A Nonlocal Transform-Domain Filter for Volumetric Data Denoising and Reconstruction

Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, Alessandro Foi

Abstract—We present an extension of the BM3D filter to volumetric data. The proposed algorithm, denominated BM4D, implements the grouping and collaborative filtering paradigm, where mutually similar d-dimensional patches are stacked together in a $(d + 1)$-dimensional array and jointly filtered in transform domain. While in BM3D the basic data patches are blocks of pixels, in BM4D we utilize cubes of voxels, which are stacked into a four-dimensional "group". The four-dimensional transform applied on the group simultaneously exploits the local correlation present among voxels in each cube and the nonlocal correlation between the corresponding voxels of different cubes. Thus, the spectrum of the group is highly sparse, leading to very effective separation of signal and noise through coefficients shrinkage. After inverse transformation, we obtain estimates of each grouped cube, which are then adaptively aggregated at their original locations. We evaluate the algorithm on denoising of volumetric data corrupted by Gaussian and Rician noise, as well as on reconstruction of volumetric phantom data with non-zero phase from noisy and incomplete Fourier-domain (k-space) measurements. Experimental results demonstrate the state-of-the-art denoising performance of BM4D, and its effectiveness when exploited as a regularizer in volumetric data reconstruction.

Index Terms—Volumetric data denoising, volumetric data reconstruction, compressed sensing, magnetic resonance imaging, computed tomography, nonlocal methods, adaptive transforms

I. INTRODUCTION

The past six years have witnessed substantial developments in the field of image restoration. In particular, for what concerns image denoising, starting with the adaptive spatial estimation strategy termed nonlocal means (NLmeans) [1], it soon became clear that self-similarity and nonlocality are the characteristics of natural images with by far the biggest potential for image restoration. In NLmeans, the basic idea is to build a pointwise estimate of the image where each pixel is obtained as a weighted average of pixels centered at regions that are similar to the region centered at the estimated pixel. The estimates are nonlocal because, in principle, the averages can be calculated over all pixels of the image. One of the most powerful and effective extensions of the nonlocal filtering approach is the grouping and collaborative filtering paradigm embodied by the BM3D image denoising algorithm [2]. This algorithm is based on an enhanced sparse representation in transform domain. The enhancement of the sparsity is achieved by grouping similar 2-D fragments of the image into 3-D data arrays which are called "group". Such groups are processed through a special procedure, named collaborative filtering, which consists of three successive steps: firstly a 3-D transformation is applied to the group, secondly the transformed group coefficients are shrunk, and finally a 3-D group estimate is obtained by inverting the 3-D transformation. Due to the similarity between the grouped

fragments, the noise can be well separated by shrinkage because the 3-D transformation discloses a highly sparse representation of the true signal in transform domain. In this way, the collaborative filtering reveals even the finest details shared by the jointly filtered 2-D fragments preserving at the same time their essential unique features. The BM3D algorithm presented in [2] represents the current state of the art in 2-D image denoising, demonstrating a performance significantly superior to that of all previously existing methods. Recent works discuss the near-optimality of this approach and offer further insights about the rationale of the algorithm [3], [4].

In this work, we present an extension of the BM3D algorithm to volumetric data denoising. While in BM3D the basic data patches are blocks of pixels, in the proposed algorithm, denominated BM4D, we naturally utilize cubes of voxels. The group formed by stacking mutually similar cubes is hence a four-dimensional orthope (hyperrectangle) whose fourth dimension, along which the cubes are stacked, embodies the nonlocal correlation across the data. Thus, collaborative filtering simultaneously exploits the local correlation present among voxels in each cube as well as the nonlocal correlation between the corresponding voxels of different cubes. As in BM3D, the spectrum of the group is highly sparse, leading to a very effective separation of signal and noise by either thresholding or Wiener filtering. After inverse transformation, we obtain the estimates of each grouped cube, which are then aggregated at their original locations using adaptive weights.

Further we exploit BM4D as a regularizer operator for the reconstruction of incomplete volumetric data. The proposed procedure generalizes [5], [6], as it addresses the reconstruction of volumetric data having non-zero phase from a set of incomplete noisy transform-domain measurements. Our reconstruction procedure works iteratively. In each iteration the missing part of the spectrum is excited with random noise; then, after transforming the excited spectrum to the voxel domain, the BM4D filter attenuates the noise present in both magnitude and phase of the data, thus disclosing even the faintest details from the incomplete and degraded observations. The overall procedure can be interpreted as a progressive approximation in which the denoising filter directs the stochastic search towards the solution.

Experimental results on volumetric data from the BrainWeb database [7] demonstrate the state-of-the-art performance of the proposed algorithm. In particular, we report significant improvement over the results achieved by the optimized volumetric implementations of the NLmeans filter [8], [9], [10], [11], which, to the best of our knowledge, are the most successful approaches in magnetic resonance (MR). We also test BM4D against real MR data provided by the OASIS database [12]. As for the reconstruction experiments, our iterative procedure achieves excellent performance for both the 3-D Shepp-Logan [13], [14] and BrainWeb phantoms sampled by various trajectories.

The remainder of paper is organized as follows. In Section II we formally define the observation model, the BM4D implementation, and the adopted parameters. The denoising experiments are analyzed in Section III. In Section IV we first describe the volumetric reconstruction procedure, and then in Section V we report its experimental validation. Concluding remarks are given in Section VI.

All authors are with the Department of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland (e-mail: firstname.lastname@tut.fi).
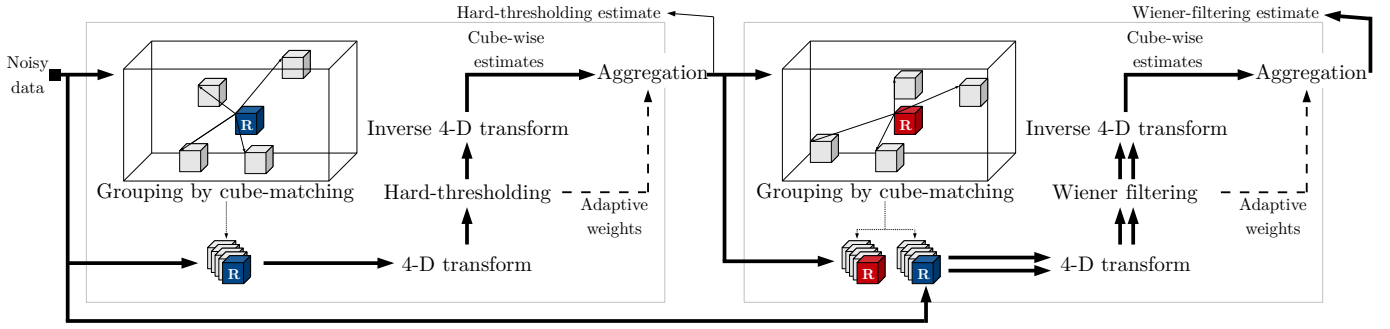
Fig. 1. Flow-diagram of the proposed BM4D algorithm. In both Hard-thresholding (left box) and Wiener-filtering (right box) stage, the grouping, collaborative filtering and aggregation steps are performed for each reference cube of the observed volumetric data.

II. BM4D ALGORITHM

A. Observation Model

For the development of the BM4D algorithm, we consider noisy volumetric observation $z : X \to \mathbb{R}$ of the form

$$z(x) = y(x) + \eta(x), \quad x \in X, \tag{1}$$

where y is the original, unknown, volumetric signal, x is a 3-D coordinate belonging to the signal domain $X \subset \mathbb{Z}^3$, and $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$ is independent and identically distributed (i.i.d.) Gaussian noise with zero mean and known standard deviation σ.

B. Implementation

The objective of the proposed BM4D is to provide an estimate $\hat{y}$ of the original y from the noisy observation z. Similarly to the BM3D algorithm, BM4D is implemented in two cascading stages, namely a hard-thresholding and a Wiener-filtering stage, each comprising three steps: grouping, collaborative filtering, and aggregation. The flow-diagram of the BM4D implementation is illustrated in Fig. I.

1) *Hard-thresholding stage:* Let $C_{x_R}^z$ denote a cube of $L \times L \times L$ voxels, with $L \in \mathbb{N}$, extracted from z at the 3-D coordinate $x_R \in X$, which identifies its top-left-front corner. In the hard-thresholding stage, the four-dimensional groups are formed by stacking together, along an additional fourth dimension, (three-dimensional) noisy cubes similar to $C_{x_R}^z$. Specifically, the similarity between two cubes is measured via the photometric distance

$$d\left(C_{x_i}^z, C_{x_j}^z\right) = \frac{\left\|C_{x_i}^z - C_{x_j}^z\right\|_2^2}{L^3}, \tag{2}$$

where $\|\cdot\|_2^2$ denotes the sum of squared differences between corresponding intensities of the two input cubes, and the denominator L^3 serves as normalization factor. No prefiltering is performed before the cube-matching, therefore the noisy observations are directly tested for similarity.

In the grouping step, a group consisting of mutually similar cubes extracted from z is built for every (reference) cube $C_{x_R}^z$. Two cubes are considered similar if their distance (2) is smaller than or equal to a predefined threshold $\tau_{\text{match}}^{\text{ht}}$ which thus controls the minimum accepted cube-similarity. Formally, we first define a set containing the indices of the cubes similar to $C_{x_R}^z$ as

$$S_{x_R}^z = \left\{x_i \in X : d\left(C_{x_R}^z, C_{x_i}^z\right) \leq \tau_{\text{match}}^{\text{ht}}\right\}. \tag{3}$$

Then, such (3) is used to build the four-dimensional group

$$\mathbf{G}_{S_{x_R}^z}^z = \coprod_{x_i \in S_{x_R}^z} C_{x_i}^z, \tag{4}$$

being $\coprod$ the disjoint union operation. This process is exemplified in Fig. I, where the reference cube, denoted by "R", is matched to a series of similar cubes located anywhere within the 3-D data. In particular, the coordinate x_R and the various x_i in (3) correspond to the tails and the heads of the arrows connecting the cubes, respectively. Observe that, since the distance of any cube to itself is always zero, from the definition of (3) follows that each group (4) necessarily contains at least the reference cube $C_{x_R}^z$.

During the collaborative filtering step, four 1-D decorrelating linear transform, which we denote as a joint four-dimensional transform $\mathcal{T}_{4D}^{\text{ht}}$, are separately applied to every dimension of the group (4). The so-obtained 4-D group spectrum is then shrunk coefficient by coefficient by a hard-thresholding operator Υ^{ht} with threshold value $\sigma\lambda_{4D}$ as

$$\Upsilon^{\text{ht}}\left(\mathcal{T}_{4D}^{\text{ht}}\left(\mathbf{G}_{S_{x_R}^z}^z\right)\right). \tag{5}$$

The transform $\mathcal{T}_{4D}^{\text{ht}}$ is assumed to have a DC term, which is never shrunk during the collaborative filtering so that the mean value of the group is preserved. Eventually, the filtered group, denoted as $\hat{\mathbf{G}}_{S_{x_R}^z}^y$, is produced by inverting the four-dimensional transform as

$$\mathcal{T}_{4D}^{\text{ht}^{-1}}\left(\Upsilon^{\text{ht}}\left(\mathcal{T}_{4D}^{\text{ht}}\left(\mathbf{G}_{S_{x_R}^z}^z\right)\right)\right) = \hat{\mathbf{G}}_{S_{x_R}^z}^y = \coprod_{x_i \in S_{x_R}^z} \hat{C}_{x_i}^y, \tag{6}$$

being each $\hat{C}_{x_i}^y$ an estimate of the original $C_{x_i}^y$ extracted from the unknown volumetric data y.

The groups (6) are an overcomplete representation of the denoised signal, because cubes in different groups, as well as cubes within the same group, are likely to overlap; as a result, within the overlapping regions, different cubes provides multiple, and in general different, estimates for the same voxel. In the aggregation step, such redundancy is exploited through an adaptive convex combination to produce the basic volumetric estimate

$$\hat{y}^{\text{ht}} = \frac{\sum_{x_R \in X}\left(\sum_{x_i \in S_{x_R}^z} w_{x_R}^{\text{ht}} \hat{C}_{x_i}^y\right)}{\sum_{x_R \in X}\left(\sum_{x_i \in S_{x_R}^z} w_{x_R}^{\text{ht}} \chi_{x_i}\right)}, \tag{7}$$

where $w_{x_R}^{\text{ht}}$ are group-dependent weights, $\chi_{x_i} : X \to \{0, 1\}$ is the characteristic (indicator) function of the domain of $\hat{C}_{x_i}^y$ (i.e. $\chi_{x_i} = 1$ over the coordinates of the voxels of $\hat{C}_{x_i}^y$ and $\chi_{x_i} = 0$ elsewhere), and every $\hat{C}_{x_i}^y$ is assumed to be zero-padded outside its domain. Note that, whereas in BM3D a 2-D Kaiser window of the same size of the blocks is used to alleviate blocking artifacts in the aggregated estimate [2], in the proposed BM4D we do not perform such windowing, because of the small size of the cubes. The weights in (7) are defined as

$$w_{x_R}^{\text{ht}} = \frac{1}{\sigma^2 N_{x_R}^{\text{ht}}}, \tag{8}$$

where σ is the standard deviation of the noise in z, and $N^{\text{ht}}_{x_R}$ denotes the number of non-zero coefficients in (5). Since the DC coefficient is always retained after thresholding, i.e. $N^{\text{ht}}_{x_R} \geq 1$, the denominator of (8) is never zero. Note that the number $N^{\text{ht}}_{x_R}$ has a double interpretation: on one hand it measures the sparsity of the thresholded spectrum (5), and on the other, as explained in [2], it approximates the total residual noise variance of the group estimate (6). Thus, those groups exhibiting a high degree of correlation are rewarded with larger weights, whereas others having a large residual noise are penalized by smaller weights.

2) Wiener-filtering stage: In the Wiener-filtering stage, the grouping is performed within the basic estimate $\hat{y}^{\text{ht}}$. We expect the obtain a more accurate and reliable matching because the noise level in $\hat{y}^{\text{ht}}$ is considerably smaller than that in z. We are interested in improving the matching because a better grouping leads to a more effective sparsification of the group spectrum, which in turn results in a superior denoising quality. Formally, for each reference cube $C^{\hat{y}^{\text{ht}}}_{x_R}$ extracted from the basic estimate $\hat{y}^{\text{ht}}$, we build the set of the coordinates of its similar cubes as

$$S^{\hat{y}^{\text{ht}}}_{x_R} = \left\{ x_i \in X : d\left(C^{\hat{y}^{\text{ht}}}_{x_R}, C^{\hat{y}^{\text{ht}}}_{x_i}\right) < \tau^{\text{wie}}_{\text{match}} \right\}, \quad (9)$$

where $d(\cdot, \cdot)$ is defined as in (2).

The collaborative filtering is implemented as an empirical Wiener filter. Analogously to (4), at first a group $\mathbf{G}^{\hat{y}^{\text{ht}}}_{S^{\hat{y}^{\text{ht}}}_{x_R}}$ is extracted from $\hat{y}^{\text{ht}}$ using the set of coordinates (9), then from the energy of its spectrum we define the empirical Wiener filter coefficients as

$$\mathbf{W}_{S^{\hat{y}^{\text{ht}}}_{x_R}} = \frac{\left|\mathcal{T}^{\text{wie}}_{4D}\left(\mathbf{G}^{\hat{y}^{\text{ht}}}_{S^{\hat{y}^{\text{ht}}}_{x_R}}\right)\right|^2}{\left|\mathcal{T}^{\text{wie}}_{4D}\left(\mathbf{G}^{\hat{y}^{\text{ht}}}_{S^{\hat{y}^{\text{ht}}}_{x_R}}\right)\right|^2 + \sigma^2}, \quad (10)$$

where σ denotes the standard deviation of the noise, and $\mathcal{T}^{\text{wie}}_{4D}$ is a transform operator composed by four 1-D linear transformations, which are in general different than those in $\mathcal{T}^{\text{ht}}_{4D}$. Subsequently, we use the same set (9) to extract a second (noisy) group, termed $\mathbf{G}^{z}_{S^{\hat{y}^{\text{ht}}}_{x_R}}$, from the observation z. The coefficients shrinkage is implemented as element-by-element multiplication between the spectrum of the noisy group and the Wiener-filter coefficients (10). The estimate of the group

$$\hat{\mathbf{G}}^{y}_{S^{\hat{y}^{\text{ht}}}_{x_R}} = \mathcal{T}^{\text{wie}^{-1}}_{4D}\left(\mathbf{W}_{S^{\hat{y}^{\text{ht}}}_{x_R}} \cdot \mathcal{T}^{\text{wie}}_{4D}\left(\mathbf{G}^{z}_{S^{\hat{y}^{\text{ht}}}_{x_R}}\right)\right) \quad (11)$$

is finally produced by applying the inverse four-dimensional transform $\mathcal{T}^{\text{wie}^{-1}}_{4D}$ to the shrunk spectrum

The final estimate $\hat{y}^{\text{wie}}$ is produced through a convex combination, analogous to (7), in which the sets (3) are replaced with (9), and the aggregation weights for a specific group estimate (11) are defined from the energy of the Wiener-filter coefficients (10) as

$$w^{\text{wie}}_{x_R} = \sigma^{-2} \left|\left|\mathbf{W}_{S^{\hat{y}^{\text{ht}}}_{x_R}}\right|\right|^{-2}_2, \quad (12)$$

where σ is the standard deviation of the noise in z. In this way, as in [2], each (12) gives an estimate of the total residual noise variance of the corresponding group (11).

III. DENOISING EXPERIMENTS

We validate the denoising capabilities of BM4D[1] using noisy magnetic resonance phantoms, because we recognize medical imaging to

---

[1] MATLAB code available at http://www.cs.tut.fi/~foi/GCF-BM3D/

---

be one of the most prominent applications based on volumetric data. We measure the objective quality of the denoising trough its PSNR

$$\text{PSNR}(y, \hat{y}) = 10 \log_{10}\left(\frac{D^2 |\tilde{X}|}{\sum_{x \in \tilde{X}} (\hat{y}(x) - y(x))^2}\right),$$

where D is the peak of y, $\tilde{X} = \{x \in X : y(x) > 10 \cdot D/255\}$ (in order not to compute the PSNR on the background as in [8]), and $|\tilde{X}|$ is the cardinality of $\tilde{X}$. We also evaluate our experiments with the structure similarity index (SSIM), that is a metric originally presented for 2-D images in [15] and extended to 3-D data in [8] that better relates to the human visual system than traditional methods based on the mean squared error such as the PSNR. In what follows, without loss of generality, we assume to deal with real-valued signals normalized to the intensity range $[0, 1]$ (i.e. $D = 1$).

The experiments are made under both Gaussian- and Rician-distributed noise. In the former case, the noisy observations z are distributed accordingly to (1); in the latter, the noisy observations $z : X \to \mathbb{R}^+$ follow the definition

$$z(x) = \sqrt{(c_r y(x) + \sigma \eta_r(x))^2 + (c_i y(x) + \sigma \eta_i(x))^2}, \quad (13)$$

where x is a 3-D coordinate belonging to the domain $X \subset \mathbb{Z}^3$, c_r and c_i are constants satisfying the condition $0 \leq c_r, c_i \leq 1 = c_r^2 + c_i^2$, and $\eta_r(\cdot), \eta_i(\cdot) \sim \mathcal{N}(0, 1)$ are i.i.d. random vectors following the standard normal distribution. In this way, $z \sim \mathcal{R}(y, \sigma)$ represents the raw magnitude MR data, modeled as a Rician distribution $\mathcal{R}$ of parameters y and σ, denoting the (unknown) original noise-free signal and the standard deviation of the Rician noise, respectively [16].

Leveraging a recently proposed method of variance-stabilization (VST) [16] for the Rician distribution, BM4D can be successfully applied to data distributed as in (13) without incorporating any adaptation to the algorithm. The purpose of the VST is to remove the dependency of the noise variance on the underlying signal before the denoising, and compensate the effects of the bias in the produced filtered estimate. Formally, the denoising of Rician data via the BM4D algorithm is expressed as

$$\hat{y} = \text{VST}^{-1}\left(\text{BM4D}\left(\text{VST}(z, \sigma), \sigma_{\text{VST}}\right), \sigma\right), \quad (14)$$

where VST^{-1} denotes the inverse variance-stabilization transformation, σ_{VST} is the stabilized standard deviation induced by the VST, and σ is the standard deviation of the noise in (13). Thus, the noisy Rician data z is first stabilized by the VST and then filtered by BM4D using a constant noise level σ_{VST}; the final estimate is finally obtained by applying the inverse VST to the output of the denoising. Note that this inverse is not the trivial algebraic inverse of the forward VST, but it includes further nonlinearities in order to compensate both the bias due to forward stabilization and the bias due to the non-zero mean of the Rician noise [16].

The volumetric test data y is the T1 BrainWeb phantom of size $181 \times 217 \times 181$ voxels having 1mm slice thickness, 0% noise, and 0% intensity non-uniformity [7]. We synthetically generate the noisy observations z accordingly to (1) and (13) using different values of standard deviation σ, ranging from 1% to 19% of the maximum value D of the original signal y.

In order to provide relevant comparisons, we validate the denoising performance of the BM4D algorithm against the optimized blockwise nonlocal means OB-NLM3D [10], the optimized blockwise nonlocal means with wavelet mixing OB-NLM3D-WM [11], the oracle-based 3-D DCT ODCT3D [8], and the prefiltered rotationally invariant nonlocal means PRI-NLM3D [8]. To the best of our knowledge, ODCT3D and PRI-NLM3D represent the state of the art in MR image denoising. The OB-NLM3D, OB-NLM3D-WM, ODCT3D, and PRI-NLM3D algorithms exist in separate implementations developed for

TABLE I
PARAMETER SETTINGS FOR THE PROPOSED BM4D ALGORITHM.

Parameter		Stage			
		Hard thresholding		Wiener filtering	
		Normal	Modif.	Normal	Modif.
Cube size	L	4		4	5
Group size	M	16	32	32	
Step	N_{step}	3			
Search-cube size	N_S	11			
Similarity thr.	τ_{match}	2.9	24.6	0.4	6.7
Shrinkage thr.	λ_{4D}	2.7	2.8	Does not apply	

TABLE III
ACQUISITION DETAILS OF THE OASIS "OAS1_0108_MR1" MRI
CROSS-SECTIONAL DATA.

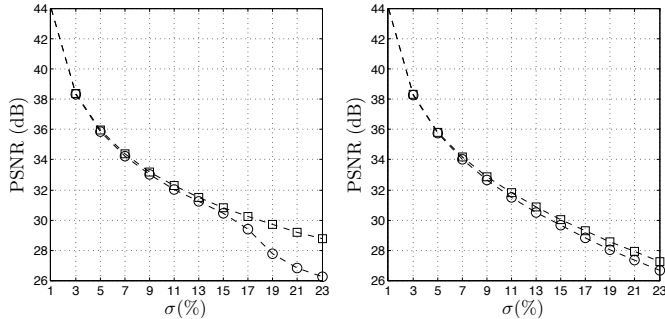| MP-RAGE OAS1_0108_MR1 sequence	
TR (msec)	9.7
TE (msec)	4.0
Flip angle (deg)	10
TI (msec)	20
TD (msec)	200
Orientation	Sagittal
Dimension (voxels)	$256 \times 256 \times 128$
Resolution (mm)	$1.0 \times 1.0 \times 1.25$

Fig. 2. PSNR denoising performance of BM4D under the normal (○) and modified (□) profile applied to the BrainWeb phantom [7] corrupted by i.i.d. Gaussian noise (left) and Rician noise (right) with varying level of σ.

Gaussian- and Rician-distributed noise, thus we decorate their names with a subscript "$\mathcal{N}$" (Gaussian) and "$\mathcal{R}$" (Rician) to denote the noise distribution addressed by the specific algorithm implementation.

A. Algorithm Parameters

We set the size of the cubes in BM4D in such a way that the cubes contain roughly as many voxels as the number of pixels in the 2-D blocks in BM3D. In this manner, we are able to successfully utilize most of the settings originally optimized for BM3D. Since the BM3D algorithm is presented under two sets of parameter, namely the normal and modified profile in which the blocks have size 8 and 11 [2], we correspondingly define for BM4D two analogous profiles having cube size $L = 4$ and $L = 5$.

The separable four-dimensional transforms of BM4D are similar to those in [2]. In the hard-thresholding stage $\mathcal{T}_{4D}^{\text{ht}}$ is a composition of a 3-D biorthogonal spline wavelet in the cube dimensions (note that, due to the small L, this transform is actually equivalent to a 3-D Haar separable transform) and a 1-D Haar wavelet in the grouping dimension; in the Wiener-filtering stage $\mathcal{T}_{4D}^{\text{wie}}$ embeds a 3-D discrete cosine transform (DCT) in the cube dimensions and, again, a 1-D Haar wavelet in the grouping dimension. The Haar transform in the fourth dimension restricts the cardinality of the groups to be a power of two, but, since such cardinality is not known a priori, we constrain the number of grouped cubes to be the largest power of 2 smaller than or equal to the minimum value between the original cardinality of the groups and a predefined value M. Then, in order to reduce the computational complexity of the algorithm, the grouping is performed within a three-dimensional window of size $N_S \times N_S \times N_S$ centered at the coordinate of the current reference cube, and all such reference cubes are separated by a step $N_{\text{step}} \in \mathbb{N}$ in every spatial dimension. Table I summarizes the role and the value of all parameters utilized by BM4D.

In the modified profile, following the comments suggested in [17], we increase the values of the similarity thresholds τ, the group size M, the cube size L, and the hard-threshold value λ_{4D}. The rationale behind such modifications consists in improving both the reliability of the matching by using larger cubes, and the effectiveness of the collaborative filtering by promoting the formation of bigger groups. The denoising performance of BM4D under both the normal and modified profile with increasing values of standard-deviation σ (for both Gaussian- and Rician-distributed data) is illustrated in Fig. 2. As one can see, the modified profile consistently provides the best PSNR performance, especially in cases when the noise variance is large, i.e. $\sigma > 15\%$. The results present a consistent behavior with Figure 9 in [2], where the two different profiles are compared in 2-D image denoising. These results are explained by the nature of MR images, as modeled by the BrainWeb phantom, predominantly characterized by low-frequency content, abundance of similar patches, and a vast smooth background. The modified profile leverages such attributes because, on one hand, it tends to form groups having maximum cardinality, and, on the other, it applies a slightly more aggressive smoothing through the larger λ_{4D}. That being so, we choose to always utilize the modified parameters for our experimental evaluation.

B. Denoising of BrainWeb Phantom

Table II reports the PSNR and SSIM performance for the OB-NLM3D, OB-NLM3D-WM, ODCT3D, PRI-NLM3D, and BM4D filters. The proposed BM4D algorithm always achieves the best results both in case of Gaussian- and Rician-distributed noise, with PSNR improvements on the current state-of-the-art filters [8] roughly ranging between 0.5dB and 1.4dB. Additionally, we observe that, among the considered algorithms, the PSNR and SSIM performance of BM4D exhibits the most graceful degradation as noise level σ increases. Fig. 8 shows a cross-section of the BrainWeb phantom, denoised by all algorithms; the illustrated noisy observation, shown in Fig. 7(c), has been corrupted by i.i.d. Gaussian noise having $\sigma = 15\%$. From a subjective point of view, BM4D achieves an excellent visual quality, as can be seen from the smoothness in flat areas, the details preservation along the edges, and the accurate preservation of the intensities in the restored phantom.

C. Denoising of Real Magnetic Resonance Data

The denoising algorithms have been also tested on real cross-sectional MR data made publicly available by the Open Access Series of Imaging Studies (OASIS) database [12]. The T1-weighted magnetization prepared rapid gradient-echo (MP-RAGE) 16-bit images have been acquired via a 1.5-T Vision scanner (Siemens, Erlangen, Germany) in a single imaging session, additional details on the

TABLE II

PSNR (LEFT VALUE IN EACH CELL) AND SSIM [15], [8] (RIGHT VALUE IN EACH CELL) DENOISING PERFORMANCES ON THE VOLUMETRIC TEST DATA FROM THE BRAINWEB DATABASE [7] OF THE PROPOSED BM4D (UNDER THE MODIFIED PROFILE) AND THE OB-NLM3D [10], OB-NLM3D-WM [11], [18], ODCT3D [8], AND PRI-NLM3D [8] FILTERS. TWO KINDS OF OBSERVATIONS ARE TESTED, ONE CORRUPTED BY I.I.D. GAUSSIAN AND THE OTHER BY SPATIALLY HOMOGENOUS RICIAN NOISE ACCORDING TO THE OBSERVATION MODELS (1) AND (13). BOTH CASES ARE TESTED UNDER DIFFERENT STANDARD-DEVIATIONS σ, EXPRESSED AS PERCENTAGE RELATIVE TO THE MAXIMUM INTENSITY VALUE OF THE ORIGINAL VOLUMETRIC DATA. VST REFERS TO THE VARIANCE-STABILIZATION FRAMEWORK DEVELOPED FOR RICIAN-DISTRIBUTED DATA [16]. THE SUBSCRIPTS $\mathcal{N}$ (GAUSSIAN) AND $\mathcal{R}$ (RICIAN) DENOTE THE ADDRESSED NOISE DISTRIBUTION.

Noise	Filter	σ									
		1%	3%	5%	7%	9%	11%	13%	15%	17%	19%
Gauss.	(Noisy data)	40.00|0.97	30.46|0.81	26.02|0.66	23.10|0.53	20.91|0.43	19.17|0.36	17.72|0.30	16.48|0.25	15.39|0.22	14.42|0.19
	OB-NLM3D$_\mathcal{N}$	42.47|**0.99**	37.57|0.97	34.73|0.95	32.82|0.92	31.42|0.90	30.32|0.87	29.40|0.84	28.61|0.82	27.91|0.79	27.28|0.77
	OB-NLM3D-WM$_\mathcal{N}$	42.52|**0.99**	37.75|0.97	35.01|0.95	33.13|0.93	31.73|0.90	30.61|0.88	29.68|0.85	28.88|0.83	28.18|0.80	27.55|0.78
	ODCT3D$_\mathcal{N}$	43.78|**0.99**	37.53|0.97	34.89|0.95	33.18|0.93	31.91|0.91	30.90|0.89	30.07|0.88	29.35|0.86	28.73|0.85	28.18|0.83
	PRI-NLM3D$_\mathcal{N}$	44.04|**0.99**	38.26|**0.98**	35.51|**0.96**	33.67|0.94	32.37|0.92	31.29|0.90	30.40|0.89	29.65|0.87	28.99|0.85	28.40|0.84
	BM4D	**44.09|0.99**	**38.39|0.98**	**35.95|0.96**	**34.38|0.95**	**33.21|0.93**	**32.28|0.92**	**31.50|0.91**	**30.82|0.90**	**30.23|0.88**	**29.70|0.87**
Rician	(Noisy data)	40.00|0.97	30.49|0.81	26.09|0.66	23.20|0.53	21.04|0.43	19.32|0.36	17.88|0.30	16.65|0.25	15.57|0.21	14.60|0.18
	OB-NLM3D$_\mathcal{R}$	42.41|**0.99**	37.45|0.97	34.54|0.94	32.51|0.91	30.97|0.88	29.71|0.85	28.62|0.81	27.64|0.78	26.74|0.74	25.91|0.70
	VST + OB-NLM3D$_\mathcal{N}$	42.48|**0.99**	37.45|0.97	34.40|0.94	32.26|0.91	30.65|0.88	29.34|0.85	28.23|0.81	27.25|0.78	26.37|0.74	25.57|0.71
	OB-NLM3D-WM$_\mathcal{R}$	42.44|**0.99**	37.54|0.97	34.66|0.95	32.61|0.92	31.01|0.88	29.69|0.85	28.53|0.81	27.50|0.77	26.57|0.74	25.71|0.70
	VST + OB-NLM3D-WM$_\mathcal{N}$	42.53|**0.99**	37.68|0.97	34.75|0.95	32.68|0.92	31.06|0.89	29.77|0.86	28.68|0.83	27.71|0.80	26.84|0.76	26.04|0.73
	ODCT3D$_\mathcal{R}$	42.96|**0.99**	37.38|0.97	34.70|0.95	32.90|0.93	31.53|0.90	30.41|0.88	29.48|0.86	28.67|0.84	27.95|0.82	27.30|0.80
	VST + ODCT3D$_\mathcal{N}$	43.74|**0.99**	37.51|0.97	34.79|0.95	32.98|0.93	31.59|0.90	30.47|0.88	29.52|0.86	28.71|0.84	27.98|0.82	27.31|0.80
	PRI-NLM3D$_\mathcal{R}$	43.97|**0.99**	38.19|**0.98**	35.34|**0.96**	33.37|**0.94**	31.94|0.91	30.74|0.89	29.75|0.87	28.88|0.85	28.10|0.82	27.39|0.80
	VST + PRI-NLM3D$_\mathcal{N}$	**44.21|0.99**	38.20|**0.98**	35.34|**0.96**	33.36|**0.94**	31.90|0.91	30.71|0.89	29.71|0.87	28.88|0.85	28.13|0.82	27.46|0.80
	VST + BM4D	44.08|**0.99**	**38.34|0.98**	**35.83|0.96**	**34.17|0.94**	**32.89|0.93**	**31.82|0.91**	**30.90|0.89**	**30.06|0.88**	**29.29|0.86**	**28.57|0.84**

acquisition process are summarized in Table III. The (anonymous) test subject is a 25-years old right-handed male with no brain damages. The noise has been assumed to be Rician-distributed, and its standard deviation, estimated as described in [16], is approximately $\sigma \approx 4\%$ of the maximum intensity value of the data. The acquired phantom is shown in Fig. 7(d), whereas Fig. 8 shows the corresponding denoised results produced by the OB-NLM3D, OB-NLM3D-WM, ODCT3D, PRI-NLM3D, and BM4D filters. It is not possible to give objective measurement of the denoising quality because the ground-truth data is unknown; however, from a subjective point of view, we note that the visual quality of the restored phantom has been significantly improved by every algorithm, as the noise has been removed without introducing disturbing artifacts. Given the relatively mild standard deviation of the corrupting noise, all algorithms produce good-quality estimates, nevertheless we note that fine details in the phantoms restored by OB-NLM3D and OB-NLM3D-WM are slightly over-smoothed whereas the estimates obtained from ODCT3D, PRI-NLM3D, and BM4D have comparable visual quality.

D. Computational Complexity and Scalability

The current single-threaded MATLAB/C implementation of the BM4D algorithm under the modified profile requires about 11 minutes to denoise the BrainWeb phantom on a machine with a 2.66-GHz processor and 8GB of RAM. About 30% of the computation time is spent during the hard-thresholding stage, and the remaining is spent during the Wiener-filtering stage. We remark that the cube-matching nonlocal search procedure, mainly parametrized by the size of the 3-D search window N_S and by the step between neighboring processed cubes N_{step}, is by far the most time-consuming task. In our current implementation only the 1-D transform applied to the fourth (grouping) dimension uses a fast algorithm, whereas the 3-D separable transform used for each cube is computed via matrix multiplications; therefore BM4D could be accelerated by employing fast transform algorithms also for the cube dimensions. Table IV shows the PSNR performance, together with the execution times, of BM4D tuned with different combinations of N_S and N_{step}.

Significant accelerations can be induced by decreasing N_S. In

TABLE IV

PSNR DENOISING PERFORMANCES OF BM4D TUNED WITH DIFFERENT COMBINATIONS OF THE PARAMETERS CONTROLLING THE CUBE-MATCHING, NAMELY THE SIZE OF THE 3-D SEARCH WINDOW N_S AND THE STEP BETWEEN NEIGHBORING PROCESSED CUBES N_{STEP}; THE LAST COLUMN SHOWS THE MEAN EXECUTION TIMES OF THE DENOISING PROVIDED BY A SINGLE-THREADED MATLAB/C IMPLEMENTATION. THE HARDWARE USED TO EXECUTE THE EXPERIMENTS IS A MACHINE WITH A 2.66-GHZ PROCESSOR AND 8GB OF RAM. THE TEST DATA IS THE BRAINWEB PHANTOM, CORRUPTED BY I.I.D. GAUSSIAN NOISE WITH STANDARD DEVIATIONS σ. THE PERFORMANCES OF BM4D UNDER THE DEFAULT SETTINGS $N_S = 11$ AND $N_{\text{STEP}} = 3$ ARE REPORTED IN ITALIC FONT.

Param.		σ				Sec.
N_S	N_{step}	7%	11%	15%	19%	
1	5	27.71	24.39	22.08	20.31	4.0
	4	30.99	28.57	26.93	25.70	6.2
	3	31.82	29.58	28.10	27.00	13.6
3	5	32.81	30.51	28.90	27.66	49.7
	4	33.36	31.13	29.57	28.37	91.2
	3	33.54	31.31	29.76	28.57	210.5
5	5	33.68	31.58	30.13	29.00	107.8
	4	33.95	31.85	30.41	29.30	204.9
	3	34.05	31.97	30.53	29.42	455.8
7	5	33.90	31.81	30.36	29.24	118.5
	4	34.17	32.08	30.63	29.51	228.5
	3	34.26	32.18	30.74	29.63	524.1
9	5	33.98	31.89	30.42	29.27	139.5
	4	34.24	32.13	30.68	29.55	253.5
	3	34.34	32.25	30.80	29.68	604.3
11	5	34.00	31.86	30.37	29.21	155.1
	4	34.27	32.17	30.69	29.56	289.8
	3	*34.38*	*32.28*	***30.83***	***29.70***	*676.7*
13	5	34.01	31.84	30.34	29.16	199.1
	4	34.30	32.18	30.70	29.55	372.7
	3	34.40	**32.30**	**30.83**	**29.70**	870.5
15	5	34.03	31.86	30.34	29.15	257.7
	4	34.31	32.18	30.69	29.53	482.5
	3	**34.42**	**32.30**	30.82	29.68	1130.1

fact, referring to Table IV, the setting $N_S = 1$ is roughly between $50\times$ and $150\times$ faster than the default size $N_S = 11$. However, $N_S = 1$ *de facto* disables the grouping procedure, because in such case the search windows, and consequently the groups, contain one and only one element, that is the reference cube itself. As a result, the sparsification induced by the collaborative filtering is less effective because the nonlocal correlation is missing in the grouped data. The repercussions are evident in the corresponding PSNR performance, which is about up to 5dB worse than those of the default case. In general, whenever N_S is enlarged and N_{step} does not vary, the execution time grows by roughly a factor of $1.2\times$ without producing a dramatic PSNR improvement. Interestingly, the PSNR sometimes worsen as $N_S \geq 11$, thus suggesting that bigger search windows do not always improve the denoising quality.

Conversely, keeping N_S fixed, and excluding the case limit $N_S = 1$, we observe that the execution time roughly halves at every increment of N_{step} with a performance degradation of only about 0.4dB. Anyway the step should not be carelessly enlarged because whenever $N_{\text{step}} > L$ any pair of adjacent reference cubes are separated by a gap of $L - N_{\text{step}}$ voxels in each dimension, and since there is no guarantee that every voxel in those gaps will be covered by non-reference cubes, the final denoised volume may contain missing estimates. In the experiments reported in Table IV, we substitute the occurring missing estimates with the corresponding values of the data used in the grouping, i.e. the z in the hard-thresholding stage and $\hat{y}^{\text{ht}}$ in Wiener-filtering stage.

In conclusion, we have verified that BM4D gracefully scale with different tuning of the search-window size N_S and the step N_{step} parameters, which in turn affect the complexity of the cube-matching search procedure. However, optimal filtering results are achieved when $N_S > 3$ and $N_{\text{step}} \leq L$, to enable a better grouping and avoid possible missing estimates in the final denoised volume.

IV. ITERATIVE RECONSTRUCTION FROM INCOMPLETE MEASUREMENTS

In several inverse imaging applications, such as magnetic resonance imaging (MRI), the observed (acquired) measurements are a severe subsample of a transform-domain representation of the original unknown signal. In this section, we propose an iterative procedure, designed for the joint denoising and reconstruction of incomplete volumetric data, that uses the proposed BM4D algorithm as a regularizer operator.

A. Problem Setting

In volumetric reconstruction, an unknown signal of interest is observed through a limited number linear functionals. In compressed-sensing problems, these observations can be considered as a limited portion of the spectrum of the signal in transform domain. In general, a direct application of an inverse operator cannot reconstruct the original signal, because we consider cases where the available data is much smaller than what is required according to the Nyquist-Shannon sampling theorem. However, it is shown that whenever the signal can be represented sparsely in a suitable transform domain, stable (and even exact) reconstruction of the unknown signal is still possible [19], [20]. The most popular reconstruction techniques are formulated as a convex optimization, usually solved by mathematical programming algorithms, that yields the solution most consistent with the available data. The optimization is typically constrained by a penalty term expressed as ℓ_0 or ℓ_1 norms, which are exploited to enable the sparsity of the assumed image priors [21], [22], [23], [20]. Our approach, inspired by [5], [6], [24], replaces such parametric

modeling of the solution with a nonparametric one implemented by the use of a spatially adaptive denoising filter.

In MRI the non-uniform coil sensitivity and inhomogeneities of the magnetic field, causing frequency shifts and distortions in both intensity and geometry of the acquired data, generate (complex) images with a non-zero phase component [31], [32], [33]. It is generally assumed that the magnitude contains most of the structural information of the underlying data and the phase is smooth varying [25], [26], [27], [28]. Thus, even though the real and imaginary parts could be processed simultaneously, e.g., enforcing smoothness priors on the complex representation of the image, in our approach the magnitude and phase of the data are independently regularized in order to preserve their unique and individual features.

B. Observation Model

The observation model for the volumetric reconstruction problem is given by

$$\theta = \mathcal{T}\left(y e^{\imath \phi}\right) + \eta, \tag{15}$$

where θ is the transform-domain representations of the unknown volumetric data having magnitude $y : X \to \mathbb{R}^+$ and absolute (unwrapped) phase $\phi : X \subset \mathbb{Z}^3 \to \mathbb{R}$, $\imath$ is the imaginary unit, $\mathcal{T}$ is, for our purposes, the Fourier transform, and $\eta(\cdot) \sim \mathcal{N}\left(0, \sigma^2\right)$ is i.i.d. complex Gaussian noise with zero mean and standard deviation σ.

Let Ω be the support of the available portion of the spectrum θ. We define a sampling operator S as the characteristic (indicator) function χ_Ω, which is 1 over Ω and 0 elsewhere. By means of S, we can split the spectrum in two complementary parts as

$$\theta = \underbrace{S \cdot \theta}_{\theta_1} + \underbrace{(1 - S) \cdot \theta}_{\theta_2},$$

where θ_1 and θ_2 are the observed (known) and unobserved (unknown) portion of the spectrum θ, respectively. Our goal is to recover an estimate $\tilde{y}$ of the unknown underlying magnitude y from the observed noisy measurements θ_1. Note that if we had the complete spectrum θ, we could trivially obtain $\tilde{y}$ by applying a volumetric denoising filter, such as BM4D, on the (exact) noisy magnitude $z = \left|\mathcal{T}^{-1}(\theta)\right|$. However, since only a small portion of the spectrum θ is available and since such portion contains noisy measurements, the reconstruction task of the magnitude y is an ill-posed problem.

In Section IV-C, we first introduce the algorithm in its more general form, suitable for data having non-zero phase. Then, in Section IV-D, we consider the simplifications to the algorithm that are relevant to the special case where the phase component is zero. In both cases, the ultimate goal consists in reconstructing the magnitude of the incomplete volumetric image.

C. Reconstruction of Volumetric Data with Non-Zero Phase

The reconstruction is carried out by an iterative procedure where the estimate of the unobserved spectrum θ_2 is improved via a stochastic search driven by the action of an adaptive denoising filter [5], [6], [24]. Specifically, we denote such filter as $\Phi(\cdot, \cdot)$ whose inputs are the (real) noisy data to be filtered and the assumed noise standard deviation of this data. In what follows, we consider Φ to be the BM4D filter.

At first, the estimate of the unobserved spectrum θ_2 is set to zero to generate the initial back-projection $\mathcal{T}^{-1}\left(\theta_1 + (1 - S) \cdot \mathbf{0}\right)$ which is then used to obtain the magnitude and phase components as

$$\hat{y}^{(0)} = \tilde{y}^{(0)} = \hat{y}_{\text{excite}}^{(0)} = \left|\mathcal{T}^{-1}\left(\theta_1 + (1 - S) \cdot \mathbf{0}\right)\right|,$$
$$\hat{\phi}^{(0)} = \tilde{\phi}^{(0)} = \hat{\phi}_{\text{excite}}^{(0)} = \angle\mathcal{T}^{-1}\left(\theta_1 + (1 - S) \cdot \mathbf{0}\right).$$

$$\hat{y}^{(0)} = \tilde{y}^{(0)} = \hat{y}^{(0)}_{\text{excite}} = \left| \mathcal{T}^{-1} \left(\theta_1 + (1-S) \cdot \mathbf{0} \right) \right| \qquad 1$$

$$\hat{\phi}^{(0)} = \tilde{\phi}^{(0)} = \hat{\phi}^{(0)}_{\text{excite}} = \angle \mathcal{T}^{-1} \left(\theta_1 + (1-S) \cdot \mathbf{0} \right) \qquad 2$$

$$k = 1 \qquad 3$$

$$\textbf{while} \ \ k \le k_{\text{final}} \qquad 4$$

$$\hat{\theta_2}^{(k)} = \mathcal{T} \left(\hat{y}^{(k-1)} e^{\tilde{\phi}^{(k-1)}} \right) \cdot (1-S) \qquad 5$$

$$\hat{\theta}^{(k)}_{\text{excite}} = \theta_1 + \hat{\theta_2}^{(k)} + (1-S) \cdot \eta^{(k)}_{\text{excite}} \qquad 6$$

$$\hat{y}^{(k)}_{\text{excite}} = \left| \mathcal{T}^{-1} \left(\hat{\theta}^{(k)}_{\text{excite}} \right) \right| \qquad 7$$

$$\hat{\phi}^{(k)}_{\text{excite}} = \angle \mathcal{T}^{-1} \left(\hat{\theta}^{(k)}_{\text{excite}} \right) \qquad 8$$

$$\hat{y}^{(k)} = \text{VST}^{-1} \left(\Phi \left(\text{VST} \left(\hat{y}^{(k)}_{\text{excite}}, \sigma^{(k)}_{\text{excite}} \right), \sigma_{\text{VST}} \right), \sigma^{(k)}_{\text{excite}} \right) \qquad 9$$

$$\hat{\phi}^{(k)} = \text{mod} \left(\Phi \left(\text{mod} \left(\hat{\phi}^{(k)}_{\text{excite}} + \zeta^{(k)}, (-\pi, \pi] \right), \sigma^{(k)}_{\text{excite}} \right) - \zeta^{(k)}, (-\pi, \pi] \right) \quad 10$$

$$\lambda_k = \left(\lambda^{-1}_{k-1} \sigma^{(k-1)^{-2}}_{\text{excite}} + \sigma^{(k)^{-2}}_{\text{excite}} \right)^{-1} \sigma^{(k)^{-2}}_{\text{excite}} \qquad 11$$

$$\tilde{y}^{(k)} e^{\imath \tilde{\phi}^{(k)}} = \lambda_k \tilde{y}^{(k-1)} e^{\imath \tilde{\phi}^{(k-1)}} + (1-\lambda_k) \hat{y}^{(k)} e^{\imath \hat{\phi}^{(k)}} \qquad 12$$

$$k \leftarrow k+1 \qquad 13$$

$$\textbf{end} \qquad 14$$

Algorithm 1. Pseudo-code of the iterative reconstruction algorithm. The input parameters are the available spectrum θ_1, the 3-D trajectory S, the excitation noise η_{excite}, and the number of iterations k_{final}. By Φ we denote the denoising algorithm used during the reconstruction, and VST is a variance-stabilization transformation for Rician-distributed data.

Subsequently, for each iteration $k \ge 1$, which we shall denote by a superscript (k), the reconstruction is carried out through three cascading steps:

1) *Noise Addition (Excitation):* The estimate of the unobserved portion of the spectrum is first extracted as

$$\hat{\theta_2}^{(k)} = \mathcal{T} \left(\hat{y}^{(k-1)} e^{\tilde{\phi}^{(k-1)}} \right) \cdot S, \qquad (16)$$

where $\hat{y}^{(k-1)}$ and $\tilde{\phi}^{(k-1)}$ are the denoised magnitude and regularized phase produced in the previous iteration $(k-1)$. Subsequently, we synthetically generate the excited spectrum

$$\hat{\theta}^{(k)}_{\text{excite}} = \theta_1 + \hat{\theta_2}^{(k)} + (1-S) \cdot \eta^{(k)}_{\text{excite}}, \qquad (17)$$

by injecting (16) with i.i.d. complex Gaussian noise $\eta^{(k)}_{\text{excite}}$ with zero mean and standard deviation $\sigma^{(k)}_{\text{excite}}$. Eventually, the volumetric (excited) magnitude

$$\hat{y}^{(k)}_{\text{excite}} = \left| \mathcal{T}^{-1} \left(\hat{\theta}^{(k)}_{\text{excite}} \right) \right| \qquad (18)$$

and (excited) phase

$$\hat{\phi}^{(k)}_{\text{excite}} = \angle \mathcal{T}^{-1} \left(\hat{\theta}^{(k)}_{\text{excite}} \right) \qquad (19)$$

are obtained by extracting the absolute value (modulus) and angle from the inverse-transformed spectrum (17), respectively.

2) *Volumetric Filtering:* The missing coefficients of the spectrum θ, previously excited in (17), are then modified by the action of the independent denoising of the excited magnitude (18) and excited phase (19). Intuitively, whenever the excited coefficients correspond to features that satisfy the sparsification induced by the grouping and collaborative filtering, these features will be preserved or enhanced, otherwise they will be attenuated.

The excited magnitude (18) is distributed accordingly to the Rician observation model as in (13) because the noise in the corresponding excited spectrum (17) is i.i.d. complex Gaussian. Thus, we need to apply a variance-stabilization transform (VST), analogously to (14), during the filtering of (18) as

$$\hat{y}^{(k)} = \text{VST}^{-1} \left(\Phi \left(\text{VST} \left(\hat{y}^{(k)}_{\text{excite}}, \sigma^{(k)}_{\text{excite}} \right), \sigma_{\text{VST}} \right), \sigma^{(k)}_{\text{excite}} \right),$$

where $\sigma^{(k)}_{\text{excite}}$ is the standard deviation of the excitation noise added in (17).

On the other hand, for the sake of simplicity, the phase is assumed to follow the Gaussian observation model (1) with noise standard deviation $\sigma^{(k)}_{\text{excite}}$. To ensure proper filtering, in particular along phase-jumps, we add before denoising and then subtract after denoising a random phase shift $\zeta^{(k)}$ as

$$\hat{\phi}^{(k)} = \text{mod} \left(\Phi \left(\text{mod} \left(\hat{\phi}^{(k)}_{\text{excite}} + \zeta^{(k)}, (-\pi, \pi] \right), \sigma^{(k)}_{\text{excite}} \right) - \zeta^{(k)}, (-\pi, \pi] \right),$$

where $\zeta^{(k)} \sim \mathcal{U}(-\pi, \pi)$ is a random variable uniformly distributed between $-\pi$ and π defining the phase shift applied to every voxel of $\hat{\phi}^{(k)}_{\text{excite}}$, and $\text{mod}(\cdot, (-\pi, \pi])$ realizes the wrapping on the interval $(-\pi, \pi]$. Such phase-shift moves the position of the phase jump at different spatial positions at each instance of filtering and in this way $\tilde{\phi}^{(k)}$ eventually approximates, modulo 2π, the result of filtering the absolute unwrapped phase.

3) *Data Reconstruction:* The sequence of estimates $\hat{y}^{(k)}$ might get trapped in local optima because the data that pilots the regularization, i.e. the available spectrum θ_1, is corrupted by noise. Thus, in order to escape from possible degenerate solutions, we aggregate the estimates $\hat{y}^{(k)}$ and $\hat{\phi}^{(k)}$ in a complex recursive convex combination as

$$\tilde{y}^{(k)} e^{\imath \tilde{\phi}^{(k)}} = \lambda_k \tilde{y}^{(k-1)} e^{\imath \tilde{\phi}^{(k-1)}} + (1-\lambda_k) \hat{y}^{(k)} e^{\imath \hat{\phi}^{(k)}}, \quad (20)$$

where $\tilde{y}^{(k)} >= 0$, and $-\pi < \tilde{\phi}^{(k)} \le \pi$ for all $k \ge 0$. The aggregation weights $0 \le \lambda_k \le 1$ are recursively defined as

$$\lambda_k = \left(\lambda^{-1}_{k-1} \sigma^{(k-1)^{-2}}_{\text{excite}} + \sigma^{(k)^{-2}}_{\text{excite}} \right)^{-1} \sigma^{(k)^{-2}}_{\text{excite}}, \qquad (21)$$

with initial condition $\lambda_0 = 1$. The explicit formulae for (20)

$$\tilde{y}^{(k)} e^{\imath \tilde{\phi}^{(k)}} = \left(\sum_{i=0}^{k} \sigma^{(i)^{-2}}_{\text{excite}} \right)^{-1} \sum_{i=0}^{k} \sigma^{(i)^{-2}}_{\text{excite}} \hat{y}^{(i)} e^{\imath \hat{\phi}^{(i)}},$$

and for (21)

$$\lambda_k = \left(\sum_{i=0}^{k} \sigma^{(i)^{-2}}_{\text{excite}} \right)^{-1} \sigma^{(k)^{-2}}_{\text{excite}},$$

illustrate that each estimate $\hat{y}^{(i)}$ contributes to the combination (20) with a weight inversely proportional to the variance $\sigma^{(i)^2}_{\text{excite}}$ of its excitation noise.

The iterative procedure can be either stopped after a pre-specified number of iterations k_{final}, or when two magnitude estimates produced at subsequent iterations do not significantly differ from each other. For instance, this can be done via the normalized p-norm as

$$|X|^{-\frac{1}{p}} \cdot \left\| \tilde{y}^{(k)} - \tilde{y}^{(k-1)} \right\|_p \le \varepsilon,$$

where $|X|$ is the cardinality of the domain X, and $\varepsilon \in \mathbb{R}^+$ is the desired tolerance value. The pseudo-code of the iterative procedure is shown in Algorithm 1.

To illustrate the role of the two separate recursive volumetric estimates $\hat{y}^{(k)}$ and $\tilde{y}^{(k)}$, let us assume that $\Omega \subsetneq X$ and that $\sigma^{(k)}_{\text{excite}} \to \sigma$. There are essentially two cases. First, if $\sigma > 0$, the system is kept permanently under excitation, which means that in practice $\hat{y}^{(k)} e^{\imath \hat{\phi}^{(k)}}$ is not able to converge. However, under the same assumptions, we have that $\lambda_k \approx k^{-1}$ for large k, and thus $\tilde{y}^{(k)} e^{\imath \tilde{\phi}^{(k)}}$ approaches the sample mean of $\hat{y}^{(k)} e^{\imath \hat{\phi}^{(k)}}$ over k. Thus, $\tilde{y}^{(k)} e^{\imath \tilde{\phi}^{(k)}}$ can be interpreted as an approximation of the expectation of $\hat{y}^{(k)} e^{\imath \hat{\phi}^{(k)}}$ over k (i.e. over the excitation noise). Second, if $\sigma = 0$, then $\hat{y}^{(k)} e^{\imath \hat{\phi}^{(k)}}$ can converge to some estimate $\hat{y} e^{\imath \hat{\phi}}$ and $\tilde{y}^{(k)} e^{\imath \tilde{\phi}^{(k)}}$

will eventually converge to the same estimate. In summary, in the ideal case where the observed spectrum θ_1 is noise-free, the two estimates $\tilde{y}^{(k)}e^{\imath\tilde{\phi}^{(k)}}$ and $\hat{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ become equivalent; conversely, when observed spectrum is noisy, $\tilde{y}^{(k)}e^{\imath\tilde{\phi}^{(k)}}$ plays a crucial role in enabling convergence to the expectation of the non-convergent $\hat{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$.

Even though in principle, for an arbitrary operator Φ, the existence of the expectation of $\hat{y}^{(k)}$ can be guaranteed only if the excitation noise vanishes sufficiently fast with k, we note that in practice, due to the denoising and to the given observations θ_1, such expectation is typically well defined, leading to a stable convergence of $\tilde{y}^{(k)}$.

We observe also that if the spectrum θ of the noisy phantom is completely available (i.e. $\theta_1 = \theta$, $\Omega = X$, and thus no subsampling is performed) and $\sigma_{\text{excite}}^{(k)} = \sigma$ for all k, Algorithm 1 coincides with a one-time application of the filter Φ on $\hat{y}_{\text{excite}}^{(0)} = \left|\mathcal{T}^{-1}(\theta)\right|$ with assumed noise standard deviation σ, because the inputs $y_{\text{excite}}^{(k)}$ of each iteration do not vary with k. On the other hand, if the whole spectrum is not available (i.e. $\Omega \subsetneq X$) and $\sigma_{\text{excite}}^{(k)} \to \sigma = 0$, as observed above we have that $\tilde{y}^{(k)}e^{\imath\tilde{\phi}^{(k)}}$ approaches $\hat{y}_{\text{excite}}^{(k)}e^{\imath\hat{\phi}^{(k)}}$. Thus, Algorithm 1 generalizes both the iterative reconstruction algorithm implemented in [5], [6] to the case of noisy observations, as well as the BM4D filter to the case of incomplete measurements.

D. Reconstruction of Volumetric Data with Zero Phase

In this section we discuss the reconstruction of volumetric data under the assumption that its phase component is null, i.e. $\phi = 0$. Since in such case the magnitude $\left|ye^{\imath\phi}\right|$ is equal to the real component $\text{Re}(ye^{\imath\phi}) = y$, the reconstruction procedure described in the previous section can be greatly simplified.

Initially, we set the initial estimate of the missing portion of the spectrum to zero, then we extract the back-projection as

$$\hat{y}_{\text{excite}}^{(0)} = \text{Re}\left(\mathcal{T}^{-1}(\theta_1 + (1 - S) \cdot \mathbf{0})\right).$$

Note that the extraction of the absolute value is no longer needed because the underlying data y is real; however since the output of $\mathcal{T}^{-1}$ is in general complex due to the noise in the data or numerical errors of the computation, we still need to extract the real component after the inverse transformation because the denoising filter Φ is implemented for real inputs.

Subsequently, for each iteration $k > 1$, the following steps are performed:

1) *Noise Addition (Excitation):* The estimated unobserved part $\hat{\theta}_2^{(k)}$ of the spectrum is excited to produce the excited spectrum

$$\hat{\theta}_{\text{excite}}^{(k)} = \theta_1 + \hat{\theta}_2^{(k)} + (1 - S) \cdot \eta_{\text{excite}}^{(k)}, \qquad (22)$$

where $\eta_{\text{excite}}^{(k)}$ is again i.i.d. complex Gaussian noise with zero mean and standard deviation $\sigma_{\text{excite}}^{(k)}$. Then, the (spatial-domain) excited volumetric data is obtained by taking the real part of the inverse transformation $\mathcal{T}^{-1}$ applied to the excited spectrum (22) as

$$\hat{y}_{\text{excite}}^{(k)} = \text{Re}\left(\mathcal{T}^{-1}\left(\hat{\theta}_{\text{excite}}^{(k)}\right)\right). \qquad (23)$$

2) *Volumetric Filtering:* The volumetric excited data (23) is denoised by the filter Φ as

$$\hat{y}^{(k)} = \Phi\left(\hat{y}_{\text{excite}}^{(k)}, \sigma_{\text{excite}}^{(k)}\right), \qquad (24)$$

being $\sigma_{\text{excite}}^{(k)}$ is the standard deviation of the excitation noise in (22). Observe that the application of the VST is no longer needed because (23) takes the real part and not the modulus of $\mathcal{T}^{-1}\left(\hat{\theta}_{\text{excite}}^{(k)}\right)$, and thus its excited observation model agrees with (1).
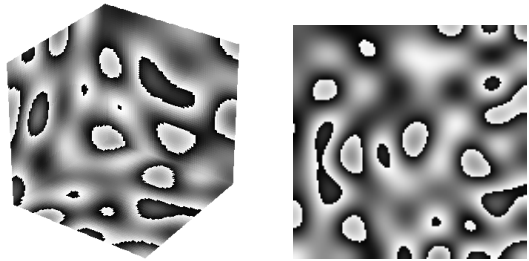

Fig. 3. Original phase ϕ used for the reconstruction experiments (black and white correspond to $-\pi$ and π, respectively).

3) *Data Reconstruction:* The volumetric reconstruction is eventually produced by the convex combination

$$\tilde{y}^{(k)} = \lambda_k \tilde{y}^{(k-1)} + (1 - \lambda_k)\hat{y}^{(k)}, \qquad (25)$$

whose weights λ_k are defined as in (21). Observe that, (25) is the particular case of (20) obtained by setting to zero every phase estimate $\hat{\phi}^{(k)}$.

V. VOLUMETRIC RECONSTRUCTION EXPERIMENTS

We show the reconstruction results of the iterative procedure described in Section IV, recalling that BM4D is used in place of the generic volumetric filter Φ. The parameters of the filter are the same reported in Section III-A, but only the hard-thresholding stage is performed during the reconstruction.

As already said, the excitation noise $\eta_{\text{excite}}^{(k)}$ is chosen to be i.i.d. complex Gaussian noise with zero mean and variance

$$\sigma_{\text{excite}}^{(k)} = \alpha^{-k-\beta} + \sigma \qquad (26)$$

where $\alpha > 0$ and $\beta > 0$ are parameters chosen so that the excitation noise lessens as the iterations increase, and σ is the standard deviation of the noise η in (15). The variance (26) (exponentially) decreases in order to diminish the aggressiveness of the filtering as the iterations increase. Moreover, the additive term σ ensures that the excitation noise level in (16) converges to the initial noise level in (15). In this manner, the noise standard deviation assumed by the denoising filter is never smaller than that of the noise corrupting the observed measurements.

In our experiments we consider volumetric data having either zero or non-zero phase ϕ. We synthetically generate ϕ by first applying a low-pass filter to a 3-D i.i.d. zero-mean Gaussian field, and then wrapping the result to the interval $(-\pi, \pi]$. Fig. 3 illustrates the so-obtained phase ϕ. Note that the sharp variations from black to white correspond to phase jumps from $-\pi$ to π.

Considerable freedom is given for the design of the 3-D sampling operator S, which can be either a multi-slice stack of identical 2-D trajectories, or a single 3-D sampling trajectory. In the former case the measurements are taken as a multi-slice stack of 2-D cross-sections transformed in Fourier (k-space) domain, each of which undergo the sampling induced by the corresponding 2-D trajectory of S. In the latter case, the observation is directly sampled in 3-D Fourier transform domain. The sampling trajectories are in general classified as Cartesian and non-Cartesian. Cartesian trajectories are extremely popular as they are less susceptible to system imperfections, and the relative reconstruction task is simple. On the other hand, non-Cartesian trajectories usually require more complicated reconstruction algorithms, but they allow for a higher under-sampling and faster acquisition times [29]. For these reasons, in our experiments we use the non-Cartesian trajectories *Radial*, *Spiral*, *Logarithmic Spiral*, *Limited Angle* and *Spherical*. Examples of such trajectories are
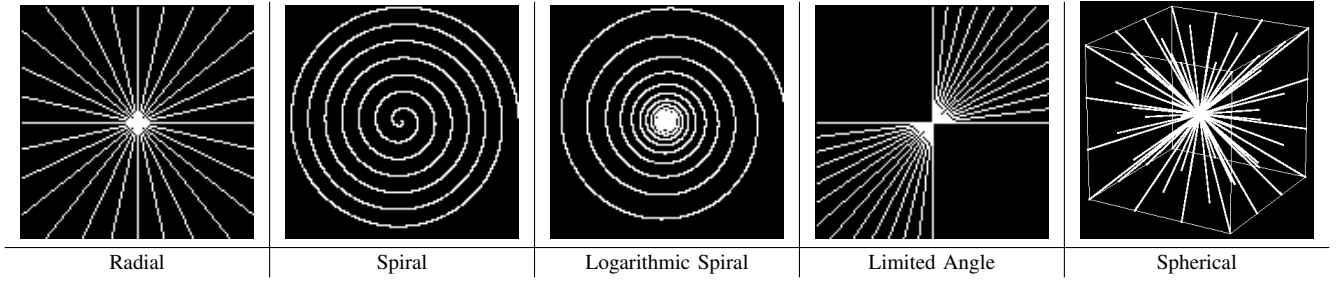
Fig. 4. Examples of different sampling trajectories. These trajectories define which k-space coefficients will be retained during the MR acquisition process.
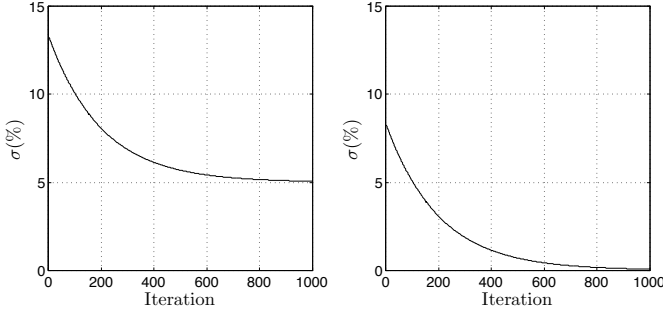

Fig. 5. Standard deviation σ_{excite} of the excitation noise (26) for noisy (left) and noise-free (right) data of parameters $\alpha = 1.01$, $\beta = 500$, and $\sigma = 5\%$.

illustrated in Fig. 4. The rationale behind these settings is to simulate the acquisition process of the most common medical imaging applications [29].

The metrics used to measure the performance of the reconstruction are again the PSNR and SSIM. We present the reconstruction performance after $k_{\text{final}} = 1000$ iterations from a set of incomplete noisy or noise-free k-space measurements. We also consider data having both zero and non-zero initial phase. The trajectories have sampling ratio $|\Omega||X|^{-1} = 30\%$, where $|\Omega|$ is the cardinality of the sampled voxels and $|X|$ is the total number of voxels in the phantom. The parameters of the excitation noise (26) are $\alpha = 1.01$ and $\beta = 500$, for all experiments. Even though in principle different sampling strategies could benefit from different excitation profiles, we use a fixed setting for α and β to enable a more direct comparison between the various experiments. Finally, we set the standard deviation of the noise in the observed measurements as $\sigma = 5\%$. Fig. 5 illustrates (26) used for the noisy (left) and noise-free (right) case. The test data of our experiment is the BrainWeb and 3-D Shepp-Logan phantom of size $128 \times 128 \times 128$ voxels; cross-sections of both original phantoms are shown in Fig. 7(b) and Fig. 7(a), respectively. The Shepp-Logan is widely used in medical imaging [13], [34], [14] but, being a piecewise constant signal, it admits a very sparse representation in transform domain which can in turn ease the reconstruction task. Thus, we also perform the reconstruction experiments on the more challenging BrainWeb phantom, as it is a more realistic model of MR data.

Fig. 6 gives a deeper insight on the PSNR progression with respect to the number of iterations. We first notice that, in every experiment, the reconstruction algorithm is able to substantially ameliorate the initial back-projections in terms of both objective and subjective visual quality. We observe that in many cases, particularly those where $\sigma = 0$, the PSNR grows almost linearly, in accordance with the exponential decay of the standard deviation of the excitation noise. Fig. 6 also empirically demonstrates that the ratio between the PSNR of $\tilde{y}^{(k)}$ and $\hat{y}^{(k)}$ approaches one, as motivated in Section IV-C.

The PSNR and SSIM performance of the reconstruction is reported

TABLE V
PSNR (LEFT VALUE IN EACH CELL) AND SSIM [15], [8] (RIGHT VALUE IN EACH CELL) RECONSTRUCTION PERFORMANCES AFTER $k_{\text{FINAL}} = 1000$ ITERATIONS OF THE BRAINWEB AND THE SHEPP-LOGAN PHANTOM OF SIZE $128 \times 128 \times 128$ VOXELS. THE TESTS ARE MADE ON BOTH NOISY ($\sigma = 5\%$) AND NOISE-FREE MEASUREMENTS, HAVING SAMPLING RATIO 30%.

Traj.	Data	Zero phase		Non-zero phase	
		$\sigma = 0\%$	$\sigma = 5\%$	$\sigma = 0\%$	$\sigma = 5\%$
Radial	*BrainWeb*	37.22|0.97	31.00|0.91	41.00|0.99	30.57|0.91
	Shepp-Log.	77.01|1.00	31.82|0.98	70.12|1.00	32.03|0.98
Spiral	*BrainWeb*	34.75|0.96	19.60|0.66	16.75|0.48	21.99|0.74
	Shepp-Log.	58.23|1.00	21.22|0.55	24.27|0.65	26.22|0.92
Log. Sp.	*BrainWeb*	40.92|0.99	31.83|0.92	41.89|0.99	31.20|0.92
	Shepp-Log.	77.51|1.00	32.04|0.98	69.36|1.00	31.91|0.98
Lim. An.	*BrainWeb*	32.48|0.94	27.17|0.85	17.93|0.54	20.74|0.65
	Shepp-Log.	42.45|1.00	28.31|0.95	21.75|0.57	24.47|0.77
Spheric.	*BrainWeb*	41.67|0.99	32.46|0.93	42.99|0.99	31.88|0.93
	Shepp-Log.	77.85|1.00	31.72|0.98	62.56|1.00	31.50|0.98

in Table V. As one can see, the objective performance is almost always excellent; Additionally, the results for $\sigma = 5\%$ often approach those obtained in the denoising experiments reported in Table II, that correspond to the ideal conditions of complete sampling and zero phase. Interestingly, the reconstruction performance of the BrainWeb phantom under the *Spiral* and *Limited Angle* sampling are higher in the noisy case. In fact, as the ill-conditioning of the reconstruction problem increases, the best results can be achieved using excitation schedule η_{excite} characterized by larger values of standard deviation because a larger variance in the excitation noise leads to a stronger filtering and, consequently, a stronger regularization.

The visual appearance of the reconstructed BrainWeb and Shepp-Logan phantoms with non-zero phase and initial noise $\sigma = 5\%$ are shown in Fig. 9 and Fig. 10, respectively. Let us remark how the reconstruction is always able to improve significantly the visual appearance of the phantom, even in those cases when the image information of the initial back-projection is extremely limited and the phase is distorted by multiple erroneous jumps.

We stress that the sampling ratio $|\Omega||X|^{-1}$ is not a fair measure of the difficulty of the reconstruction task, because different trajectories having the same $|\Omega||X|^{-1}$ extract different coefficients from the Fourier domain. As a matter of fact, the energy of MR images is concentrated in the centre (DC term) of the k-space, thus trajectories such as *Spherical* having denser sampling near the DC term are more advantaged than others, such as *Spiral* or *Limited Angle*, not giving any preference for the central part of the spectrum. Such differences are clearly visible from the visual appearance of the back-projections shown in Fig. 9 and Fig. 10 and from the final objective reconstruction
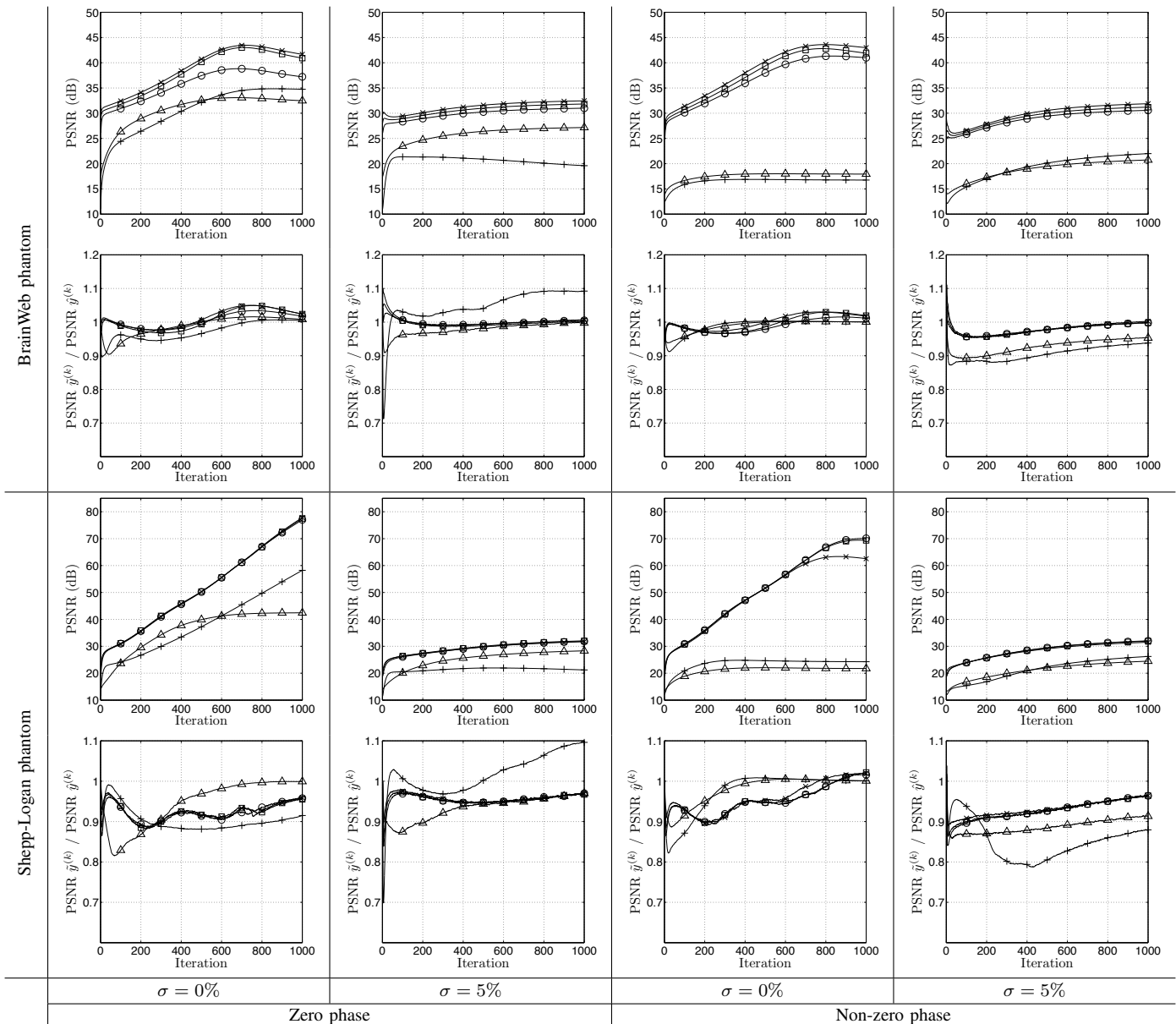
Fig. 6. PSNR progression for the iterative reconstruction of the noisy and noise-free BrainWeb having zero or non-zero phase. The plots in the top row illustrate the PSNR progressions of $\tilde{y}^{(k)}$, whereas the plots in the bottom row illustrate the progression of the ratio between the PSNR of $\tilde{y}^{(k)}$ and $\hat{y}^{(k)}$. The sampling trajectories are *Radial* (○), *Spiral* (+), *Logarithmic Spiral* (□), *Limited Angle* (△), and *Spherical* (×). The sampling ratio is in all cases 30%.

results reported in V, because, as expected, the worst objective and subjective reconstruction results are obtained under the *Spiral* or *Limited Angle* sampling, whereas the *Spherical* trajectory emerges as the best-performing sampling strategy. However, a significant drawback of the *Spherical* sampling is the higher scanning time required to complete the acquisition process.

VI. DISCUSSION AND CONCLUSIONS

A. Video vs. Volumetric Data Filtering

Both volumetric data and videos are defined over a 3-D domain. The first two dimensions always identify the width and the height of the data, but the connotation of the third dimension embodies completely different meanings. In the case of volumetric data the third dimension represents an additional spatial dimension (the depth), whereas in the case of videos it represents the temporal index along the the frame sequence (the time). We remark the importance of

designing algorithms that are able to leverage the specific connotation of the data to be filtered, i.e. the local spatial similarity in volumetric data and the motion information of videos.

To support our claim, we apply BM4D and the state-of-the-art video filter V-BM4D [35] to the BrainWeb phantom and the test videos *Tennis*, *Salesman*, *Flower Garden*, and *Miss America*. For all cases, the corrupting noise is i.i.d. Gaussian with zero mean and standard deviation $\sigma \in \{7\%, 11\%, 15\%, 19\%\}$. We recall that in V-BM4D mutually similar 3-D spatiotemporal volumes, built concatenating blocks along the direction defined by the motion vectors, are first grouped together and then jointly filtered in a 4-D transform domain [35]. Analogously, each cube in BM4D can be interpreted as a spatiotemporal volume built along null motion vectors, i.e. a sequence of blocks extracted from consecutive frames at the same spatial coordinate.

Table VI reports the PSNR and SSIM results of our tests. As

TABLE VI
PSNR (LEFT VALUE IN EACH CELL) AND SSIM [15], [8] (RIGHT VALUE
IN EACH CELL) DENOISING PERFORMANCES OF BM4D AND V-BM4D
[35] APPLIED TO THE BRAINWEB PHANTOM AND THE STANDARD VIDEO
TEST SEQUENCES *Tennis, Salesman, Flower Garden,* AND *Miss America*
CORRUPTED BY I.I.D. GAUSSIAN NOISE WITH DIFFERENT STANDARD
DEVIATION σ (%).

Data	Filter	σ			
		7%	11%	15%	19%
BrainWeb	BM4D	**34.38|0.95**	**32.28|0.92**	**30.82|0.90**	**29.70|0.87**
	V-BM4D	33.41|0.93	31.25|0.89	29.80|0.86	28.71|0.83
Tennis	BM4D	31.75|0.84	29.69|**0.78**	28.22|**0.73**	27.36|**0.70**
	V-BM4D	**32.00|0.85**	**29.88**|0.78	**28.56**|0.73	**27.59**|0.70
Salesm.	BM4D	**34.48|0.91**	**32.29|0.87**	**30.72|0.83**	**29.86|0.81**
	V-BM4D	34.28|0.90	32.01|0.85	30.50|0.81	29.38|0.78
Fl. Gard.	BM4D	28.42|**0.93**	25.90|0.88	22.96|0.81	22.37|0.77
	V-BM4D	**29.21|0.93**	**26.60|0.89**	**24.79|0.84**	**23.34|0.79**
Miss Am.	BM4D	**38.47|0.92**	**37.00|0.91**	**35.75|0.90**	**35.30|0.90**
	V-BM4D	38.13|**0.92**	36.57|0.90	35.37|0.88	34.40|0.86

expected, for volumetric data the PSNR performance of BM4D is consistently about 1dB higher than those of V-BM4D; conversely, as for video denoising, an interesting behavior occurs. We observe that the BM4D model is more effective whenever the corrupted video is characterized by low motion activity and the standard deviation of the noise is large. In fact, when the signal-to-noise ratio is very low, the motion estimation is likely to match the random patterns of the noise rather than the underlying structures to be tracked. For this reason, the zero-motion assumption, intrinsically enforced by BM4D, is an effective prior for the motion estimation of stationary videos, such as *Miss America* and *Salesman*, especially when σ is large. However, as motion activity gets higher, e.g., in *Tennis* and *Flower Garden*, V-BM4D clearly emerges as the best filtering paradigm.

B. Conclusions

The contributions of this work are twofold: first, we have introduced a powerful volumetric denoising algorithm, termed BM4D, which embeds the grouping and collaborative filtering paradigm; second, we have presented an iterative system for the reconstruction of incomplete volumetric data, enabled by the action of the aforementioned BM4D filter.

Experimental results on simulated brain phantom data show that the proposed BM4D filter significantly outperforms the current state of the art in volumetric data denoising. In particular, the denoising performance on MR images corrupted by either Gaussian- or Rician-distributed noise demonstrates the superiority of the proposed approach in terms of both objective (PSNR and SSIM) and subjective visual quality [4]. BM4D has been also successfully tested on the denoising of real MRI data, made publicly available by the OASIS database [12].

The viability of the volumetric reconstruction procedure has been tested using different volumetric phantoms measured in transform domain according to various sampling trajectories. The reconstruction has been evaluated using data with either zero or non-zero phase from incomplete, and possibly noisy, Fourier-domain (k-space) measurements. Experimental results on the Shepp-Logan and BrainWeb phantoms demonstrate the objective (PSNR and SSIM) and subjective effectiveness of the proposed method applied to under-sampled data.

Additional features, which can be embedded in BM4D, as is done for BM3D, include sharpening (α-rooting), non-white noise

removal (thus leading to a 3-D deblurring procedure as in [36]), and multichannel/multimodal filtering.

REFERENCES

[1] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, Washington, DC, USA, 2005, pp. 60–65.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, August 2007.

[3] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011.

[4] P. Milanfar, "A tour of modern image filtering," *Invited feature article to IEEE Signal Processing Magazine (preprint at http://users.soe.ucsc.edu/~milanfar/publications/)*, 2011.

[5] K. Egiazarian, A. Foi, and V. Katkovnik, "Compressed sensing image reconstruction via recursive spatially adaptive filtering," in *IEEE International Conference on Image Processing.*, vol. 1, October 2007, pp. 549–552.

[6] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Spatially adaptive filtering as regularization in inverse imaging: compressive sensing, upsampling, and super-resolution," in *Super-Resolution Imaging*. CRC Press / Taylor & Francis, 2010.

[7] R. Vincent, "Brainweb: Simulated brain database," http://mouldy.bic.mni.mcgill.ca/brainweb/, 2006.

[8] J. V. Manjón, P. Coupé, A. Buades, D. L. Collins, and M. Robles, "New methods for MRI denoising based on sparseness and self-similarity," *Medical Image Analysis*, vol. 16, no. 1, pp. 18–27, 2012.

[9] J. V. Manjón, P. Coupé, L. Martí-Bonmatí, D. L. Collins, and M. Robles, "Adaptive non-local means denoising of MR images with spatially varying noise levels," *Journal of Magnetic Resonance Imaging*, vol. 31, pp. 192–203, 2010.

[10] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images," *IEEE Transactions on Medical Imaging*, vol. 27, no. 4, pp. 425–441, April 2008.

[11] P. Coupé, P. Hellier, S. Prima, C. Kervrann, and C. Barillot, "3D wavelet subbands mixing for image denoising," *Journal of Biomedical Imaging*, pp. 1–11, January 2008.

[12] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults," *Journal of Cognitive Neuroscience*, vol. 22, no. 12, pp. 2677–2684, 2010. [Online]. Available: http://www.oasis-brains.org/

[13] L. Shepp and B. Logan, "The fourier reconstruction of a head section," *IEEE Transaction on Nuclear Science*, vol. 21, pp. 21–34, 1974.

[14] M. Schabel, "3D Shepp-Logan phantom," http://www.mathworks.com/matlabcentral/fileexchange/9416-3d-shepp-logan-phantom, 2006.

[15] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[16] A. Foi, "Noise estimation and removal in MR imaging: the variance-stabilization approach," in *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, Chicago, IL, USA, 2011.

[17] Y. Hou, C. Zhao, D. Yang, and Y. Cheng, "Comment on "Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering"," *IEEE Transaction on Image Processing*, July 2010.

[18] N. Wiest-Daesslé, S. Prima, P. Coupé, S. P. Morrissey, and C. Barillot, "Rician noise removal by non-local means filtering for low signal-to-noise ratio MRI: Applications to DT-MRI," in *Proceedings of the 11th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2008, pp. 171–179.

[19] E. Candes, 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, February 2006.

[20] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[21] M. Lustig and J. M. Pauly, "SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space," *Magnetic Resonance in Medicine*, vol. 64, no. 2, pp. 457–471, 2010.

[22] M. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso)," *IEEE Transaction on Information Theory*, vol. 55, no. 5, pp. 2183–2202, May 2009.

[23] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, December 2007.

[24] H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003. [Online]. Available: http://dx.doi.org/10.1007/b97441

[25] F. Zhao, D. Noll, J.-F. Nielsen, and J. Fessler, "Separate magnitude and phase regularization via compressed sensing," *submitted to IEEE Transactions on Medical Imaging*, 2011.

[26] J. Fessler and D. Noll, "Iterative image reconstruction in MRI with separate magnitude and phase regularization," in *IEEE International Symposium on Biomedical Imaging: Nano to Macro.*, vol. 1, April 2004, pp. 209–212.

[27] A. Funai, J. Fessler, D. Yeo, V. Olafsson, and D. Noll, "Regularized field map estimation in MRI," *IEEE Transactions on Medical Imaging*, vol. 27, no. 10, pp. 1484–1494, October 2008.

[28] M. Zibetti and A. De Pierro, "Separate magnitude and phase regularization in MRI with incomplete data: Preliminary results," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, April 2010, pp. 736–739.

[29] M. Lustig, D. Donoho, J. Santos, and J. Pauly, "Compressed sensing MRI," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, March 2008.

[30] J. Fessler, "Model-based image reconstruction for MRI," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 81–89, July 2010.

[31] G. Wright, "Magnetic resonance imaging," *IEEE Signal Processing Magazine*, vol. 14, no. 1, pp. 56–66, January 1997.

[32] Z. P. Liang and P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley-IEEE Press, October 1999.

[33] M. E. Haacke, R. W. Brown, M. R. Thompson, and R. Venkatesan, *Magnetic resonance imaging : physical principles and sequence design*, 1st ed. Wiley, June 1999.

[34] H. Gach, C. Tanase, and F. Boada, "2D & 3D Shepp-Logan phantom standards for MRI," in *19th International Conference on Systems Engineering (ICSENG)*, August 2008, pp. 521–526.

[35] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4D nonlocal spatiotemporal transforms," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers Electronic Imaging (SPIE)*, vol. 7870, January 2011.

[36] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," in *Proceedings of the Society of Photo-Optical Instrumentation Engineers Electronic Imaging (SPIE)*, vol. 6812-07, January 2008.

(a) 3-D Shepp-Logan phantom [13], [14].

(b) BrainWeb phantom [7].

(c) Noisy BrainWeb phantom (Gaussian noise $\sigma = 15\%$).
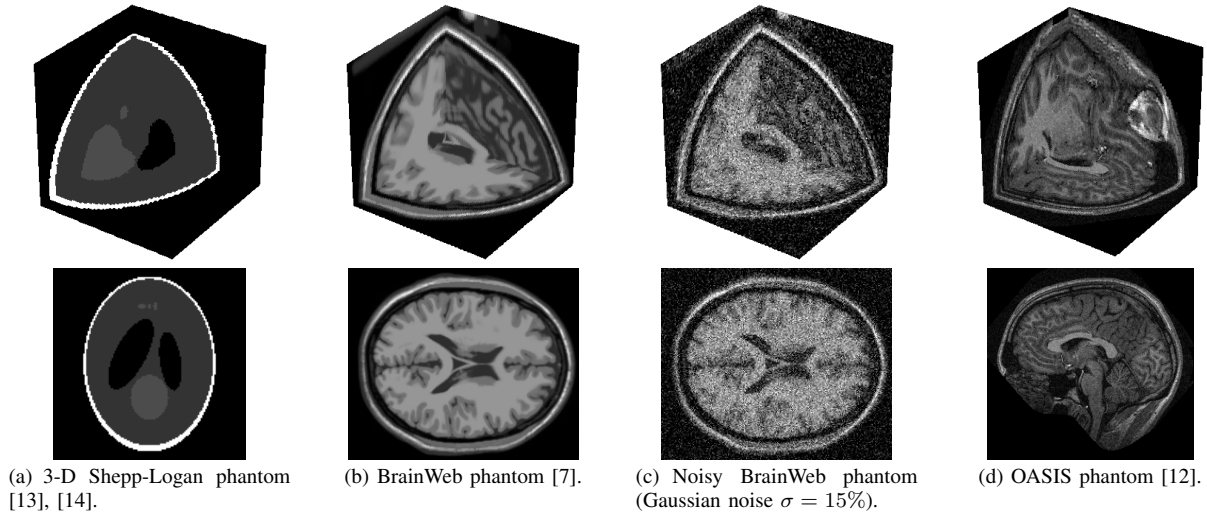
(d) OASIS phantom [12].

Fig. 7. Volumetric phantoms used in the denoising and reconstruction experiments. The 3-D and 2-D transversal cross-section of each phantom are presented in the top and bottom row of each subfigure, respectively.
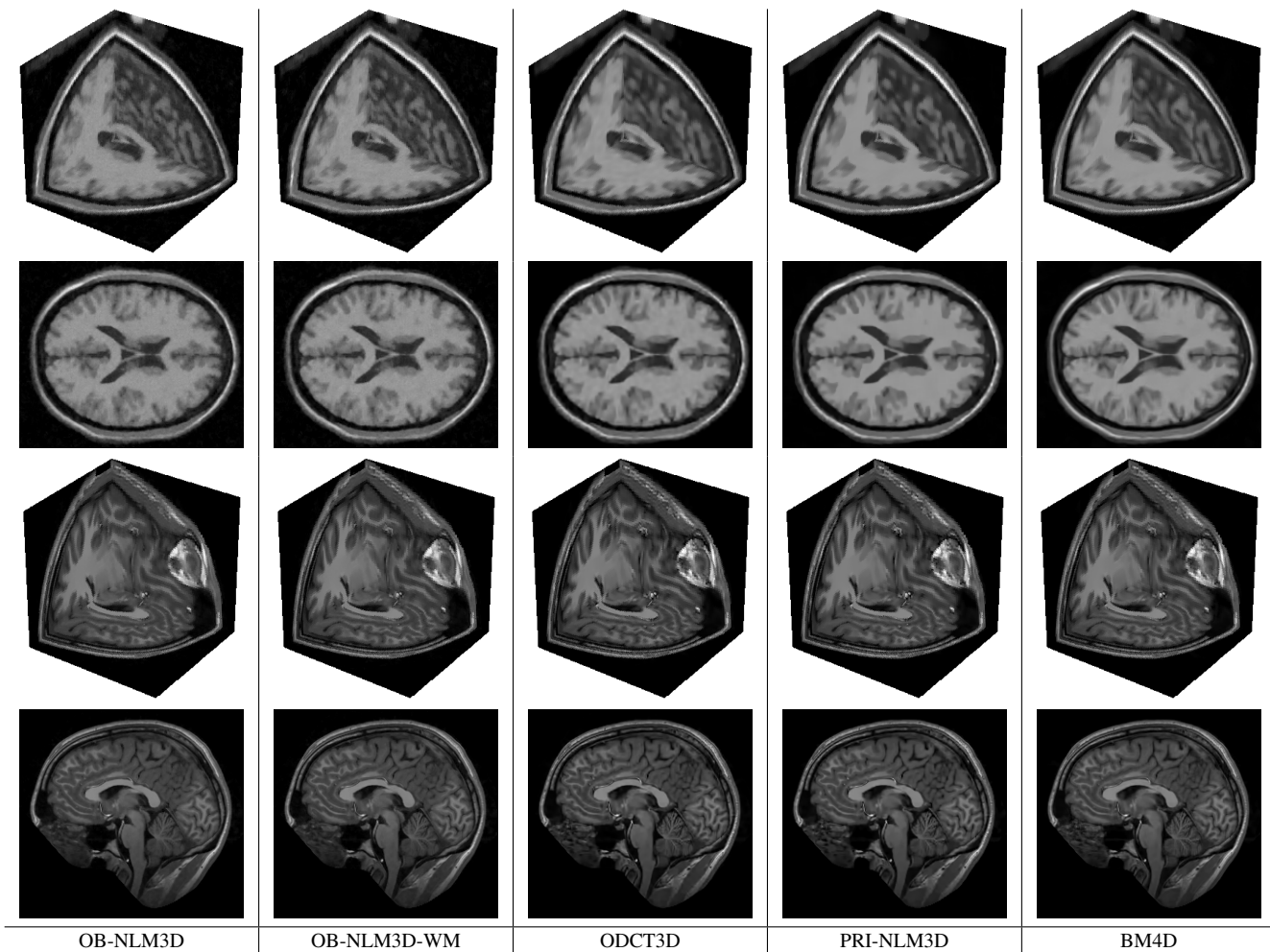

OB-NLM3D   OB-NLM3D-WM   ODCT3D   PRI-NLM3D   BM4D

Fig. 8. From left to right, denoising results of the OB-NLM3D, OB-NLM3D-WM, ODCT3D, PRI-NLM3D, and the proposed BM4D filter applied to the BrainWeb phantom corrupted by i.i.d. Gaussian noise with standard deviation $\sigma = 15\%$ (top) and the OASIS phantom (bottom) corrupted by Rician noise with standard deviation $\sigma \approx 4\%$ estimated as proposed in [16]. The corresponding noisy phantoms can be seen in Fig. 7(c), and Fig. 7(d), respectively. For each algorithm and phantom, both the 3-D and 2-D transversal cross-section are presented.
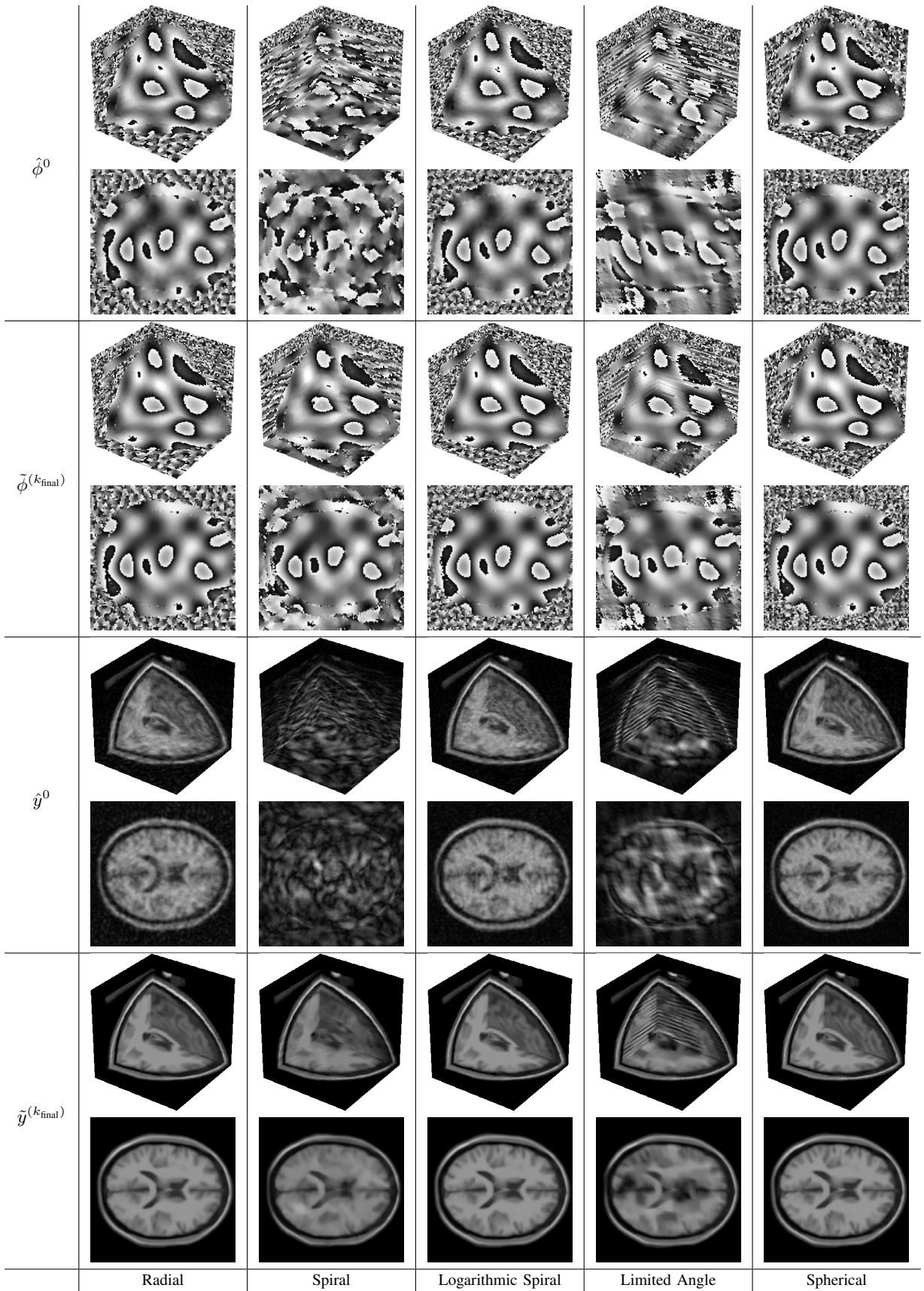
Fig. 9. Initial back-projections and final estimates of the magnitude and phase after $k_{\text{final}} = 1000$ iterations of the noisy reconstruction of the BrainWeb phantom ($\sigma = 5\%$) subsampled with ratio 30%. The original magnitude and phase volumes are shown in Fig. 7(b) and Fig. 3, respectively.
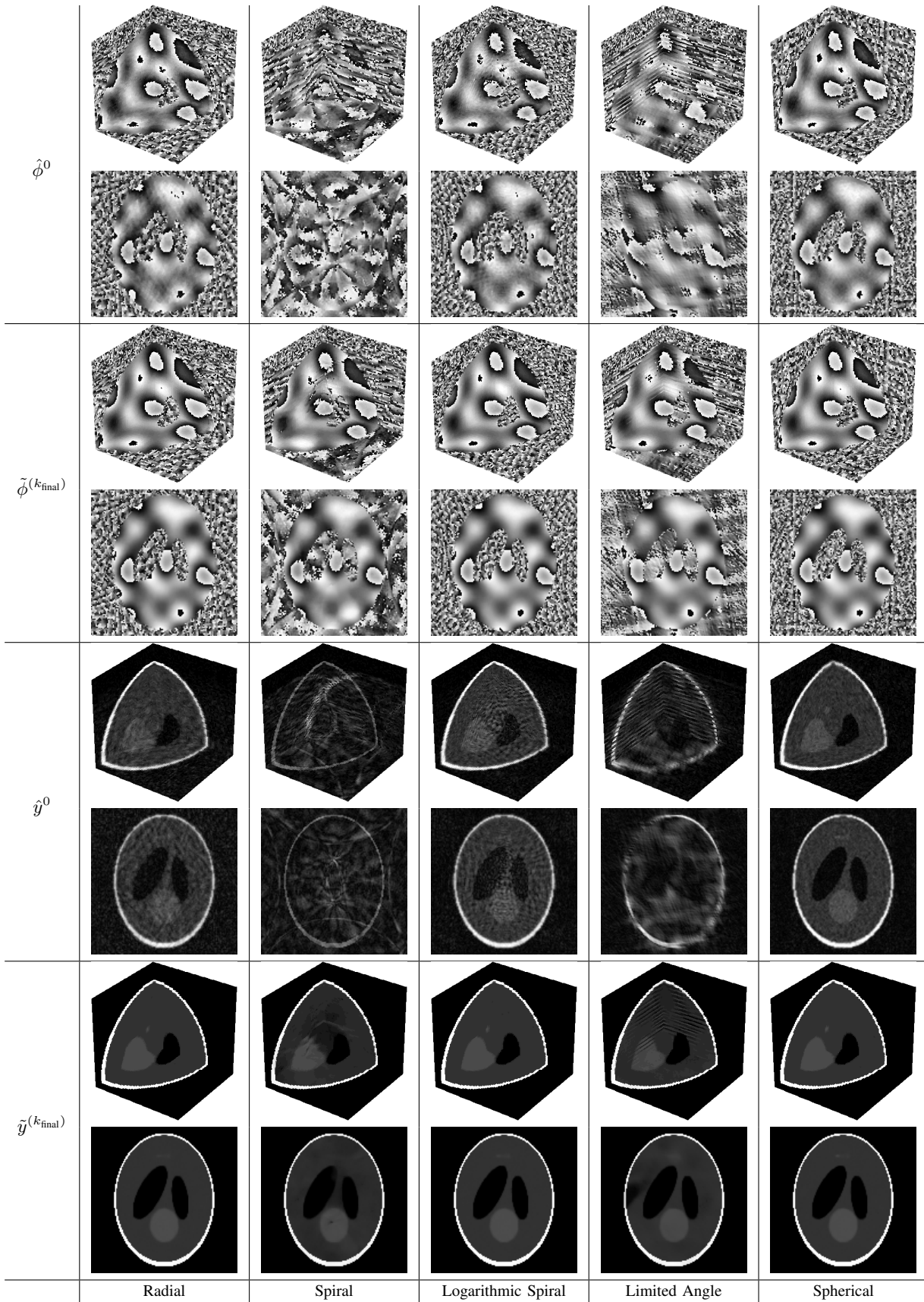
Fig. 10. Initial back-projections and final estimates of the magnitude and phase after $k_{\text{final}} = 1000$ iterations of the noisy reconstruction of the Shepp-Logan phantom ($\sigma = 5\%$) subsampled with ratio 30%. The original magnitude and phase volumes are shown in Fig. 7(b) and Fig. 3, respectively.