Sparse approximations in complex domain based on BM3D modeling

Vladimir Katkovnik, Mykola Ponomarenko and Karen Egiazarian

Laboratory of Signal Processing, Tampere University of Technology, Tampere, Finland
E-mail: vladimir.katkovnik@tut.fi, mykola.ponomarenko@tut.fi, karen.egiazarian@tut.fi

Abstract

In this paper the concept of sparsity for complex-valued variables is introduced in the following three types: directly in complex domain and for two real-valued pairs phase/amplitude and real/imaginary parts of complex variables. The nonlocal block-matching technique is used for sparsity implementation and filter design for each type of sparsity. These filters are complex domain generalizations of the Block Matching 3D collaborative (BM3D) filter based on the high-order singular value decomposition (HOSVD) in order to generate group-wise adaptive analysis/synthesis transforms. Complex domain denoising is developed and studied as a test-problem for comparison of the designed filters as well as the different types of sparsity modeling.

Keywords: Block matching, Complex domain, Denoising, Phase imaging, Sparsity

Sample document, Elsevier Article

2008 MSC: 68U10, 68W25, 68T05, 78A45

1. Introduction

This paper is focused on a special class of image/data processing dealing with complex-valued variables. For instance, in optics, a monochromatic wavefront is modeled in the form $u_o(x) = a_o(x)\exp(i\varphi_o(x))$ where an amplitude $a_o(x)$ and a phase $\varphi_o(x)$ are unknown variables of interest. Application areas for this type of models are many, ranging from astronomy and engineering to medicine and biology, usually dealing with *phase reconstruction* or *phase imaging*. Phase imaging is a key instrument in optics, allowing, in particular, to make visible tiny details of an internal structure of specimens usually completely invisible in intensity based incoherent light (e.g. [1, 2, 3]). Internal phase imaging is possible due to a strong phase sensitivity of the wavefront propagating through a specimen with respect to variations in its shape and internal structure. However, the wavefront phase cannot be measured directly due to the fact that measurement instruments register only a light intensity and are insensitive to a light phase. One of the main problems of data processing for coherent wavefronts is therefore the extraction of phase information from measured intensities. As an example, in interferometry and holography, measurements have the form:

$$I(x) = |u_o(x) + u_r(x)|^2 \tag{1}$$

where $u_o(x)$ and $u_r(x)$ stand for object and reference wavefronts, respectively, and $I(x)$ is a measured intensity (power).

In monochromatic coherent imaging both the object and reference wavefronts are complex-valued:

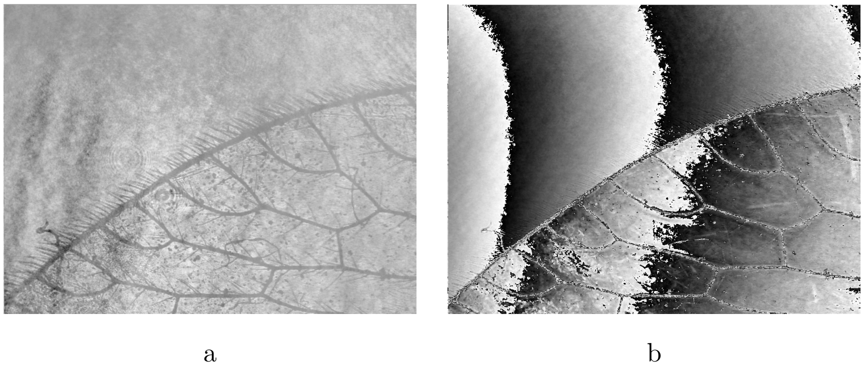
$$u_o(x) = a_o(x)\exp(i\varphi_o(x)),\ u_r(x) = a_r(x)\exp(i\varphi_r(x)). \tag{2}$$

In this scenario, the intensity takes the form

$$I(x) = a_o^2(x) + a_r^2(x) + 2a_o(x)a_r(x)\cos(\varphi_o(x) - \varphi_r(x)). \tag{3}$$

Then, phase imaging aims to reconstruct the phase $\varphi_o(x)$ from the, usually noisy, measurements of $I(x)$. The accurate phase reconstruction requires knowledge of the amplitude $a_o(x)$, consequently, a complex domain data processing deals with reconstruction of both phase and amplitude.

In Fig.1 we show phase and amplitude images of a fly's wing as a specimen obtained through the phase-shifting holography. While the laser beam propagates through the wing, its phase and amplitude are changing due to variations of a refractive index (for phase) and a transparency (for amplitude). One may note that, while the phase and amplitude images are far from being identical,



a

b

Figure 1: Fly's wing amplitude (a) and phase (b) images.

similar patterns can be noticed in these images. It is quite common that this sort of a strong cross-correlation (similar patterns) can be seen in both phase and amplitude of the object.

Under *modeling* of $u_o(x)$, we understand a mathematical tool for prediction, interpolation, denoising, etc., of $u_o(x)$ as a function of x.

We wish to note the following basic approaches applicable for the reconstruction of $u_o(x)$:

(1) *Point-wise* (*pixel-wise*) nonparametric estimation means that reconstruction of the object $u_o(x)$ is produced separately for each x . This kind of techniques, typical for many conventional algorithms, are originated in 60–70 of the last century and routinely used in our days (e.g. [2]).

(2) *Parametric series* (*approximations*) for $u_o(x)$. The object $u_o(x)$ is complex-valued, and it is natural that complex-valued bases are essential instruments in these approximations. Here we wish to mention the Fourier, windowed Fourier*,* Gabor transforms [4]*,* [5] and the complex domain wavelet transforms [4], [6], [7], [8].

(3) *Sparsity.* In computational imaging, sparse and redundant representations have been successfully developed in the last years as a general concept. Sparsity is based on the assumption that there exists a small number of basis functions (items) such that the image can be represented exactly or approximately with a good accuracy (e.g. [4], [9]). In term of statistics, a sparse

representation can be thought of as a low-order parametric approximation. In this classical form, the sparsity concept consists on zeroing small amplitude components of approximations. A principal point is that a sparse basis is unknown in advance and should be found, estimated or designed. One of the ways to approach this problem is linked with variational formulations for reconstruction of $u_o(x)$, where priors and regularizers serve as implicit tools for sparse basis design. The modern popularity and success of sparse representation are mainly due to the attractive theory and multiple successive applications where sparsity is enabled by different kind of penalizations for the object in questions. Here we refer to the recent overview [10], where, in particular, the recent sparsity based mathematical methods are discussed: semidefinite programming phase lifting using matrix completion *(PhaseLift algorithm*) and greedy sparse phase retrieval (*GESPAR algorithm*).

(4) *Group (or structured) sparsity.* In this technique the *sparsity is introduced artificially by a special grouping of observations.* In particular, for $2D$ images small similar $2D$ patches are grouped in $3D$ arrays of data which is then processed jointly. The similarity of the patches in these arrays enables a high-level sparsity of the grouped data and high efficiency of small-size approximations (e.g. [9], [11]). This kind of techniques became extremely popular after the work on Block Matching 3D (BM3D) filtering [12], which is recognized as the state of the art for denoising as well as for various other imaging problems (e.g. [13, 14, 15]). The recent development in this field can be seen, for instance, in [16, 17, 18].

Up to now in complex domain, the sparsity concept and algorithms well developed for real-valued variables are applied separately for phase and amplitude without consideration of their possible cross-correlation.

In the works on complex domain compressive sensing (CS), the sparsity prior is given by l_o-or l_1-norm which means that it is addressed only to amplitude of the wavefront (e.g. [19, 20, 21]) while the phase is ignored.

There are a few publications using penalization on both phase and amplitude (e.g. [22]). In particular, BM3D group sparsity, with independent sparsity modeling for phase and amplitude, was demonstrated for complex domain problems in optics [23, 24, 25, 26].

Recently, the sparsity concept was generalized for complex domain by simultaneously taking into consideration both phase and amplitude including their possible cross-correlation. In this development the algorithm works directly with complex-valued variables without separating phase and amplitude. The main problem is to design complex-valued bases enabling sparsity in complex domain. It was done in two different ways: the SpInPhase algorithm is based on the complex domain dictionary learning with internal and external dictionaries [27] and the BM3D group sparsity algorithm uses High-Order Singular Value Decomposition (HOSVD) for internal dictionary design based on self-similarity of the complex domain patches [28], [29].

The contribution of this paper concerns a further development of the approach initiated in [28], [29].

The complex-valued u_o can be considered as a function of two pairs of real-valued variables: amplitude and phase, a_o, φ_o, and real and imaginary parts of u_o, $\mathrm{Re}(u_o) = a_o \cos\varphi_o$, $\mathrm{Im}(u_o) = a_o \sin\varphi_o$.

Respectively, complex domain sparsity can be imposed in the following three different types:

(I) Complex domain sparsity treating u_o as a complex-valued variable;

(II) Sparsity imposed jointly on real and imaginary parts of u_o;

(III) Sparsity imposed jointly on amplitude and phase.

There is also a fourth type of the complex domain sparsity linked with the phase/amplitude pair: absolute phase $\varphi_{o,abs}$ and amplitude a_o. The phase φ_o as an argument of the complex u_o and of real and imaginary parts of u_o is the so-called principal (interferometric, wrapped) phase reduced to the 2π interval $[-\pi, \pi)$. The absolute phase can be inferred from the interferometric phase by adding of an integer number of 2π multiples. In some applications, where the range of phase variation exceeds the 2π interval the absolute phase sparse modeling may enable a more efficient phase imaging as compared with the sparse modeling for the corresponding wrapped phase. This happens because the fringes appearing in the wrapped phase make it more difficult to approximate, when compared with the corresponding absolute phase.

In this paper we do not deal with the absolute phase sparsity because it would require iterative algorithms for the phase reconstruction (e.g. [23], [24], [25]), which are out of scope of this paper. We study only less complex non-iterative algorithms.

The contribution of this paper is two fold. Firstly, a development of the novel complex domain denoising algorithms for each type of sparsity presented above. The algorithm based on the sparsity Type I is preceded by the algorithm from [28], [29]. There is, however, an essential difference between this already published algorithm and the novel algorithm developed in this paper. The algorithm in [28], [29], derived from the variational formulation of the complex domain denoising, is a single stage algorithm corresponding to the first thresholding part of the standard BM3D. The algorithm developed in this paper is a two stage algorithm including both the thresholding and Wiener parts. Moreover, a separate thresholding of real and imaginary parts of complex-valued analysis spectra is used in our novel algorithm that essentially improves the accuracy as compared with the predecessor algorithm from [28], [29], where the thresholding is applied to the absolute values of filtered spectra. The algorithms developed for the sparsity Type II and Type III are completely new and have no predecessors. Both these algorithms are also equipped with the Wiener filters as the second stage of processing.

Secondly, for comparison of the different types of sparsity and the corresponding algorithms we study their efficiency for complex domain denoising, i.e. retrieval of the complex-valued $u_o(x)$ from noisy data given with an additive complex-valued noise. A set of the extended simulation experiments is produced with thorough accuracy analysis in order to identify the most efficient types of the complex domain sparsity and the most efficient denoising algorithms.

There are at least two reasons to use the denoising as a test-problem for evaluation of the sparsity type:

1) The denoising of phase and amplitude is one of the basic procedures which appear as an unavoidable routine in many heuristic algorithms as well as in algorithms derived from the variational settings;

2) Extensive research in image processing shows that a good denoiser can be used as an efficient regularizator for algorithms produced by solving various optimization problems (e.g. [13], [30]).

Thus, the denoising algorithms developed and studied in this paper are of a general interest for different applications.

In what follows the paper is organized as follows. The observation model is discussed in Section 2. The proposed algorithms are presented in Section 3. The simulation results and experimental data processing are demonstrated and discussed in Section 4.

2. Observation modeling

Let us assume that the observed data $z : X \rightarrow \mathbb{C}$, where $X \subset Z^2$ is $2D$ grid of size $\sqrt{n} \times \sqrt{n}$, is modeled as

$$z(x) = u_o(x) + \varepsilon(x) \tag{4}$$

where $x \in X$, $u_o(x) \in \mathbb{C}^{\sqrt{n}\times\sqrt{n}}$ is a clean complex-valued object, and $\varepsilon(x) = \varepsilon_I(x) + j\varepsilon_Q(x) \in \mathbb{C}^{\sqrt{n}\times\sqrt{n}}$ is a complex-valued zero-mean Gaussian circular white noise of variance σ^2 (i.e., ε_I and ε_Q are zero-mean independent Gaussian random variables with variance $\sigma^2/2$).

Note that for real and imaginary parts of z(x) the noise is additive

$$\text{Re}(z(x)) = \text{Re}(u_o(x)) + \varepsilon_I(x),\ \text{Im}(z(x)) = \text{Im}(u_o(x)) + \varepsilon_Q(x) \tag{5}$$

while the noise dependence of the measured amplitude and phase is nonlinear and non-additive:

$$\begin{aligned} a_z(x) &= \sqrt{(\text{Re}(u_o(x) + \varepsilon_I(x))^2 + (\text{Im}(u_o(x)) + \varepsilon_Q(x))^2}, \\ \varphi_z(x) &= arctg(\frac{\text{Im}(u_o(x)) + \varepsilon_Q(x)}{\text{Re}(u_o(x) + \varepsilon_I(x)}). \end{aligned} \tag{6}$$

It can be proved by linearization of $z(x) = a_z(x)\exp(i\varphi_z(x))$ produced for small σ, i.e. for small random components in $a_z(x)$ and $\varphi_z(x)$, that the expectations and variances of amplitude and phase of the observed $z(x)$ are such that

$$E\{a_z(x)\} \simeq a_o(x),\, \sigma^2_{a_z}(x) = var(a_z(x)) \simeq \sigma^2, \tag{7}$$
$$E\{\varphi_z(x)\} \simeq \varphi_o(x),\, \sigma^2_{\varphi_z}(x) = var(\varphi_z(x)) \simeq \sigma^2/a_o^2(x).$$

Thus, for a small noise the observation z is unbiased with respect to u_o. The variance in the observed amplitude is equal to the additive noise variance and does not depend on the phase φ_o, while the variance in the observed phase depends on the amplitude a_o becoming larger for smaller amplitudes.

The problem considered in this paper is to reconstruct $u_o(x)$ from the noisy observation $z(x)$.

3. Algorithms

The proposed algorithms copy the structure of the standard BM3D filter as it is introduced in [12]. The difference concerns the following important aspects: the grouping is produced for complex-valued variables instead of real-valued ones in [12]; the transforms used for the data analysis are complex-valued and not fixed as is in [12] but data adaptive and generated by HOSVD; the input variables of HOSVD are different for the different algorithms: complex-valued, imaginary and real parts or phase and amplitude of complex-valued variables. Similarly to the standard BM3D, the proposed algorithms are composed from two successive stages: thresholding and Wiener filtering. The proposed algorithms could be understood as a generalization for the complex domain of the BM3D-SAPCA algorithm [31] where Singular Value Decomposition (SVD) is used for design of 2D orthonormal bases for patches, and also for HOSVD-BM3D proposed in [32] where HOSVD is exploited as a generator of 3D/4D real domain orthonormal transforms.

The flow chart shown in Fig.2 is the same for the all proposed algorithms.

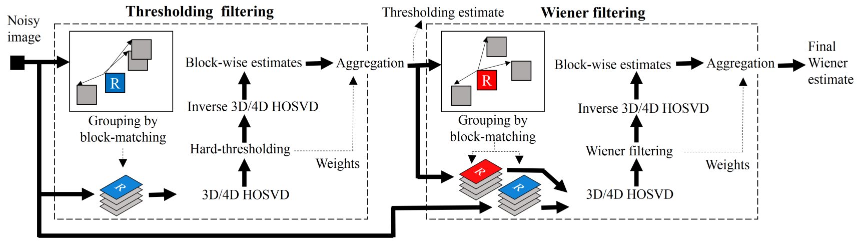


Figure 2: The flow chart of the proposed complex domain algorithms.

3.1. Sparsity Type I: CD-BM3D algorithm

As already mentioned, the algorithm is composed from two successive stages: thresholding and Wiener filtering.

3.1.1. Thresholding filtering

According to the conventional procedure in the patch-based image restoration, the noisy $\sqrt{n} \times \sqrt{n}$ image $\mathbf{z} \equiv \{z(x),\ x \in X\}$ is partitioned into small overlapping rectangular/square patches $N_1 \times N_2$ defined for each pixel of the image. The algorithm is built from the five basic steps: grouping, collaborative filtering via HOSVD transform, hard-thresholding of HOSVD spectrum, inverse HOSVD transform, and aggregation of the overlapping patch estimates. Here we only overview the structure of the thresholding stage of the algorithm referring for details to [29] (Section 2.1).

1. *Grouping*

Patches similar to the reference patch r are stacked together and form a 3D group (array, tensor) of size $N_1 \times N_2 \times J_r$, denoted by $\mathbf{Z}^r$, of the length J_r. These groups are built for all defined patches and may have different lengths, according to the used grouping rule.

2. *3D HOSVD*

The 3D group $\mathbf{Z}^r \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ is treated as a tensor of dimension $N_1 \times N_2 \times J_r$. The analysis of the group is produced using HOSVD. It gives the core tensor (*spectrum of the group*) $\mathbf{S}^r \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ and the three orthonormal complex domain transforms.

3. *Hard-thresholding*

The small items of the core tensor $\mathbf{S}^r$ are zeroed (thresholded) giving the filtered version $\hat{\mathbf{S}}^r \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ of the core tensor. Furthermore details of the used thresholding rule are given in Subsection 4.4.2.

4. *Inverse 3D HOSVD*

The filtered group data (core tensor) is transformed back into the image domain based on $\hat{\mathbf{S}}^r$ and the orthonormal transforms obtained in Step 2 of this algorithm.

5. *Aggregation*

Due to the overlapping patches and the grouping process, the set of patches contained in the 3D groups provides an overcomplete representation of the estimated image. The final estimate is obtained by calculation of the weighted mean of the overlapped estimates for each pixel.

The input-output representation of this stage can be given in the form

$$\hat{u}_o^{ht} = CD\text{-}BM3D^{th}(z, \delta) \tag{8}$$

where the superscript 'ht' corresponds to the thresholding and δ stands for the thresholding parameter.

The flow chart of this stage is shown in the left part of Fig.2.

3.1.2. Wiener filtering

The Wiener filter has two input variables: the noisy observation z and the reference variable u^{ref} carrying information on u_o. As can be seen in Fig.2, $u^{ref} = \hat{u}_o^{ht}$. The structures of the thresholding and Wiener filters are similar but with serious differences in variables and data processing.

1. *Grouping***.** The grouping rules used for the reference variable u^{ref} are identical to the grouping rules for the thresholding algorithm (see for details Section 2.1 in [29]). It gives the groups denoted as $G_r^{u^{ref}} \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$. In parallel, we form the groups from the second input z. These groups for z are formed exactly accordingly to $G_r^{u^{ref}}$, i.e. the group $G_r^z \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$includes the same patches as $G_r^{u^{ref}}$ but taken from z. Thus, the matched patches of u^{ref}

and z form two 3D tensors of size $N_1 \times N_2 \times J_r$ denoted by $\mathbf{Z}^r_{u^{ref}} \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ and $\mathbf{Z}^r_z \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$, respectively.

2. *3D HOSVD*. The arrays $\mathbf{Z}^r_{u^{ref}}$ and $\mathbf{Z}^r_z$ are processed differently. The HOSVD analysis is produced for the reference tensor allowing to represent it in the form [33, 34, 35]:

$$\mathbf{Z}^r_{u^{ref}} = \mathbf{S}^{u^{ref}}_r \times_1 \mathbf{T}^{ref}_{1,r} \times_2 \mathbf{T}^{ref}_{2,r} \times_3 \mathbf{T}^{ref}_{3,r} \tag{9}$$

giving the core tensor $\mathbf{S}^{u^{ref}}_r \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ (*reference spectrum*) and the orthonormal transforms $\mathbf{T}^{ref}_{1,r} \subset \mathbb{C}^{N_1 \times N_1}$, $\mathbf{T}^{ref}_{2,r} \subset \mathbb{C}^{N_2 \times N_2}$ and $\mathbf{T}^{ref}_{3,r} \subset \mathbb{C}^{N_{J_r} \times N_{J_r}}$. The symbols $\times_1$, $\times_2$, $\times_3$ stand for the products of the corresponding modes. The matrix transform $\mathbf{T}_{1,r}$ acts with respect to the variable l_1 in $\mathbf{Z}^r_{u^{ref}}(l_1, l_2, l_3)$ provided that l_2 and l_3 are fixed, the mode transforms $\mathbf{T}_{2,r}$ and $\mathbf{T}_{3,r}$ with respect to the variables l_2 and l_3 have similar meaning.

These transforms are applied for the analysis of the observation tensor $\mathbf{Z}^r_z$. The result of this analysis is the corresponding observation core spectrum $\mathbf{S}^z_r$ calculated as

$$\mathbf{S}^z_r = \mathbf{Z}^r_z \times_1 (\mathbf{T}^{ref}_{1,r})^H \times_2 (\mathbf{T}^{ref}_{2,r})^H \times_3 (\mathbf{T}^{ref}_{3,r})^H \tag{10}$$

where H stays for the conjugate (Hermitian) transpose.

3. Wiener filtering (empirical) defines the filtered version of the core tensor $\mathbf{S}^z_r \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ in the form

$$\hat{\mathbf{S}}^z_r = \frac{|\mathbf{S}^{u^{ref}}_r|^2 \cdot \mathbf{S}^z_r}{|\mathbf{S}^{u^{ref}}_r|^2 + \beta} \tag{11}$$

where all operations with the tensors are element-wise and $\beta > 0$ is a regularization parameter.

4. *Inverse 3D HOSVD***.** The filtered core tensor $\hat{\mathbf{S}}^z_r$ is transformed into the image domain using the reference transforms found in (*9*):

$$\hat{\mathbf{U}}^r_z = \hat{\mathbf{S}}^z_r \times_1 \mathbf{T}^{ref}_{1,r} \times_2 \mathbf{T}^{ref}_{2,r} \times_3 \mathbf{T}^{ref}_{3,r}. \tag{12}$$

5. *Aggregation* of the overlapped estimates is produced as in the thresholding algorithm (see Section 2.1 in [25]) with the weights g_r:

$$g_r = 1/||\hat{\mathbf{S}}^z_r||^2_2.$$

Here the norm $||\cdot||_2^2$ is calculated as the sum of the squared absolute values of all items of the tensor-argument.

The corresponding input-output relation in this stage is denoted as

$$\hat{u}_o^{wi} = CD\text{-}BM3D^{wi}(z, u^{ref}\beta) \tag{13}$$

where the superscript 'wi' (Wiener) marks the stage as well as the output, and β stands for the regularization parameter.

The flow chart of this stage can be seen in the right part of Fig.2.

3.2. Sparsity Type II and Type III: ImRe-BM3D and PhAm-BM3D algorithms

As discussed in the introduction, the complex domain sparsity can be thought of as sparsity of the pairs: imaginary/real parts and phase/amplitude of complex-valued variables (sparsity Type II and Type III, respectively).

Each of the algorithms of these groups is composed from the thresholding and Wiener filtering stages, which usually have structures identical to what was shown for the algorithm in Subsection 3.1.

3.2.1. The thresholding filtering

The essential difference with the algorithm presented above in Subsection 3.1 concerns the steps 2 and 4, where HOSVD is used for the analysis and synthesis transformations:

- For the sparsity Type II, after grouping we calculate the imaginary and real parts for all items of the 3D tensor $\mathbf{Z}^r \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ and obtain two tensors, each of dimension $N_1 \times N_2 \times J_r$. We join these two tensors in a single tensor of dimension $N_1 \times N_2 \times J_r \times 2$, where $N_1 \times N_2$ is dimension of the patches, J_r is the length of the group (tensor) and 2 is a number of $3D$ tensors combined in a single one of dimension $4D$.
- For the sparsity Type III, we calculate the amplitude and phase for all items of the 3D tensor $\mathbf{Z}^r \subset \mathbb{C}^{N_1 \times N_2 \times J_r}$ and obtain two $3D$ tensors which again are combined in a single $4D$ tensor of dimension $N_1 \times N_2 \times J_r \times 2$.

The 4D tensors obtain in this way require 4D HOSVD for analysis and synthesis , which makes a big deal of difference when compared with the sparsity Type I restricted to 3D HOSVD.

All other steps of the both algorithms are exactly as they are in Section 3.1.1. These steps define the ImRe-BM3Dth and PhAm-BM3Dth algorithms. The superscript 'ht' emphasizes that these algorithms use the thresholding for filtering of the corresponding analysis core tensors.

3.2.2. The Wiener filtering

The similar procedures based on the steps given in Sections 3.1.2 with replacements of the $3D$ tensor by the corresponding $4D$ tensors give the ImRe-BM3D^{wi} and PhAm-BM3D^{wi} algorithms, where the superscript 'wi' notes that the Wiener filtering is applied, respectively, to the pairs phase/amplitude and real/imaginary parts of complex-valued variables.

3.2.3. The complete algorithms

The notation of the algorithms without the superscript 'ht' or 'wi' as CD-BM3D, ImRe-BM3D and PhAm-BM3D denote that these are the two stage algorithms as shown in the flowchart Fig.2, where the algorithm's output is the output of the Wiener filter. The fact that 3D or 4D HOSVD are used in the different algorithms is depicted in the generic flowchart Fig.2.

In what follows, our experimental results are shown for these complete, two stage, algorithms. Overall, we obtained the three different algorithms to be compared: CD-BM3D, ImRe-BM3D and PhAm-BM3D.

It is important to note that using the higher order 4D HOSVD allows processing of the phase/amplitude and real/imaginary pairs while automatically taking into consideration the cross-correlation between the elements of these pairs.

We make publicly available the MATLAB demo-codes[1] of the developed

[1] http://www.cs.tut.fi/sgn/imaging/sparse

algorithms, which can be used to reproduce the experiments presented in this paper as well as for further tests.

4. Simulation experiments

In this section we present simulation results illustrating and comparing the accuracy of the developed algorithms. We are focused on phase imaging, and the following standard performance criteria are used. For the interferometric phase it is the peak signal-to-noise ratio ($PSNR$):

$$PSNR_{\varphi} = 10 \log_{10} \frac{n(2\pi)^2}{||\mathcal{W}(\hat{\varphi}_o - \varphi_o)||_2^2} [dB] \qquad (14)$$

where $\hat{\varphi}_o$ and φ_o are the phase reconstruction and the true phase, respectively; n is the image size in pixels; the phase wrapping operator $\mathcal{W}$ is used in order to eliminate the phase shifts in errors multiple to 2π. The factor $(2\pi)^2$ in the numerator of (14) stays for the squared maximum value of the interferometric phase.

We unwrap the estimated interferometric phase with the PUMA algorithm [36] in order to get estimates of the true absolute phase $\varphi_{o,abs}$. The accuracy of the absolute phase reconstruction is measured by the root-mean-squared error (RMSE):

$$RMSE_{\varphi_{abs}} = \sqrt{\frac{1}{n}||(\hat{\varphi}_{o,abs} - \varphi_{o,abs} - \Delta_{\varphi})||_2^2} \qquad (15)$$

where a scalar Δ_{φ} compensates an invariant shift in the absolute phase estimate due to the unwrapping procedure. This Δ_{φ} is calculated as the mean value of the difference $\hat{\varphi}_{o,abs} - \varphi_{o,abs}$.

The reconstruction accuracy for the amplitude a_o is characterized by $PSNR$ as:

$$PSNR_{ampl} = 10 \log_{10} \frac{n \max^2(a_o)}{||a_o - \hat{a}_o||_2^2} [dB]. \qquad (16)$$

The noisy observation data is simulated according to the model (4), where the standard deviation σ of the noise is calculated as

$$\sigma{=}\sigma_{\varphi_z} \cdot mean_x(a_o(x)) \qquad (17)$$

where σ_{φ_z} is the standard deviation of the noisy phase in noisy observations z (see Eq.(7)).

If $a_o(x)$ is invariant and $a_o(x) \equiv 1$, it follows from (7) that the standard deviation of the noisy phase $\sigma_{\varphi_z} = \sigma$. In general, if the amplitude a_o is varying, the formula (17) allows to keep the variance of the noisy phase proportional to σ. Being concentrated in this work on the phase reconstruction, we use (17) in order to control the noise level in the observed phase for any amplitude variable $a_o(x)$. The experiments are produced for the set $\sigma_{\varphi_z} = \{0.05, 0.1, 0.2, 0.5, 0.9\}$. The largest value of σ_{φ_z} corresponds to the very noisy observations.

Note that in our experiments the Wiener filter is used in the following way:

$$\hat{u}_o^{wi} = CD\text{-}BM3D^{wi}(0.1z + 0.9\hat{u}_o^{ht},\, \hat{u}_o^{ht},\, \beta). \tag{18}$$

Here instead of the noisy input signal z we use a mix of this z with the hard-thresholding estimate $\hat{u}_o^{ht}$ as $0.1z + 0.9\hat{u}_o^{ht}$ and $\beta = \sigma^2/4$. It was found by the experimental study that this mixed input signal gives better result, when compared with the case where z is used as the observation argument of the Wiener filter.

Overall, in comparison with the hard-thresholding, the Wiener filtering gives an improvement in $PSNR_\varphi$ of about $0.2 - 0.3$ dB. As is already mentioned above all experimental results are shown for the two stage algorithms where the thresholding is complemented by the Wiener filtering.

For comparison we show also results obtained by the standard BM3D applied independently to real and imaginary parts of z accordingly to the flowchart in Fig.3. Further, we use the abbreviation BM3D for this application of the standard BM3D filter to complex-valued data.

It is of special interest to compare this straightforward application of BM3D with ImRe-BM3D, where the data adaptive basis is generated by $4D\ HOSVD$ and correlation between real and imaginary parts of observations are taken into consideration. As it will be shown later, the proposed algorithms and in particular ImRe-BM3D outperform BM3D.

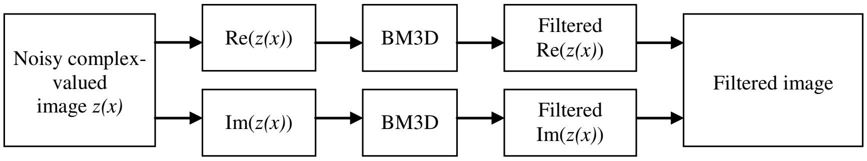


Figure 3: Application of the standard BM3D filter [12] for complex-valued data denoising.

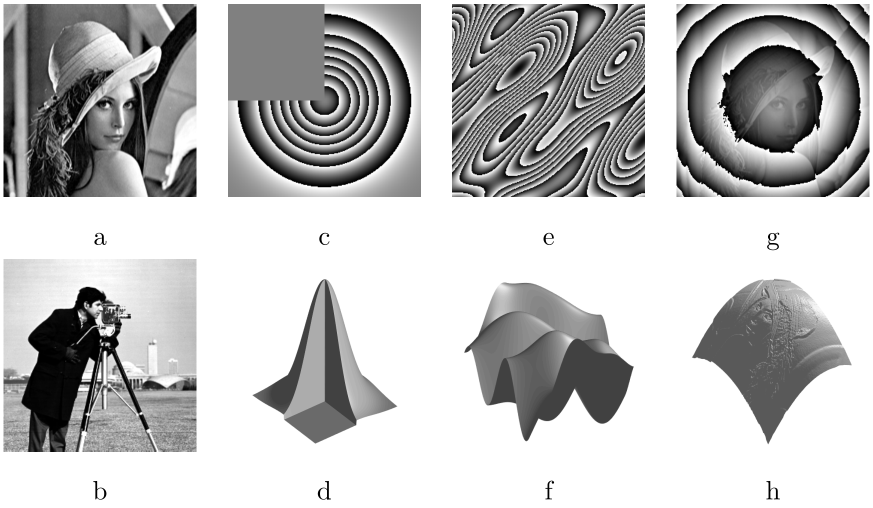


Figure 4: Test phase images.

4.1. Test phase images

The test phase images $\varphi_o(x)$, of size 256 × 256, are shown in Fig.4.

The images lena (a) and cameraman (b) are interferometric phases normalized to the interval [0, $\pi/2$]. The absolute phase corresponding to the images (c) and (d) is a truncated Gaussian distribution with maximum value equal to 44 radians. One of the the quadrants of this distribution is zeroed, which makes this absolute phase discontinuous and especially difficult for reconstruction. In image (c) we show the wrapped version of this phase. The corresponding absolute phase is presented as a 3D surface in image (d).

The "*mountains*" images (e)-(f) are shown as wrapped and 3D absolute phase, which is smooth with multiple picks. The range of this phase is from -5 to 3.3 radians.

The absolute phase for images (g)-(h) is the sum of the lena phase from the image (a) and the quadratic phase with the maximum value 26 radians. In image (g) the lena features are destructed by the fringes of the quadratic phase. In the absolute phase image (h) lena is nearly invisible due to the large magnitude of the quadratic phase.

Note that our algorithms work with wrapped phase variables as shown in images c, e, g, and the absolute phase reconstructions are obtained by unwrapping the estimates of the corresponding wrapped phases. Closed fringes, typical for the wrapped phase, are well seen in the images (c, e, g) and make the problem of the phase restoration very hard. In order to differentiate between the test images (a,b) and (c)-(h) we will refer to the former as interferometric phases and to the later as wrapped phases.

4.2. Test amplitude images

As is illustrated in Fig.1 phase and amplitude images can be quite correlated. In order to imitate this sort of dependence between phase and amplitude, we introduce the following function for calculation of amplitude as a function of phase:

$$a_o(x) = k_0 + k_1\alpha(x) \tag{19}$$

where a_o is the test-amplitude, k_0 and k_1 are parameters. Different values of k_0 and k_1 allow to obtain test images u_o with various signal-to-noise ratio (SNR) in phase and amplitude as well as in real and imaginary components of u_o.

The following $\alpha(x)$ in (19) define three types of amplitude test images:

Type 1:

$$\alpha(x) = 1;$$

Type 2:

$$\alpha(x) = |\varphi_{o,abs}(x)| / \max_x (|\varphi_{o,abs}(x)|);\tag{20}$$

Type 3:

$$\alpha(x) = |cos(15\varphi_{o,abs}(x))|.\tag{21}$$

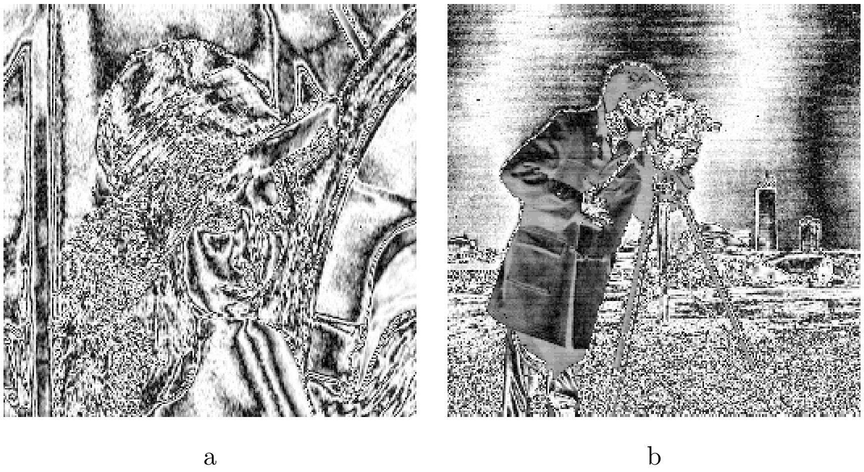


Figure 5: Test amplitudes of Type 3 for the test phases a) and b).

In Type 1, the amplitude is phase invariant; in Types 2 and 3 the amplitude is varying and phase dependent.

Fig.5 shows examples of the varying amplitudes obtained for lena and cameraman phases according to the formula (21) for $k_o = k_1 = 1$. One may note features of lena and cameraman in these amplitude images, while the amplitudes are obviously different from the corresponding phases.

4.3. Complex-valued test images

Combining the introduced phase and amplitude test images with the later being calculated for different k_0 and k_1, we generate a large set of different complex-valued $u_o(x)$. As a result of our preliminary study, we selected 20 representative models shown in Table 1. The first 12 models are used with wrapped phases and the others (13–20) with interferometric phases.

These models are assembled in 9 groups having more or less similar features and demonstrating good performance for the same algorithms. For example, ImRe-BM3D is advantageous for the complex-valued models in Groups 2 and 3.

Table 1: Complex-valued test images

Image #	Phase type as in Fig. 3	Amplitude type	k_0	k_1	Comments	Group	
1	*c, d*	*1*	*1*	*1*	Invariant amplitude	1	wrapped phases
2	*e, f*	*1*	*1*	*1*			
3	*g, h*	*1*	*1*	*1*			
4	*c, d*	*2*	*1*	*1*	Small amplitude very similar to phase	2	
5	*e, f*	*2*	*1*	*1*			
6	*g, h*	*2*	*1*	*1*			
7	*c, d*	*2*	*1*	*3*	Large amplitude very similar to phase	3	
8	*c, d*	*3*	*1*	*1*	Small amplitude of low similarity to phase	4	
9	*e, f*	*3*	*1*	*1*			
10	*g, h*	*3*	*1*	*1*			
11	*c, d*	*3*	*1*	*5*	Large amplitude of low similarity to phase	5	
12	*e, f*	*3*	*1*	*5*			
13	*a*	*1*	*1*	*1*	Invariant amplitude	6	interferometric phases
14	*b*	*1*	*1*	*1*			
15	*a*	*2*	*0.1*	*1*	Close to zero amplitude very similar to phase	7	
16	*b*	*2*	*0.1*	*1*			
17	*a*	*2*	*1*	*3*	Large amplitude very similar to phase	8	
18	*b*	*2*	*1*	*3*			
19	*a*	*3*	*1*	*3*	Large amplitude of low similarity to phase	9	
20	*b*	*3*	*1*	*3*			

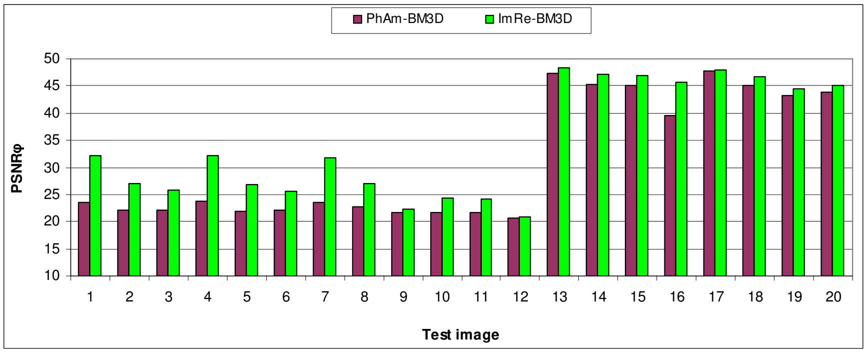


Figure 6: Comparison of the algorithms PhAm-BM3D and ImRe-BM3D for $\sigma_{\varphi_z} = 0.05$.

4.4. Comparative analysis of the algorithms

4.4.1. ImRe-BM3D versus PhAm-BM3D

Fig.6 shows the diagram of $PSNR_\varphi$ values in dB for the algorithms PhAm-BM3D and ImRe-BM3D, σ_φ=0.05. These results are obtained for the complex-valued test objects from Table 1.

Keep in mind that the first 12 test objects have as a model the wrapped phases and the others the interferometric phases. It is seen that for all tests

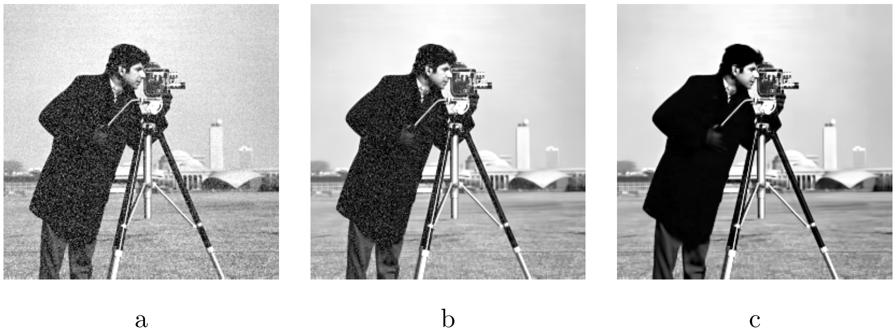


Figure 7: Complex-valued test image 16: a) noisy phase, b) phase reconstruction by PhAm-BM3D, $PSNR_{\varphi}$=39.6 dB, c) phase reconstruction by ImRe-BM3D, $PSNR_{\varphi}$=45.7 dB.

ImRe-BM3D outperforms PhAm-BM3D by showing higher values of $PSNR_{\varphi}$. While the results are more or less comparable for the test images 13-20, for the tests 1-12 the difference is in some cases very large up to 9 dB for the test images 1,4,7.

For the test image 16, the gain of ImRe-BM3D versus PhAm-BM3D is about 5 dB. In this case the level of the noise in φ_z is quite high and more importantly, it has a heavy-tailed distribution very different from the Gaussian one. In this situation HOSVD in PhAm-BM3D is inefficient. The noisy phase and the phase reconstructions by ImRe-BM3D versus PhAm-BM3D for this case are shown in Fig.7. High level errors in the noisy phase and the phase reconstruction by PhAm-BM3D are well seen in dark image regions which have low amplitude values.

The comparative tests of ImRe-BM3D versus PhAm-BM3D produced for other values of σ_{φ_z} show similar results with the clear advantage for ImRe-BM3D.

As a result of this analysis we drop PhAm-BM3D from further consideration and concentrate on comparing ImRe-BM3D with CD-BM3D.

4.4.2. CD-BM3D: imag/real thresholding versus amplitude thresholding

The thresholding (hard-thresholding) in CD-BM3D is applied to the complex-valued items of the core tensors. It can be implemented as the thresholding of the amplitude of the input variable x with the rule

$$\hat{x} = x \cdot 1(|x| \leq \delta_r) \tag{22}$$

where $1(x)$ is an indicator function, $1(x) = 1$ for $x > 0$ and otherwise $1(x) = 0$, and δ is the threshold parameter.

As an alternative the hard-thresholding could be applied independently to real and imaginary parts of x with the similar rule:

$$\begin{aligned} \operatorname{Re}\hat{x} &= \operatorname{Re} x \cdot 1(|\operatorname{Re} x| \leq \delta_r), \\ \operatorname{Im}\hat{x} &= \operatorname{Im} x \cdot 1(|\operatorname{Im} x| \leq \delta_r). \end{aligned} \tag{23}$$

Following to [37] we select for these rules the universal threshold

$$\delta_r = \eta\sigma\sqrt{2\log N_1 N_2 J_r} \tag{24}$$

where the parameter η is selected from experiments.

We compare the performance of CD-BM3D with the two different thresholding rules (22) and (23).

In this subsection we denote CD-BM3D with the amplitude thresholding (22) as CD-BM3D-ABS, and with imag/real thresholding (23) as CD-BM3D-REIM.

The curves in Fig.8 compare the accuracy of the phase reconstruction by CD-BM3D-ABS and CD-BM3D-REIM. These results are obtained as an average for the all 20 test complex-valued objects. For the high level of the noise standard deviation, the results are nearly identical with a slight advantage of CD-BM3D-REIM. The advantage of CD-BM3D-REIM becomes more pronounced about $0.2 - 0.3$ dB for the lower noise level.

As a result of this analysis we keep the imag/real thresholding and drop the amplitude thresholding from further consideration. In what follows the CD-BM3D is always used with the imag/real thresholding rule (23).

This allows us to reduce the further analysis to comparison of only two algorithms CD-BM3D and ImRe-BM3D both with the imag/real thresholding defined by the rule (23).

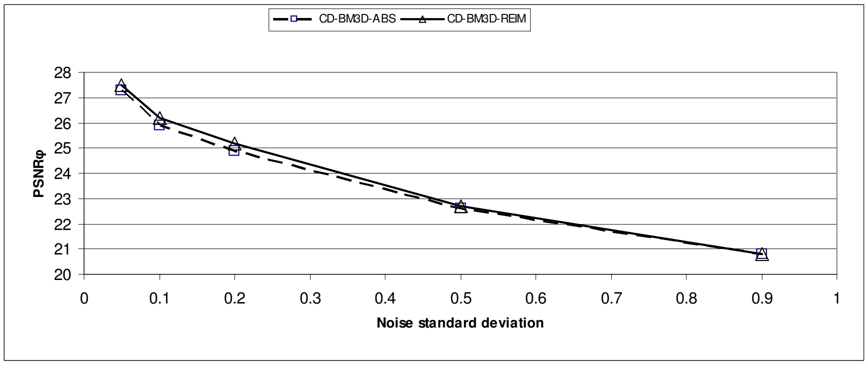


Figure 8: CD-BM3D, comparison of two thresholding rules: amplitude versus imag/real thresholdings.

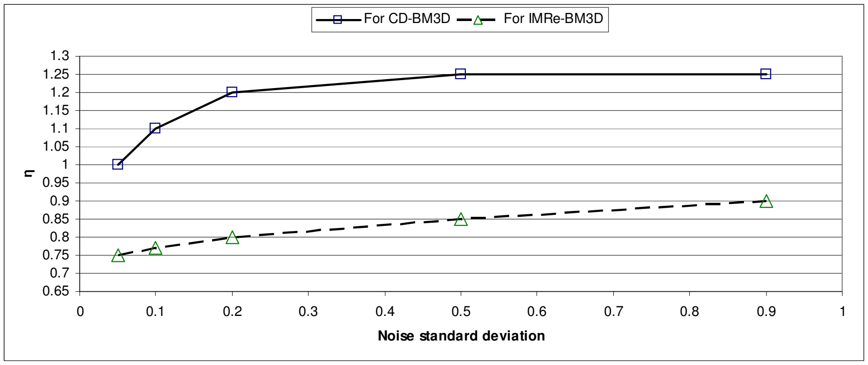


Figure 9: Optimization of threshold parameter η in (24) for CD-BM3D and ImRe-BM3D.

Note, that the amplitude thresholding rule (22) is used by default in [28] and [29], which defines a further difference between CD-BM3D in those and this papers.

4.4.3. Threshold parameter optimization

Let us consider optimization of the threshold parameter η in (24) for CD-BM3D and ImRe-BM3D. In Fig.9 we show the optimal values of this parameter found for different values of the standard deviation σ_{φ_z}. Naturally, a higher level of the noise requires a larger threshold parameter. This optimization is produced including the all 20 complex-valued test images.

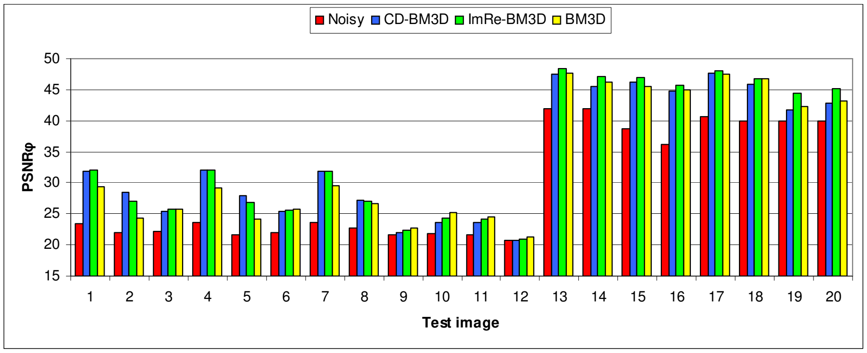


Figure 10: $PSNR_\varphi$ values for noisy data, CD-BM3D, ImRe-BM3D and BM3D, $\sigma_{\varphi_z} = 0.05$.

4.5. CD-BM3D versus ImRe-BM3D

In these experiments we use the optimal threshold parameters shown in Fig.9. The simulation is produced for the all 20 test objects and $\sigma_{\varphi_z}= \{0.05, 0.1, 0.2, 0.5, 0.9\}$. However, only the results for $\sigma_{\varphi_z}= \{0.05, 0.2, 0.9\}$ are shown in the following analysis,i.e. for low, middle and high level noise. The values of $PSNR_\varphi$ are given for: noisy phase (raw data) and reconstructions by CD-BM3D, ImRe-BM3D and BM3D.

Let us start from the low noise case, $\sigma_{\varphi_z} = 0.05$, Fig.10.

In addition to the data in Fig.10 we calculate mean values of $PSNR_\varphi$ for the entire test set. For CD-BM3D, ImRe-BM3D and BM3D we obtain 27.5 dB, 27.8 dB and 27.2 dB, respectively. Thus, in average for small noise level, the best performance is demonstrated by ImRe-BM3D outperforming BM3D by 0.6 dB and CD-BM3D by 0.3 dB.

In comparison with the noisy raw phase, the all three algorithms are quite efficient with an improvement in $PSNR_\varphi$ of up to 10 dB. However, some test-images are proved to be challenging. For instance, for the test image 12, the reconstruction results and the noisy phase have nearly identical $PSNR_\varphi$ values. In this case the amplitude estimation is very difficult for block matching as it affects the noise in both real and imaginary components of the image.

The test-image 2 is also of interesting analysis. The amplitude is invariant

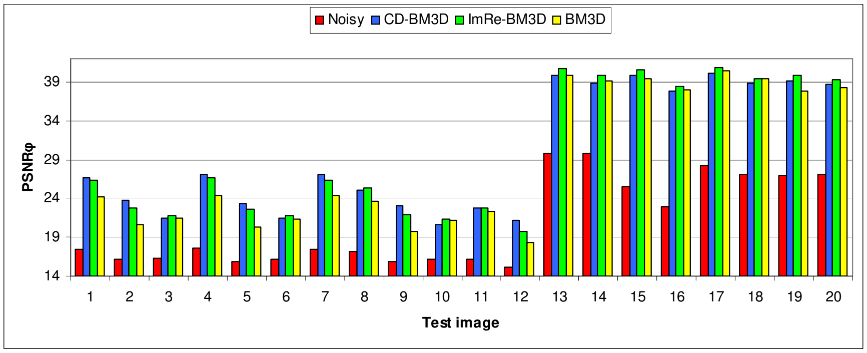


Figure 11: $PSNR_\varphi$ values for noisy data, CD-BM3D, ImRe-BM3D and BM3D, $\sigma_{\varphi_z} = 0.2$.

and does not affect the shapes of phase, and real and imaginary components. As a result edges of these object components are well matched and effectiveness of block matching significantly benefits both CD-BM3D and ImRe-BM3D which outperforms BM3D by 4 dB. This advantage is typical for objects with an invariant amplitude.

Comparing CD-BM3D and ImRe-BM3D in Fig.10 we may note that ImRe-BM3D nearly always outperforms CD-BM3D.

Fig.11 allows the comparison of the algorithms for the middle level noise in observations, $\sigma_{\varphi_z} = 0.2$. The conclusion is now slightly different. For the wrapped phase test images $1 - 12$ the best results are shown mainly by CD-BM3D, while for other tests with the interferometric phase ImRe-BM3D continues to show better accuracy than CD-BM3D.

The average gap between ImRe-BM3D and BM3D increases from 0.6 dB to 1.9 dB as compared with the case $\sigma_{\varphi_z} = 0.05$.

In order to illustrate the benefit of this 1.9 dB, we show the phase and amplitude of the noisy test-image 19 and the phase/amplitude reconstructions obtained by ImRe-BM3D, CD-BM3D and BM3D, Fig.12.

As is seen in Fig.12, the residual phase noise in the reconstructions by CD-BM3D and BM3D is visually noticeable as artifacts in areas of high contrast edges on the amplitude. In the reconstructions by ImRe-BM3D that provides

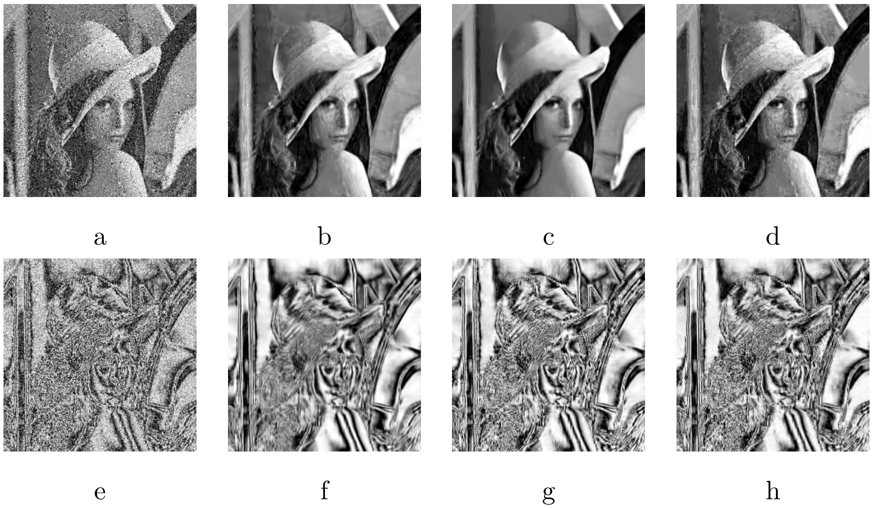


Figure 12: Test image 19: a) noisy phase for σ_{φ_z}=0.2, $PSNR_{\varphi}$=26.9 dB, b) phase reconstruction by CD-BM3D, $PSNR_{\varphi}$=39.1 dB, c) phase reconstruction by ImRe-BM3D, $PSNR_{\varphi}$=39.8 dB, d) phase reconstruction by BM3D, $PSNR_{\varphi}$=37.8 dB, e) noisy amplitude, $PSNRampl$=16.7 dB, f) amplitude reconstruction by CD-BM3D, $PSNR_{ampl}$=17.1 dB, g) amplitude reconstruction by ImRe-BM3D, $PSNR_{ampl}$=19.1 dB, g) amplitude reconstruction by BM3D, $PSNR_{ampl}$=18.6 dB.

a better $PSNR_{\varphi}$ with no such artifacts, but the algorithm oversmooths low contrast textures and small details. Overall, ImRe-BM3D provides a better visual quality suppressing noise and enabling good edge preservation.

Concerning the amplitude reconstruction none of the compared algorithm is able to improve significantly the reconstruction accuracy for such complicated noise-like amplitude.

Finally, let us analyze the abnormally high level noisy data with $\sigma_{\varphi} = 0.9$, Fig.13. In average for the entire test set, $PSNR_{\varphi}$ for noisy phase images is increased by 9.1 dB as compared with $\sigma_{\varphi} = 0.05$.

Overall, CD-BM3D outperforms ImRe-BM3D by 0.4 dB and outperforms BM3D by 2.1 dB.

Comparing the results for different test images shown in Fig.13, we may note that for the wrapped phase test images $1-12$ the best results are achieved by CD-BM3D, while for other tests with the interferometric phase ImRe-BM3D

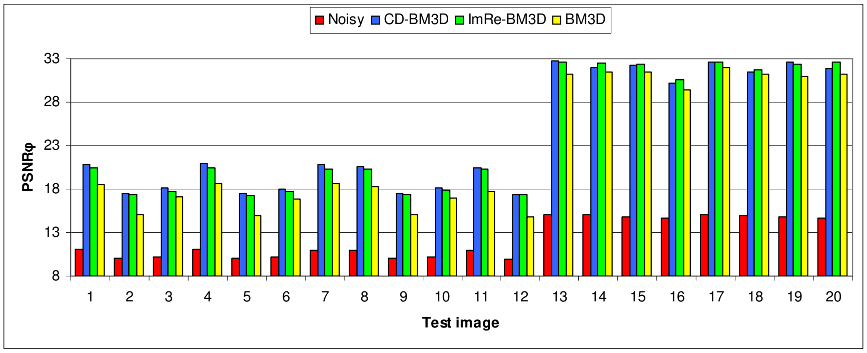


Figure 13: $PSNR_{\varphi}$ values for noisy data, CD-BM3D, ImRe-BM3D and BM3D, $\sigma_{\varphi_z} = 0.9$.

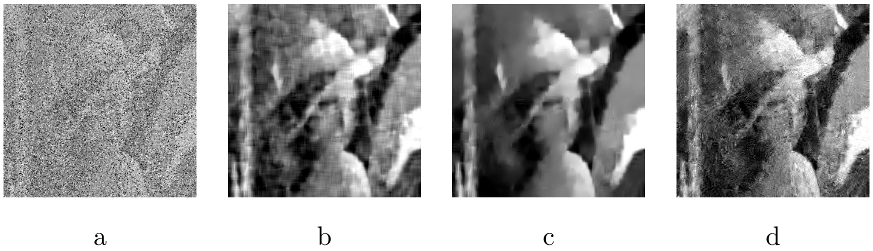


Figure 14: Test image 19: a) noisy phase for σ_{φ_z}=0.9, $PSNR_{\varphi}$=14.8 dB, b) phase filtered by CD-BM3D, $PSNR_{\varphi}$=32.6 dB, c) phase filtered by ImRe-BM3D, $PSNR_{\varphi}$=32.4 dB, c) phase filtered by BM3D, $PSNR_{\varphi}$=30.9 dB.

continues to demonstrate a better accuracy than CD-BM3D.

In order to give an idea how strong is the high level noise with $\sigma_{\varphi_z} = 0.9$, we show the noisy phase for the test-image 19 as well as the reconstructions by the compared algorithms, Fig.14.

As is seen in Fig.14, the small details of the images are not reconstructed but completely destroyed. Nevertheless, ImRe-BM3D (Fig.14c) visually provides a better noise suppression, while CD-BM3D (Fig.14b) is better at preserving details and has a slightly higher $PSNR_{\varphi}$.

Fig.15 shows 3D images of the absolute phases for the test-image 3: noisy phase (Fig.15a), phase reconstruction given by CD-BM3D (Fig.15b) and by ImRe-BM3D (Fig.15c). These absolute phases are obtained by the PUMA algorithm [36] applied to the noisy and reconstructed wrapped phases. It is clear

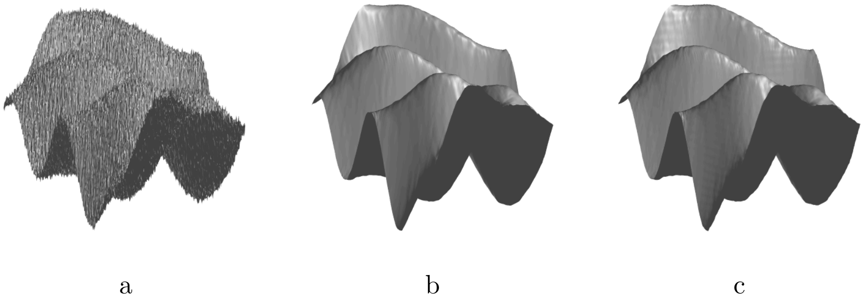


Figure 15: Test image 3, σ_{φ_z}=0.9: a) noisy absolute phase, $RMSE_{\varphi_{abs}}$=1.08, b) phase given by CD-BM3D, $RMSE_{\varphi_{abs}}$=0.15, c) phase given by ImRe-BM3D, $RMSE_{\varphi_{abs}}$=0.16.

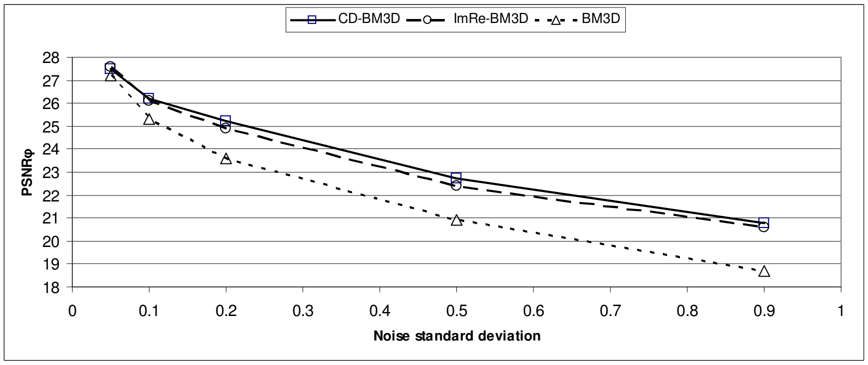


Figure 16: Comparison of CD-BM3D, ImRe-BM3D and BM3D as functions of the noise standard deviation σ_{φ_z} for the entire set of the test images.

that both CD-BM3D and ImRe-BM3D show very good results despite the very high level of noise in observations.

4.5.1. Comparison of the algorithms for different noise levels

Let us analyze the averaged $PSNR_\varphi$ as a function of σ_{φ_z}. Fig.16 shows the results obtained for the entire set of the test images $1-20$, $PSNR_\varphi$ is calculated as the mean of $PSNR_\varphi$ obtained for the test images.

First of all, the curves in Fig.16 indicate a better accuracy of the HOSVD based algorithms versus BM3D. For noisy data, these algorithms outperform BM3D by almost 2 dB. Comparing the HOSVD based algorithms, note that the accuracy of CD-BM3D is higher that the accuracy of ImRe-BM3D.

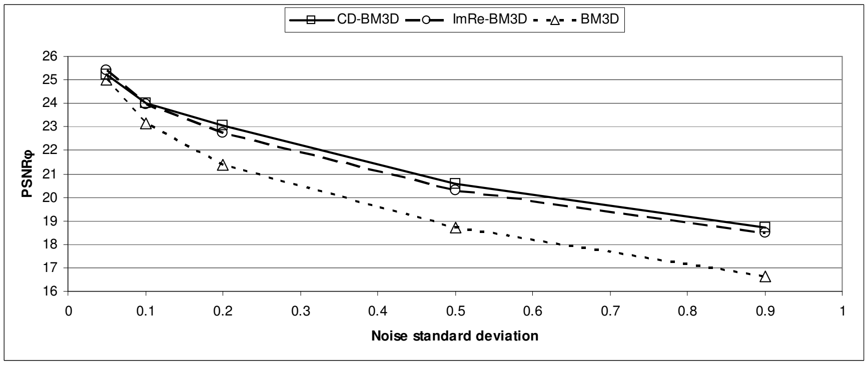


Figure 17: Comparison of CD-BM3D, ImRe-BM3D and BM3D as functions of the noise standard deviation σ_{φ_z} for the wrapped phase test images, 1 - 12.

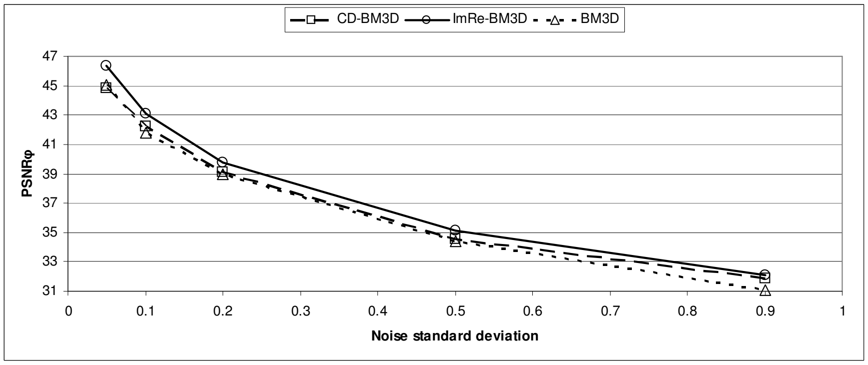


Figure 18: Comparison of CD-BM3D, ImRe-BM3D and BM3D as functions of the noise standard deviation σ_{φ_z} for the interferometric phase test images, 1 - 12.

Figs.17 and 18 allow a more detailed analysis because these averaged $PSNR_{\varphi}$ are calculated separately for the wrapped phase test images $1-12$ in Fig.17 and for the interferometric phase test images $13-20$ in Fig.18. Concerning comparison of CD-BM3D with ImRe-BM3D: the former algorithm shows better results with an improvement of about 0.2 dB for the wrapped phases, while the later algorithm is better for the interferometric phase tests with an improvement of more than 1.0 dB (Fig.18).

We may note the essential advantage of the HOSVD based algorithm versus BM3D for the wrapped phase tests (Fig.17), while for the interferometric phase

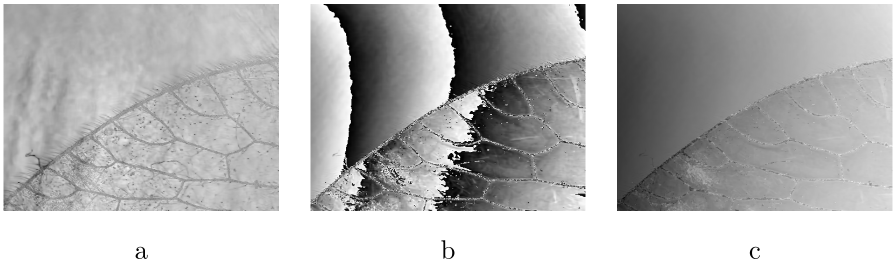


Figure 19: Fly's wing amplitude and phase reconstructions. From left-to-right: amplitude (a), wrapped phase (b), absolute (unwrapped) phase (c).

tests in Fig.18 BM3D sometimes gives results close to those by CD-BM3D with the definite advantage of CD-BM3D for the noisier data.

In conclusion, we may note that in average, for the wrapped phase test-images, CD-BM3D outperforms ImRe-BM3D by about 0.2 dB. At the same time, ImRe-BM3D performs better than CD-BM3D for the interferometric phase test-images with an advantage of about 1.0 dB for the low noise level.

4.6. Experimental data processing

We wish to complete this section, dealing with the simulation study of the algorithms, by showing phase and amplitude reconstructions obtained for the experimental data in Fig.1. This test is of interest due to reality of a phase/amplitude dependence and strong noise well seen in the phase and amplitude data. In Fig.19, the reconstructions produced by CD-BM3D, are demonstrated: amplitude, wrapped phase and absolute phase from left-to-right, respectively.

The evaluation of these results can be only qualitative because the true phase and amplitude of the fly's wing are unknown. Thus, we are not able to be sure, for example, how to interpret the multiple black spots in the amplitude reconstruction: noise, optical artifacts or real spots on the wing. However qualitatively, the noise suppression is obvious and the delicate features such as thin black hairs along the edge of the wing are preserved and well seen. The fringes in the wrapped phase images are due to the nonuniformity of the laser beam. These fringes disappear in the absolute phase image, Fig.19c, with the absolute phase range from 1.7 to 25.2 radians. Comparing the wrapped phase reconstruction with the experimental wrapped phase we may conclude that the filtering is efficient as the smooth fringes are obtained, and, in the same time, the thin lines observed in the phase data are preserved.

Processing this data with another our algorithms demonstrate more or less similar results.

4.7. Computational complexity of the algorithms

The theoretical complexity analysis produced in [32] for the real domain BM3D incorporating HOSVD is completely applicable to the algorithms considered in this paper. In particular, it is shown by this analysis that provided fixed parameters of the algorithms the computational time is proportional to the image size.

Here we characterize the computational complexity by the computational time tested in our experiments. Parameters of the computer and the software: Processor Intel (R) Core(TM) i7-4790 CPU 3.6 GHz; Operating system Windows 7, 64-bit; Matlab R2016a, 64-bit.

The following computational time (in sec.) is required for processing image 256×256: CD-BM3D without Wiener and with Wiener filtering 21.5 sec. and 43.5 sec., respectively; ImRe-BM3D without and with Wiener filtering: 27.2 sec. and 55.1 sec., respectively. Thus, the computational complexity of CD-BM3D and ImRe-BM3D is more or less the same.

5. Conclusion

In complex domain, sparsity can be understood in terms of three different variables: complex-valued (straightforward approach); phase and amplitude; real and imaginary parts of a complex-valued variable. The complex domain denoising is exploited as a test-problem for a comparison of different types of the sparsity modeling.

For implementation and algorithm design we use the nonlocal block-matching with the structural sparsity developed for 3D/4D tensors obtained by grouping similar patches of complex-valued data. This approach results in the following novel algorithms: CD-BM3D, ImRe-BM3D and PhAm-BM3D, where CD-BM3D uses 3D HOSVD dealing with complex-valued variables directly, while ImRe-BM3D and PhAm-BM3D use 4D HOSVD, respectively, dealing with the pairs phase/ amplitude and real/imaginary parts of complex valued HOSVD spectra. The algorithms are composed of two stages: hard-thresholding and Wiener filtering.

The comparative study shows that the best accuracy for denoising is achieved by the two competing algorithms: CD-BM3D and ImRe-BM3D. Thus, the complex domain sparsity and sparsity in the space of real/imaginary parts of complex-valued variables provide the best results.

Comparison versus the standard real domain BM3D algorithm applied separately to real and imaginary parts of the complex-valued data (see Fig.3) is definitely in favor of CD-BM3D and ImRe-BM3D.

Further development of the obtained results concerns applications of these novel algorithms and their modifications to problems with real-valued intensity observations, where the complex domain denoising appeared as sub-routines embedded in more complex iterative procedures.

6. Acknowledgment

This work is supported by Academy of Finland, project no. 287150, 2015-2019. The authors would like to thank the anonymous reviewers for helpful comments and Dr. Igor Shevkunov for the experimental data shown in Fig.1.

References

[1] R. K. Tyson, Principles of adaptive optics, CRC press, 2015.

[2] T. Kreis, Handbook of holographic interferometry: optical and digital methods, John Wiley & Sons, 2006.

[3] B. C. Kress, P. Meyrueis, Applied Digital Optics: from micro-optics to nanophotonics, John Wiley & Sons, 2009.

[4] S. Mallat, A wavelet tour of signal processing: the sparse way, Academic press, 2008.

[5] Q. Kemao, Windowed Fringe Pattern Analysis, SPIE Press Bellingham, Wash, USA, 2013.

[6] A. E. Shortt, T. J. Naughton, B. Javidi, Compression of digital holograms of three-dimensional objects using wavelets, Optics Express 14 (7) (2006) 2625–2630.

[7] I. W. Selesnick, R. G. Baraniuk, N. C. Kingsbury, The dual-tree complex wavelet transform, IEEE signal processing magazine 22 (6) (2005) 123–151.

[8] M. Liebling, T. Blu, M. Unser, Fresnelets: new multiresolution wavelet bases for digital holography, IEEE Transactions on image processing 12 (1) (2003) 29–43.

[9] J. Mairal, F. Bach, J. Ponce, et al., Sparse modeling for image and vision processing, Foundations and Trends® in Computer Graphics and Vision 8 (2-3) (2014) 85–283.

[10] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, M. Segev, Phase retrieval with application to optical imaging: a contemporary overview, IEEE Signal Processing Magazine 32 (3) (2015) 87–109.

[11] J. Zhang, D. Zhao, W. Gao, Group-based sparse representation for image restoration, IEEE Transactions on Image Processing 23 (8) (2014) 3336–3351.

[12] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Transactions on image processing 16 (8) (2007) 2080–2095.

[13] C. A. Metzler, A. Maleki, R. G. Baraniuk, From denoising to compressed sensing, IEEE Transactions on Information Theory 62 (9) (2016) 5117–5144.

[14] S. Mandal, A. Bhavsar, A. K. Sao, Noise adaptive super-resolution from single image via non-local mean and sparse representation, Signal Processing 132 (2017) 134–149.

[15] H. Zhong, K. Ma, Y. Zhou, Modified BM3D algorithm for image denoising using nonlocal centralization prior, Signal Processing 106 (2015) 342–347.

[16] Y. Wang, C. Xu, S. You, C. Xu, D. Tao, DCT regularized extreme visual recovery, IEEE Transactions on Image Processing.

[17] Y. Wang, C. Xu, C. Xu, D. Tao, Beyond RPCA: Flattening complex noise in the frequency domain, Assoc. Adv. Artif. Intell (2017) 500–505.

[18] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 208–220.

[19] I. F. Gorodnitsky, B. D. Rao, Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm, IEEE Transactions on signal processing 45 (3) (1997) 600–616.

[20] A. Sharif-Nassab, M. Kharratzadeh, M. Babaie-Zadeh, C. Jutten, How to use real-valued sparse recovery algorithms for complex-valued sparse recovery?, in: Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European, IEEE, 2012, pp. 849–853.

[21] E. Ollila, Multichannel sparse recovery of complex-valued signals using huber's criterion, in: Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa), 2015 3rd International Workshop on, IEEE, 2015, pp. 26–30.

[22] A. Bourquard, N. Pavillon, E. Bostan, C. Depeursinge, M. Unser, A practical inverse-problem approach to digital holographic reconstruction, Optics express 21 (3) (2013) 3417–3433.

[23] V. Katkovnik, J. Astola, High-accuracy wave field reconstruction: decoupled inverse imaging with sparse modeling of phase and amplitude, JOSA A 29 (1) (2012) 44–54.

[24] V. Katkovnik, J. Astola, Compressive sensing computational ghost imaging, JOSA A 29 (8) (2012) 1556–1567.

[25] V. Katkovnik, J. Astola, Sparse ptychographical coherent diffractive imaging from noisy measurements, JOSA A 30 (3) (2013) 367–379.

[26] C. A. Metzler, A. Maleki, R. G. Baraniuk, BM3D-PRGAMP: Compressive phase retrieval based on BM3D denoising, in: Image Processing (ICIP), 2016 IEEE International Conference on, IEEE, 2016, pp. 2504–2508.

[27] H. Hongxing, J. M. Bioucas-Dias, V. Katkovnik, Interferometric phase image estimation via sparse coding in the complex domain, IEEE Transactions on Geoscience and Remote Sensing 53 (5) (2015) 2587–2602.

[28] V. Katkovnik, K. Egiazarian, J. Bioucas-Dias, Phase imaging via sparse coding in the complex domain based on high-order svd and nonlocal bm3d techniques, in: Image Processing (ICIP), 2014 IEEE International Conference on, IEEE, 2014, pp. 4587–4591.

[29] V. Katkovnik, K. Egiazarian, Sparse phase imaging based on complex domain nonlocal bm3d techniques, Digital Signal Processing 63 (2017) 72–85.

[30] A. Danielyan, V. Katkovnik, K. Egiazarian, BM3D frames and variational image deblurring, IEEE Transactions on Image Processing 21 (4) (2012) 1715–1728.

[31] A. Foi, V. Katkovnik, K. Egiazarian, Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images, IEEE Transactions on Image Processing 16 (5) (2007) 1395–1411.

[32] A. Rajwade, A. Rangarajan, A. Banerjee, Image denoising using the higher order singular value decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (4) (2013) 849–862.

[33] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM journal on Matrix Analysis and Applications 21 (4) (2000) 1253–1278.

[34] L. R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.

[35] P. M. Kroonenberg, J. De Leeuw, Principal component analysis of three-mode data by means of alternating least squares algorithms, Psychometrika 45 (1) (1980) 69–97.

[36] J. M. Bioucas-Dias, G. Valadao, Phase unwrapping via graph cuts, IEEE Transactions on Image processing 16 (3) (2007) 698–709.

[37] D. L. Donoho, I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika (1994) 425–455.