithm. This observation enables us to present sufficient conditions for the convergence of the AMP algorithm.

In convex analysis, iterative algorithms for solving optimization problems of the form

$$\min\left\{F(x) = f_1(x) + f_2(x) : x \in \mathcal{H}\right\} \tag{6}$$

have been studied extensively. Here, $\mathcal{H}$ denotes a Hilbert space, and f_1, f_2 are functionals from $\mathcal{H}$ to $\mathbb{R} \cup \{\infty\}$ which are lower semicontinuous, convex, and not identically equal to $+\infty$. If f_2 is moreover differentiable in $\mathcal{H}$ with a Lipschitz continuous gradient for some $\beta > 0$, then the following theorem, based on proximity operators, has been proved in [8].

Theorem 3.1: *Suppose that (6) possesses at least one solution. Let $\{\beta_k\}_{k\in\mathbb{N}}$ be a sequence in $(0,\infty)$ such that $\inf_{k\in\mathbb{N}} \beta_k > 0$ and $\sup_{k\in\mathbb{N}} \beta_k < 2/\beta$, let $(\mu_k)_{k\in\mathbb{N}}$ be a sequence in $(0,1]$ such that $\inf_{k\in\mathbb{N}} \mu_k > 0$, and let a_n and b_n be sequences in $\mathcal{H}$ such that $\sum_{k\in\mathbb{N}} \|a_k\| < +\infty$ and $\sum_{k\in\mathbb{N}} \|b_k\| < +\infty$. Fix $x_0 \in \mathcal{H}$, and for every $k \in \mathbb{N}$ set*

$$x_{k+1} = x_k + \mu_k\left(\operatorname{prox}_{\beta_k f_1}\left(x_k - \beta_k(\nabla f_2(x_k) + b_k)\right) + a_k - x_k\right). \tag{7}$$

Then the sequence $(x_k)_{k\in\mathbb{N}}$ converges weakly to the solution of (6).

We want to apply the above theorem to solve (1) and put $\mathcal{H} = \mathbb{R}^N$,

$$f_1(x) := \lambda\|x\|_1, \qquad f_2(x) := \frac{1}{2}\|Ax - f\|_2^2.$$

Then the gradient

$$\nabla f_2(x) = A^*(Ax - f)$$

is Lipschitz continuous of order $\beta = \|A^*A\|_2$, where we may use the spectral matrix norm. We assume that A consists of randomly sampled rows of a unitary matrix (as the Fourier matrix), such that we have $AA^* = I_n$ and $\beta = 1$. Further, the proximity operator

$$\operatorname{prox}_{\beta_k f_1}(x) := \arg\min_{y\in\mathbb{R}^N}\left\{\beta_k f_1(y) + \frac{1}{2}\|x - y\|_2^2\right\} = \arg\min_{y\in\mathbb{R}^N}\left\{\beta_k\lambda\|y\|_1 + \frac{1}{2}\|x - y\|_2^2\right\}$$

is given by

$$\operatorname{prox}_{\beta_k f_1}(x) = \mathcal{T}_{\beta_k\lambda}(x),$$

where $\mathcal{T}_{\beta_k\lambda}$ denotes the soft shrinkage operator as before. Hence, the iteration (7) reads

$$x_{k+1} = x_k + \mu_k\left(\mathcal{T}_{\beta_k\lambda}(x_k - \beta_k(A^*(Ax_k - f) + b_k)) + a_k - x_k\right).$$

In the special case $\mu_k = 1$, $a_k = 0$ and $t_k = \beta_k\lambda$ for all $k \in \mathbb{N}$, we obtain the special forward-backward splitting iteration

$$x_{k+1} = \mathcal{T}_{t_k}\left(x_k - \beta_k(A^*(Ax_k - f) + b_k)\right). \tag{8}$$

Observe that the IST algorithm in (2)-(3) is of the above form with $b_k = 0$. Indeed, we have a simple heuristic interpretation of IST. The addition of the term $-\beta_k\nabla f_2(x_k) = -\beta_k(A^*(Ax_k -$

f) forces a decrease of the first term of the functional in (1), while the application of the soft shrinkage operator $\mathcal{T}_{t_k}$ forces the reduction of the 1-norm of $x_k - \beta_k(A^*(Ax_k - f)$. We now obtain

Theorem 3.2: *The AMP iteration (4)- (5) with the threshold recursion*

$$t_{k+1} = \lambda + \frac{Nt_k}{n}\langle \mathcal{T}'_{t_k}(x_k + A^*z_k)\rangle = \lambda + \frac{t_k}{n}\|x_{k+1}\|_0$$

can be represented as a forward-backward splitting algorithm of the form

$$x_{k+1} = \mathcal{T}_{\beta_k\lambda}\left(x_k - \beta_k(A^*(Ax_k - f) + b_k)\right) \tag{9}$$

with starting values $x_0 = 0$, $b_0 := 0$ and $\beta_0 := 1$, where the sequences $\{\beta_k\}_{k\geq 1}$ and $\{b_k\}_{k\geq 1}$ are given by the iterations

$$\beta_{k+1} := 1 + \frac{\beta_k}{n}\|x_{k+1}\|_0 \tag{10}$$

$$b_{k+1} := \frac{\beta_{k+1} - 1}{\beta_{k+1}}\left(b_k - A^*A(x_{k+1} - x_k)\right) \tag{11}$$

for $k = 0, 1, 2, \ldots$.
The AMP sequence $\{x_k\}_{k\geq 0}$ converges if $\sup_{k\in\mathbb{N}}\beta_k < 2$ and $\sum_{k\in\mathbb{N}}\|b_k\| < \infty$.

Proof.

Considering the iteration in (4)-(5) and applying the definition of the soft shrinkage operator, we define

$$\gamma_k := \frac{N}{n}\langle \mathcal{T}'_{t_{k-1}}(x_{k-1} + A^*z_{k-1})\rangle = \frac{1}{n}\|x_k\|_0,$$

where $\|x_k\|_0$ denotes the number of nonzero components in the current iteration vector x_k. Then the AMP method in (4)-(5) simply reads

$$x_{k+1} = \mathcal{T}_{t_k}(x_k + A^*z_k),$$
$$z_{k+1} = f - Ax_{k+1} + \gamma_{k+1}z_k,$$

or equivalently,

$$x_{k+1} = \mathcal{T}_{t_k}(x_k + A^*(f - Ax_k) + A^*\gamma_k z_{k-1}), \tag{12}$$
$$z_{k+1} = f - Ax_{k+1} + \gamma_{k+1}z_k. \tag{13}$$

The iteration (12) is equivalent with (8) if

$$\beta_k(A^*(f - Ax_k) - b_k) = A^*(f - Ax_k) + A^*\gamma_k z_{k-1},$$

i.e.,

$$\beta_k b_k = (1 - \beta_k)A^*(Ax_k - f) - A^*\gamma_k z_{k-1}.$$

Hence, using (13), we can derive the following recursion formula for b_k,

$$\begin{aligned}
\beta_k b_k &= (1 - \beta_k)A^*(Ax_k - f) + \gamma_k A^*(Ax_{k-1} - f - \gamma_{k-1}z_{k-2}) \\
&= (1 - \beta_k)A^*(Ax_k - f) + \gamma_k A^*(Ax_{k-1} - f) \\
&\quad + \gamma_k\left(\beta_{k-1}b_{k-1} - (1 - \beta_{k-1})A^*(Ax_{k-1} - f)\right).
\end{aligned}$$

We apply the threshold recursion that has been already proposed in [14],

$$t_{k+1} = \beta_{k+1}\lambda = \lambda + \frac{Nt_k}{n}\langle\mathcal{T}'_{t_k}(x_k + A^*z_k)\rangle = \lambda + t_k\gamma_{k+1} = \lambda(1 + \beta_k\gamma_{k+1}),$$

that yields $\gamma_{k+1} = \frac{t_{k+1}-\lambda}{t_k} = \frac{\beta_{k+1}-1}{\beta_k}$, and the recursion for b_k simplifies to

$$\beta_k b_k = (\beta_k - 1)(b_{k-1} - A^*A(x_k - x_{k-1})).$$

For β_k we have the recursion $\beta_{k+1} = 1 + \beta_k\gamma_{k+1}$. Finally, the conditions for convergence follow directly from Theorem 3.1. $\qquad\square$

Remarks.

1. We remember that for $f_2(x) = \frac{1}{2}\|Ax - f\|^2$ we have $\nabla f_2(x) = A^*(Ax - f)$. Hence, the "correction vector" b_k in the above AMP iteration in (9) is nothing but a suitable average of the gradients of f_2 at the iteration vectors x_k. Indeed we find $b_1 = \frac{\beta_1-1}{\beta_1}(\nabla f_2(x_0) - \nabla f_2(x_1))$, and a simple induction argument yields

$$b_k = \sum_{\ell=0}^{k} \alpha_\ell \nabla f_2(x_\ell)$$

with

$$\alpha_k = \frac{1 - \beta_k}{\beta_k}, \quad \alpha_0 = \prod_{\nu=1}^{k}\frac{\beta_\nu - 1}{\beta_\nu}, \quad \alpha_\ell = \frac{1}{\beta_\ell}\prod_{\substack{\nu=1\\\nu\neq\ell}}^{k}\frac{\beta_\nu - 1}{\beta_\nu} \quad \text{for} \quad 1 \le \ell \le k - 1.$$

In particular, we observe that $\sum_{\ell=0}^{k}\alpha_\ell = 0$. Hence, the AMP method can heuristically be interpreted as follows. Instead of applying only the gradient of the current iteration vector x_k a certain linear combination of gradients of the preceding iteration vectors is taken in order to ensure faster convergence.

2. The above considerations about the sequence $\{b_k\}_{k\in\mathbb{N}}$ particularly imply that $A^*Ab_k = b_k$ for $k \in \mathbb{N}$. Hence, with $f_k := f - Ab_k$, the iteration in (9) can be rewritten as

$$x_{k+1} = \mathcal{T}_{\beta_k\lambda}\left(x_k - \beta_k(A^*(Ax_k - f_k))\right).$$

The recursion for b_k in Theorem 3.2 simply leads to the iteration formula for f_k,

$$f_{k+1} = \frac{1}{\beta_{k+1}}f + \frac{\beta_{k+1} - 1}{\beta_{k+1}}\left(f_k + A(x_{k+1} - x_k)\right).$$

This representation of the AMP-method shows a certain connection to iteration algorithms based on Bregman distances, see e.g. [15], [24], [35], where corrected vectors f_k are used in the IST iteration.

3. The threshold recursion in Theorem 3.2 has been proposed by Donoho et al. [14]. In order to satisfy the convergence condition $\sup_{k\in\mathbb{N}}\beta_k < 2$, we may rather put the more general threshold rule according to

$$t_{k+1} = \beta_{k+1}\lambda = \lambda + \frac{C_k t_k}{n}\|x_{k+1}\|_0 = \lambda(1 + \frac{C_k\beta_k}{n}\|x_{k+1}\|_0)$$

with some constants $C_k > 0$, see Section 5. In this case the iterations (10) and (11) for $\{\beta_k\}_{k \in \mathbb{N}}$ and $\{b_k\}_{k \in \mathbb{N}}$ have to be modified and read

$$
\begin{aligned}
\beta_{k+1} &:= 1 + \frac{C_k \beta_k}{n} \|x_{k+1}\|_0 \\
b_{k+1} &:= \frac{\beta_{k+1} - 1}{\beta_{k+1}} \left(C_k b_k - A^*(Ax_{k+1} - f) + C_k A^*(Ax_k - f) \right)
\end{aligned}
$$

for $k = 0, 1, 2, \ldots$. Furthermore, it can be advantageous to allow that $\lambda = \lambda_k$ decays to zero for $k \to \infty$, and

$$
t_{k+1} = \beta_{k+1} \lambda_{k+1} = \lambda_k + \frac{N t_k}{n} \langle \mathcal{T}'_{t_k}(x_k + A^* z_k) \rangle = \lambda_k + t_k \gamma_{k+1} = \lambda_k (1 + \beta_k \gamma_{k+1}),
$$

see [14]. This yields the recursion $\beta_{k+1} = \frac{\lambda_k}{\lambda_{k+1}} (1 + \beta_k \gamma_{k+1})$ and according slight changes in the iteration procedure.

4. The numerical implementation of the above AMP method is not numerically stable, therefore, we replace the term $\|x_k\|_0$ by the number of components in x_k with modulus greater than a fixed small value ϵ in our numerical experiments.

5. The above observation that the AMP algorithm can be seen as a special forward-backward splitting method, may also help to understand the powerfulness of this general iteration scheme in [8]. In particular, it shows that the sequences $\{a_k\}_{k \in \mathbb{N}}$ and $\{b_k\}_{k \in \mathbb{N}}$ in (7) may not be just understood as tolerances for inexact evaluations. Instead, a suitable choice of these sequences may yield essential convergence acceleration.

6. We also explored a possible connection between the AMP method and the Bregman operator splitting (BOS) algorithm in [37], where a single forward-backward operator splitting step is used to solve the subproblems of the Bregman iteration. Although the formulations may look similarly at first glance, there are no close connection between them. The main reason is that there is almost no freedom in the BOS algorithm for changing constants depending on the level. The two constants δ and μ in [37] are fixed and do not depend on the level k of the iteration. Even if we use relaxation parameters (e.g. δ_k instead of δ), there is no simple connection. This is due to the fixed determination of f^k without any freedom to put a constant, while in the AMP method we have a constant in the second iteration that depends on $\|x^{k-1}\|_0$.

IV. AMP FOR COMPRESSED VIDEO SENSING

Now, we want to apply the AMP algorithm with sparsity assumption in the 3D wavelet domain for data reconstruction. Let $\widehat{A}$ denote a block matrix of the form

$$
\widehat{A} = I_d \otimes \tilde{A} := \begin{pmatrix} \tilde{A} & & & \\ & \tilde{A} & & \\ & & \ddots & \\ & & & \tilde{A} \end{pmatrix} \in \mathbb{C}^{dn \times dN},
$$

where $\tilde{A} \in \mathbb{R}^{n \times N}$ is the 2D-measurement matrix in each slice of the considered 3D domain. In our experiments, we shall use 2D partial FFT measurements, i.e., $\tilde{A}$ is obtained by a random

choice of about 10% of the pseudo-random sampling of the unitary Fourier matrix (see Figure 1 (a)). One can also use a structured random matrix, e.g., a circular random matrix (see Figure 1 (b)), which can easily be stored in memory, and where the matrix vector multiplication $\tilde{A}y$ can be efficiently calculated by the FFT algorithm. Generally, the pseudo-random Fourier matrix achieves better performance than a circular random matrix in the case of low-rate measurement.

| (a) | (b) |
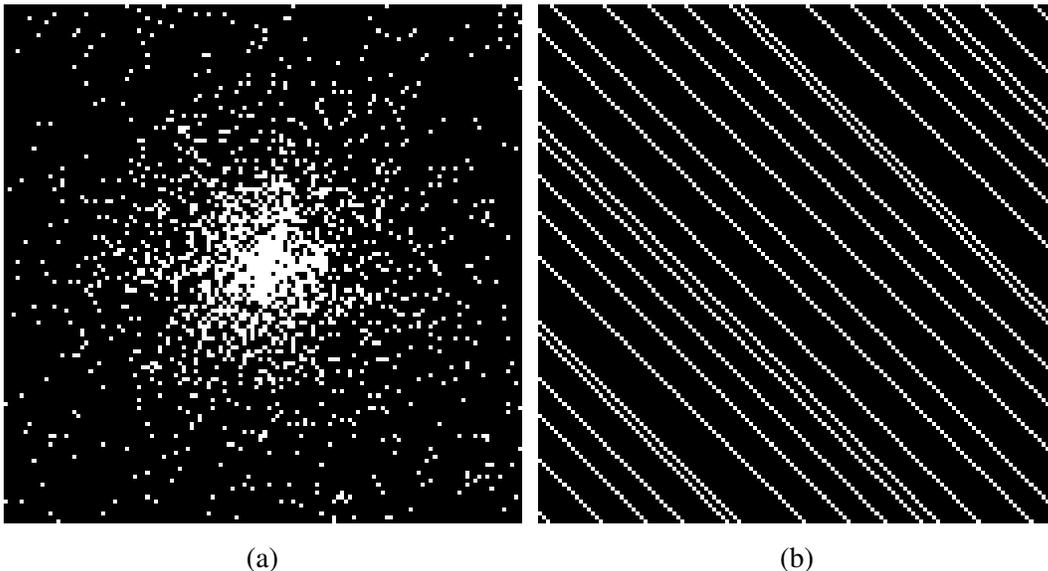
Fig. 1. CS measurement matrices with 10% sampling. (a) Pseudo-random sampling of the Fourier matrix. (b) Circular random matrix (the first row is produced by a random vector). The white points denote sampling.

For decoding, we apply the AMP-method together with a 3D-wavelet transform exploiting the sparsity of video data and of the high-speed jet flow in the wavelet domain. Particularly, we implement the 3D discrete wavelet transform (DWT) with Daubechies filters and the three-dimensional dual-tree complex wavelet transform (3D DTCWT).

Let Ψ be the transform matrix of the DWT resp. the DTCWT, and Ψ^{-1} the inverse transform. Then the AMP iteration reads in the transform domain

$$x_{k+1} \;=\; \mathcal{T}_{t_k}(x_k + \Psi\widehat{A}^* z_k), \tag{14}$$

$$z_{k+1} \;=\; f - \widehat{A}\Psi^{-1} x_{k+1} + \widehat{c}_{k+1} z_k, \tag{15}$$

where

$$\widehat{c}_{k+1} = \frac{N}{n}\langle \mathcal{T}'_{t_k}(\Psi\widehat{A}^* z_k + x_k)\rangle = \frac{1}{n}\|x_{k+1}\|_0,$$

and with initial data $x_0 = z_0 = 0$. The techniques described in Remark 3 and 4 can be also applied to the DTCWT-domain AMP.

Originally, the DTCWT has been designed as a two-dimensional transform enhancing the usual tensor-product DWT regarding its (insufficient) shift invariance and its directional selectivity. For the DTCWT, we refer especially to the survey by Selesnick et al. [28]. The 3D DTCWT has been already successfully applied for 3D video data, see e.g. [29]. The principle by which

the 2D DTCWT resolves the problem of the mixing of orientations (as it happens for the 2D tensor-product DWT) can also be used to resolve the mixing of orientations in the 3D case. The oriented 3D DTCWT is expansive by 4, while the application of the corresponding filter banks remains to be simple and efficient. It is important to notice that the checkerboard artifacts of the conventional separable 3D wavelet transform become even more serious in 3 dimensions. The advantages of the 3D DTCWT compared to the separable 3D wavelet transform can be observed in our numerical examples. Furthermore, the DCTWT is more efficient than 3D curvelets or other generalizations of directional wavelet frames to three dimensions. The Matlab software for the 3D DTCWT can be found on `http://taco.poly.edu/WaveletSoftware/`.

V. NUMERICAL EXPERIMENTS

We have tested the methods for an animated Cartoon video sequence, the standard Foreman video sequence and a real high-speed jet flow with a frame ratio of 2500 frames per second. We have used 2D partial FFT frame-by-frame measurements in our experiments, where the size of each frame is 128×128 and we applied 30% measurements of the videos and 10% measurements of jet flow.

The AMP method is now used for a 3D joint-frame decoding for a 64-frame sequent flow. In our numerical experiments we compare the performances of the following decoding methods:

1) Iterative soft thresholding with sparsity assumption in spatial domain (with $\Psi = I_N$ and $\tilde{A} = A$) (IST),

2) approximate message passing method with sparsity assumption in spatial domain (AMP),

3) iterative soft thresholding with sparsity assumption in wavelet domain (Ψ denotes the transform matrix of the 3D discrete wavelet transform with Daubechies (D4) filters) (IST-DWT),

4) iterative soft thresholding with sparsity assumption in the complex wavelet domain (Ψ denotes the transform matrix of the 3D dual-tree complex wavelet transform) (IST-DTCWT),

5) approximate message passing method with sparsity assumption in the complex wavelet domain (AMP-DTCWT).

The SNR (signal-to-noise ratio) is computed by $20\log_{10}\frac{\|f\|_2}{\|f-\hat{f}\|_2}$ where f is the original image and $\hat{f}$ is the reconstruction. We use the decreasing sparsity-measurement tradeoff parameter $C_k = C_0(1 - k/N_{iter})$ (see Remark 3 in Section 3) and the thresholding value $\lambda_k = \lambda_0(1 - k/N_{iter})$, where k denotes the iteration index and N_{iter} is the total number of iterations. In our experiments, we use $N_{iter} = 20$. Generally, a larger N_{iter} results in a better recovery at the cost of more computational time. The sparsity-measurement tradeoff parameter C_0 and threshold λ_0 of course depend on the data. We take the values $\lambda_0 = 0.6$ and $C_0 = 1.2$ for the AMP-DTCWT method and $C_0 = 0.8$ for the spatial-domain AMP method. Our experiments show, that the choice $C_k = 1$ proposed by Donoho et al. [14] leads not to a robust algorithm for the general cases. As already seen in Theorem 3.2, the condition $\frac{C_k\|x_k\|_0}{n} < 1$ is crucial for convergence of the AMP algorithm, and this can be achieved by a careful choice of the threshold values during the iterations.

For (almost) exact recovery by the reconstruction algorithm, the number of measurements n should be 2-10 times as high as the sparsity $||y||_0$ in spatial domain or as $||x||_0 = ||\Psi y||_0$ in coefficient domain. However, in our work, we try to use less measurements in order to reduce the cost of online compression since we do not need 100% exact reconstruction. In this case we usually have $||x_k||_0/n > 1$, and the algorithm originally proposed in [14] will not converge without a suitable choice of λ_k and C_k.

In Figure 2 (a), we display the 32nd frame (the middle frame) of the cartoon video. Figures 2 (b)-(f) show the decoding results by IST, AMP, IST-DWT, IST-DTCWT and AMP-DTCWT, respectively. Furthermore, Figure 3 shows the comparison of SNR values for the five algorithms during the iterative decoding procedure. Generally, we observe that the algorithms with sparsity constraints in the wavelet domain/complex wavelet domain are favorable and yield reasonable results already after a small number of iterations. In our case, the DTCWT is a better choice than the 3D wavelet transform because the reconstruction of structural features is most important for the cartoon video. The AMP-DTCWT based method obviously enhances the convergence performance. Here we only need 7 iterations to achieve a good reconstruction with a high SNR value.

Figures 4 and 5 show the decoding comparisons for the 32nd frame of the high-speed jet flow. Again, the AMP techniques gain the performance of reconstruction. In contrast to the cartoon video, the original measured jet flow (as shown in Fig. 4 (a)) includes noises. So the SNR value may be not the best quantity to assess the reconstruction. At least regarding the visual quality, the AMP-DTCWT displays best results where edges are reconstructed and noises are removed efficiently. In fact, we only need the edges and structures from the off-line decompressed image in order to observe the evolution of the turbulent flow.

Figures 6 and 7 show the decoding comparisons for the 32nd frame of the Foreman video. In this case, the algorithms with wavelet/complex wavelet sparsity constraints gain the performance very well. Obviously, the AMP technique further improves the results. The AMP-DTCWT achieves the best results in terms of SNR values and visual quality.

Finally, we also give the detailed comparisons of SNR values for other frames after reconstruction with 20 iterations. In Table I, we provide the information for middle frames from 24th to 39th frame for the Cartoon video, the Jet flow and the Foreman video. The last row lists the average SNR values of all 64 frames for the different methods. As it can be seen from Figures 3, 5 and 7, the difference in performance is even much more in favor of the AMP-DTWCT if we had compared the results already after 10 iterations.

However, in current understanding, the direct CS methods (without additional encoder/decoder for spatiotemporal correlation of video signals) can not simply achieve low-bit compression rates for images or videos as produced by JPEG2000 or by H.264. In Figure 8, we show a comparison to traditional compression methods based on the discrete curvelet transform [5] and on the discrete wavelet transform with hard thresholding, respectively. As usual, we decompose the images by the curvelet and the wavelet transform, respectively, and then perform the reconstruction using only the 1628 most significant wavelet/curvelet coefficients (1 % of data).

Both methods obtain higher SNR values than CS coding. This is of course reasonable, because both the curvelet and the wavelet method choose the best transform coefficients 'adaptively' while CS encoding chooses the Fourier coefficients randomly. But for the traditional methods, one has also to store and to transmit the location information of the curvelet or wavelet coefficients for each frame, while for CS coding one only keeps the location of random sampling one time and does not need to transmit it. Comparing the encoding effort, one can indeed apply the CS technique for low-complexity and energy-efficient online encoding of videos. In this case of 128×128 images, the online computational time for the CS encoding, wavelet transform and curvelet transform is 0.0016 seconds, 0.032 seconds, and 0.281 seconds, respectively. Thus, CS-methods are especially attractive for data compression of high-speed cameras, where the encoding is required to be simple and fast. Another important potential advantage of CS encoding is that it can be implemented by hardware for simultaneous data acquisition and compression.

VI. Conclusions

In this paper, the compressed sensing (CS) technique is applied to low-complexity and energy-efficient online compression of video sequences and high-speed jet flows. The CS encoding is promising since it can be implemented easily by hardware, and it is very efficient. This is especially attractive for data compression by high-speed cameras, where the encoding is required to be simple and fast. The CS-technique shifts the computational cost of high-fidelity video decompression to offline processing. We applied a recently proposed AMP method together with a 3D dual-tree complex wavelet transform for our decoding. The method can efficiently recover the original video signals using very few partial Fourier measurements. The spatiotemporal correlations of the video signal, being ignored by the simple encoding, are exploited well by the proposed decoding method. We have shown that the AMP method can be seen as a general forward-backward splitting algorithm. In particular, we can derive conditions for the convergence of AMP that lead to improvements of its performance by suitable parameter choice. Further, significant connections of the AMP with Bregman iterations are also addressed.

To further improve the current results, one can apply other measurement matrices (e.g., structured random matrices) to achieve faster computation in the encoding phase, and incorporate inter-frame techniques from conventional video compression to explore the temporal redundancy in the decoding phase. How to apply optimal thresholding parameters is another next work.

VII. Acknowledgement

REFERENCES

[1] S. Arora, X. Daskalakis, D. Steurer, Message passing algorithms and improved LP decoding, Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC'09), ACM, New York, 2009.

[2] Y. Baig, E. Lai, J. Lewis, Quantization effects on compressed sensing video, IEEE 17th Int. Conf. on Telecommunications (ICT), 2010.

[3] M. Bayati, A. Montanari, The dynamics of message passing on dense graphs, with applications to compressed sensing, IEEE Trans. Inform. Theory, **57**(2), 764-785 (2011).

[4] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci., **2**, 183-202 (2009).

[5] E. Candès, L. Demanet, D. Donoho, L. Ying, Fast discrete curvelet transforms, Multiscale Model. Simul., **5** (3), 861-889 (2006).

[6] E. Candès, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inform. Theory, **52** (2), 489-590 (2006).

[7] V. Chandar, D. Shah, G.W. Wornell, A simple message-passing algorithm for compressed sensing, ISIT 2010, Austin, Texas, June 13-18, 2010.

[8] P. Combettes, V. Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Model. Simul., **4**, 1168-1200 (2005).

[9] M. Cossalter, G. Valenzise, M. Tagliasacchi, S. Tubaro, Joint compressive video coding and analysis, IEEE Tran. Multimedia, **12** (3), 168-183 (2010).

[10] I. Daubechies, M. Defrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math., **57** (11), 1413-1457 (2004).

[11] T. Do. X. Lu, J. Sole, Compressive sensing with adaptive pixel domain reconstruction for block-based video coding, IEEE 17th Int. Conf. Image Process., Hongkong, Sep. 26-29, 2010.

[12] D. Donoho, Compressed sensing, IEEE Trans. Inform. Theory, **52** (4), 1289-1306 (2006).

[13] D. Donoho, A. Maleki, A. Montanari, Message passing algorithms for compressed sensing, Proc. National Academy of Science, **10** (45), 18914-18919 (2009).

[14] D. Donoho, A. Maleki, A. Montanari, Message passing algorithms for compressed sensing: II. Analysis and Validation, Proc. IEEE ITW, Cairo, Egypt, Jan. 2010.

[15] T. Goldstein, S. Osher, The split Bregman method for L_1 regularized problems, SIAM J. Imaging Sci., **2** (2), 323-343 (2009).

[16] V. Goyal, A. Fletcher, S. Rangan, Compressive sampling and lossy compression, IEEE Signal Process. Mag., **25** (2), 48-56 (2008).

[17] S. Jafarpour, W. Xu, B. Hassibi, R. Calderbank, Efficient and robust compressed sensing using optimized expander graphs, IEEE Trans. Inform. Theory, **55** (9), 4299-4308 (2009).

[18] L. Kang, C. Lu, Distributed compressive video sensing, 43rd Annual Conference on Information Sciences and Systems, 2009.

[19] Z. Liu, A. Elezzabi, H. Zhao, Maximum frame rate video acquisition using adaptive compressed sensing, IEEE Trans. Circuits Sys. Video Tech., to appear, 2011.

[20] A. Maleki, D. Donoho, Optimally tuned iterative thresholding algorithm for compressed sensing, IEEE J. Selected Areas in Signal Process., **4** (2), 330-341 (2009).

[21] A. Maleki, A. Montanari, Analysis of approximate massage passing algorithm, 44th Annual Conf. Inform. Sci. Systems, Princeton, New Jersey, pp. 1-7, March 17-19, 2010.

[22] R. Marcia, R. Willett, Compressive coded aperture video reconstruction, 16th European Signal Process. Conf. (EUSIPCO 2008), Lausanne, Switzerland, August 2008.

[23] J. Park, M. Wakin, A multiscale framework for compressive sensing of video, 2009 Picture Coding Symposium, Chicago, Illinois, May 2009.

[24] G. Plonka, J. Ma, Curvelet-wavelet regularized split Bregman iteration for compressed sensing, Int. J. Wavelet, Multires. Inform. Process., **9** (1), 79-110 (2011).

[25] J. Prades-Nebot, Y. Ma, T. Huang, Distributed video coding using compressive sampling, 2009 Picture Coding Symposium, Chicago, Illinois, May 2009.

[26] S. Som, L. Potter, P. Schniter, On approximate message passing for reconstruction of non-uniformly sparse signals, Proc. National Aerospace and Electronics Conf., Dayton, OH, July 2010.

[27] S. Sarvotham, D. Baron, R. Baraniuk, Bayesian compressive sensing via belief propagation, IEEE Trans. Signal Process., **58** (1), 269-280 (2010).

[28] I. Selesnick, R. Baraniuk, N. Kingsbury, The dual-tree complex wavelet transform, IEEE Signal Process. Mag., **22** (6), 123-151 (2005).

[29] I. Selesnick, K. Li, Video denoising using 2D and 3D dual-tree complex wavelet transforms, in: Wavelet Appl. Signal Image Proc. X, Proc. SPIE, vol. 5207, pp. 607-618, 2003.

[30] V. Stankovic, L. Stankovic, S. Cheng, Compressive video sampling, 16th European Signal Process. Conf. Proc. (EUSIPCO2008), Lausanne, Switzerland, August 2008.

[31] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, R. Baraniuk, Compressive imaging for video representation and coding, 2006 Picture Coding Symposium, Beijing, China, 2006.

[32] S. Wright, R. Nowak, M. Figueiredo, Sparse reconstruction by separable approximation, IEEE Trans. Signal Process., **57** (7), 2479-2493 (2009).

[33] J. Xu, J. Ma, D. Zhang, Y. Zhang, S. Lin, Compressive video sensing based on user attention model, 2011 Picture Coding Symposium, Tokyo, Japan, Dec. 8-10, 2010.

[34] W. Xu, B. Hassibi, Efficient compressive sensing with deterministic guarantees using expander graphs, ITW 2007, Lake Tahoe, California, September 2-6, 2007.

[35] W. Yin, S. Osher, D. Goldfarm, J. Darbon, Bregman iterative algorithms for ℓ_1 minimization with applications to compressed sensing, SIAM J. Imaging Sci., **1** (1), 143-168 (2008).

[36] J. Zheng, E. Jacobs, Video compressive sensing using spatial domain sparsity, Optical Engineering, **48** (8), 087006 (2009).

[37] X. Zhang, M. Burger, X. Bresson, S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, SIAM J. Imaging Sci., **3** (3), 253-276 (2010).

(a)

(b) SNR = 18.0879 dB

(c) SNR= 18.0234 dB

(d) SNR=18.5335 dB

(e) SNR= 18.9709 dB

(f) SNR=19.3739 dB

Fig. 2. Cartoon video: the comparison of the 32nd frame with SNR values. (a) Original data. (b) IST reconstruction. (c) AMP reconstruction. (d) IST-DWT reconstruction. (e) IST-DTCWT reconstruction. (f) AMP-DTCWT reconstruction.
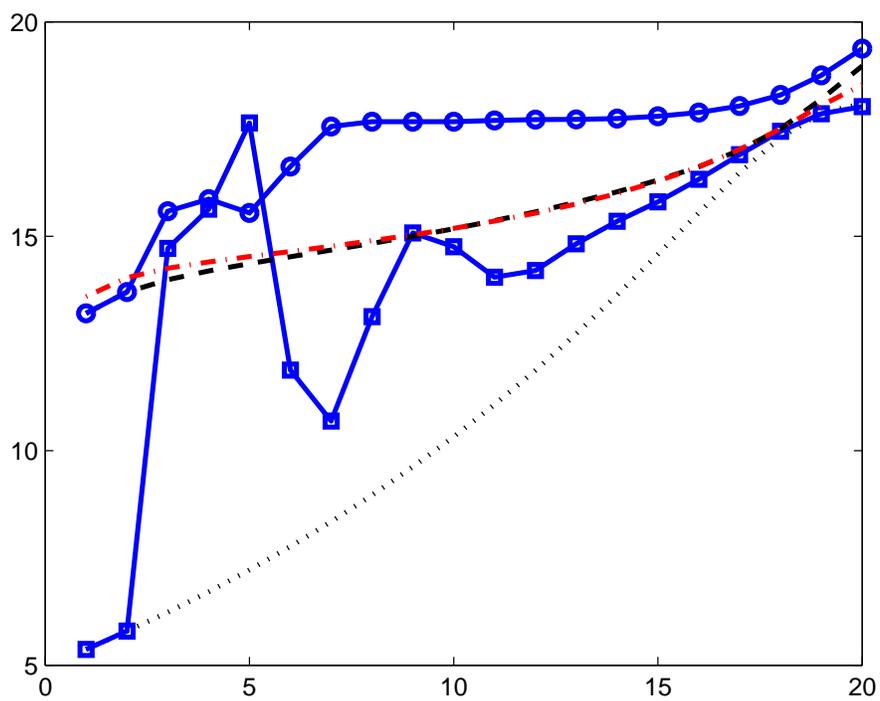
Fig. 3. Cartoon video: comparison of the SNR change as the number of iteration increases, by different methods: IST (dotted line), AMP (solid line with squares), IST-DWT (dot-dashed line), IST-DTCWT (dashed line), AMP-DTCWT (solid line with circles).

(a)

(b) SNR = 20.2282 dB

(c) SNR= 20.5407 dB

(d) SNR=19.5246 dB

(e) SNR= 19.9223 dB

(f) SNR=20.0399 dB
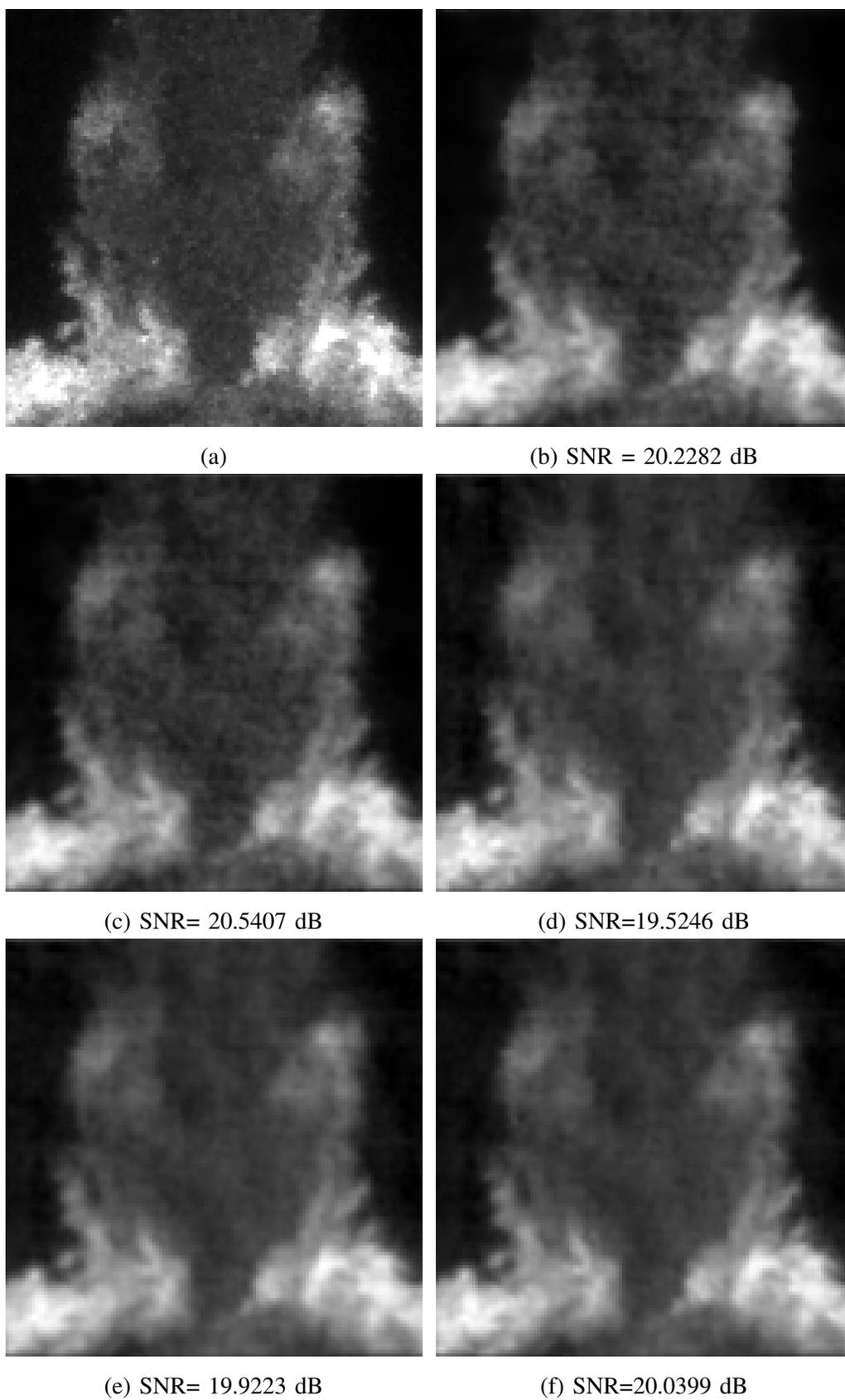
Fig. 4.   Jet flow: the comparison of the 32nd frame with SNR values. (a) Original data. (b) IST reconstruction. (c) AMP reconstruction. (d) IST-DWT reconstruction. (e) IST-DTCWT reconstruction. (f) AMP-DTCWT reconstruction.
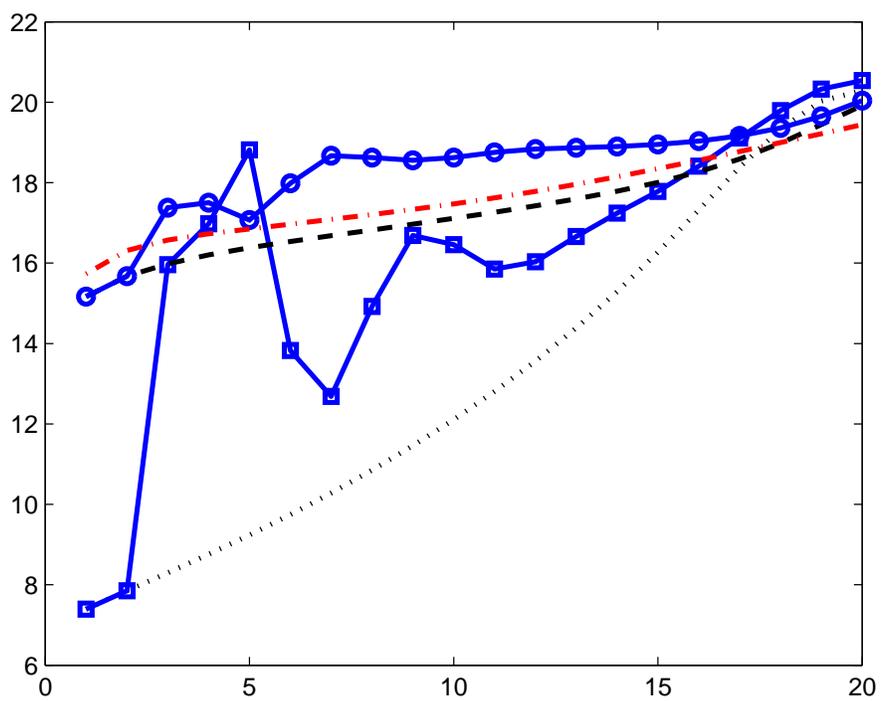
Fig. 5. Jet flow: comparison of the SNR change as the number of iteration increases, by different methods: IST (dotted line), AMP (solid line with squares), IST-DWT (dot-dashed line), IST-DTCWT (dashed line), AMP-DTCWT (solid line with circles).

(a)

(b) SNR = 24.1243 dB

(c) SNR= 24.2749 dB

(d) SNR=27.2177 dB

(e) SNR= 27.5431 dB

(f) SNR=28.8220 dB

Fig. 6. Foreman: the comparison of the 32nd frame with SNR values. (a) Original data. (b) IST reconstruction. (c) AMP reconstruction. (d) IST-DWT reconstruction. (e) IST-DTCWT reconstruction. (f) AMP-DTCWT reconstruction.
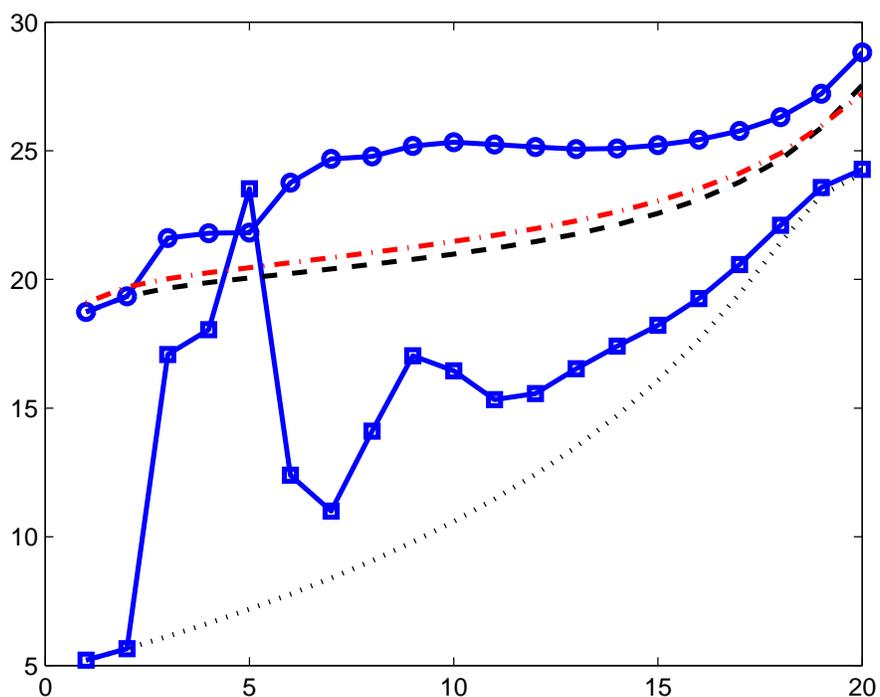
Fig. 7. Foreman: comparison of the SNR change as the number of iteration increases, by different methods: IST (dotted line), AMP (solid line with squares), IST-DWT (dot-dashed line), IST-DTCWT (dashed line), AMP-DTCWT (solid line with circles).

(a) SNR=20.2524 dB   (b) SNR=26.2758 dB
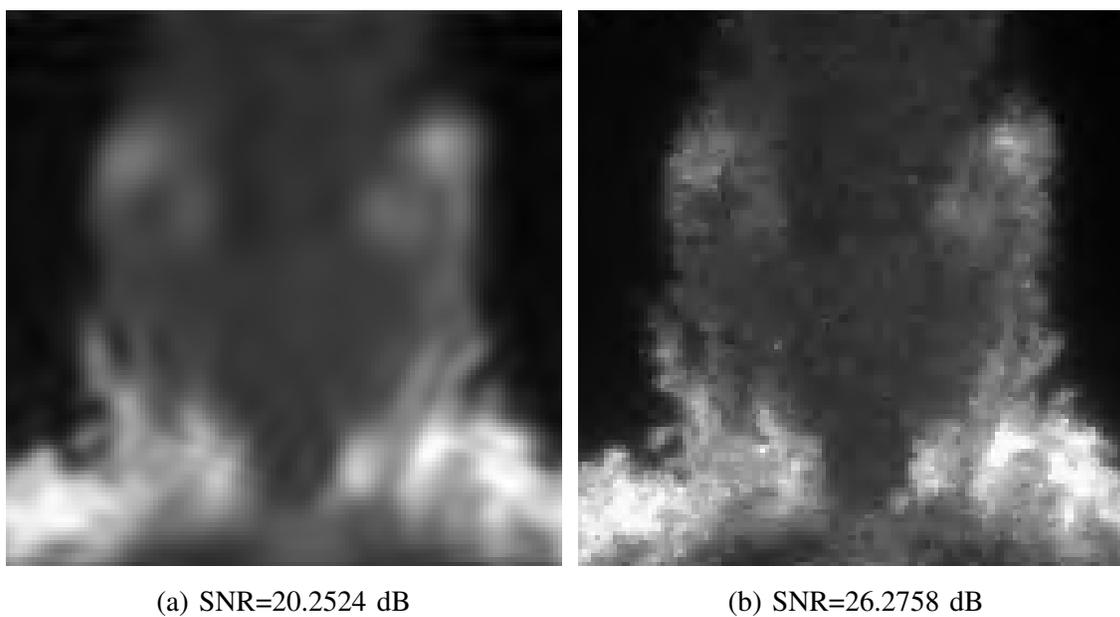
Fig. 8. Compression by tradition methods by keeping 1628 most significant coefficients. (a) Curvelet thresholding and reconstruction. (b) Wavelet thresholding and reconstruction.

Index of frames		IST	AMP	IST-DWT	IST-DTCWT	AMP-DTCWT
	Cartoon	17.7345	17.6737	18.1336	18.6439	19.0492
24	Jet Flow	20.4525	20.9124	20.2709	20.6069	20.7331
	Foreman	24.3111	24.5051	27.2768	27.3916	28.5506
	Cartoon	17.75711	17.6990	18.1104	18.6921	19.1120
25	Jet Flow	21.4331	22.0000	21.2142	22.0077	22.1323
	Foreman	24.2957	24.5088	27.1866	27.5058	28.6755
	Cartoon	17.7920	17.7315	19.9471	18.6703	19.0913
26	Jet Flow	21.5136	21.9198	20.9791	21.8937	22.0583
	Foreman	24.1805	24.3838	27.0332	27.4727	28.6829
	Cartoon	17.8206	17.7554	18.0934	18.7439	19.1499
27	Jet Flow	20.5071	20.9212	19.8141	20.5992	20.7513
	Foreman	24.1164	24.3196	26.9539	27.4545	28.6222
	Cartoon	17.8514	17.7809	18.1302	18.7549	19.1449
28	Jet Flow	20.618	20.7305	19.5977	20.4009	20.5224
	Foreman	24.1414	24.3415	27.1282	27.5848	28.8416
	Cartoon	17.8936	17.8280	18.2154	18.7797	19.1433
29	Jet Flow	20.9468	21.2433	19.8733	20.9726	21.1308
	Foreman	24.1539	24.3556	27.1438	27.6813	28.9765
	Cartoon	17.9993	17.9340	18.2078	18.8576	30.2334
30	Jet Flow	21.0428	21.3640	20.0200	20.9726	21.1461
	Foreman	24.1786	24.3753	27.0515	27.6987	28.9597
	Cartoon	18.0626	17.9962	18.3554	18.9867	19.3739
31	Jet Flow	21.2379	20.6074	19.3877	20.0516	20.1830
	Foreman	24.1752	24.3495	27.1046	27.7054	28.9376
	Cartoon	18.0879	18.0234	18.5335	18.9709	19.3739
32	Jet Flow	20.2882	20.5407	19.5246	19.9223	20.0399
	Foreman	24.1243	24.2749	27.2177	27.5431	28.8220
	Cartoon	18.0703	18.013	18.4899	18.9079	19.2490
33	Jet Flow	20.9096	21.1414	20.4178	20.7871	20.9408
	Foreman	24.1045	24.2510	27.3676	27.3849	28.5790
	Cartoon	18.1053	18.0517	18.3943	18.9227	19.2574
34	Jet Flow	21.4247	21.6615	20.3987	21.2156	21.4216
	Forman	24.2307	24.3910	27.5602	27.6002	28.7615
	Cartoon	18.1589	18.0998	18.4187	19.0421	19.3853
35	Jet Flow	20.8079	20.9425	19.6219	20.3421	20.5023
	Foreman	24.1418	24.3129	27.5474	27.6081	28.7687
	Cartoon	18.1354	18.0790	18.3487	19.0620	19.4218
36	Jet Flow	20.2938	20.4672	19.5286	20.3488	20.5420
	Foreman	24.1169	24.2945	27.5100	27.5818	28.7233
	Cartoon	18.0913	18.0342	18.3958	18.9946	19.3949
37	Jet Flow	21.2720	21.3680	20.1066	21.2472	21.4048
	Foreman	24.1788	24.3677	27.6053	27.7354	28.8988
	Cartoon	18.0896	18.0339	18.4531	18.9652	19.3962
38	Jet Flow	21.4698	21.5835	20.1596	21.4440	21.6561
	Foreman	24.2667	24.4673	27.7694	27.8174	28.9921
	Cartoon	18.1181	18.0584	18.5669	19.0419	19.4949
39	Jet Flow	21.0320	21.1768	19.8859	20.8085	20.9638
	Foreman	24.3000	24.5068	27.7838	27.8284	28.9658
	Cartoon	17.9859	17.9167	18.3585	18.8232	19.2287
Average	Jet Flow	20.7646	21.0397	20.1236	20.7259	20.9044
	Foreman	24.2973	24.4589	27.5235	27.4845	28.7078

TABLE I

PERFORMANCE COMPARISON IN SNR VALUE BY DIFFERENT METHODS, FOR CARTOON VIDEO, JET FLOW AND
FORMAN VIDEO.