TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Matteo Maggioni

Adaptive Nonlocal Signal Restoration and Enhancement Techniques for High-Dimensional Data

Julkaisu 1277 • Publication 1277

Tampere 2015

Tampereen teknillinen yliopisto. Julkaisu 1277
Tampere University of Technology. Publication 1277

Matteo Maggioni

Adaptive Nonlocal Signal Restoration and Enhancement Techniques for High-Dimensional Data

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB109, at Tampere University of Technology, on the 16th of January 2015, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2015

ISBN 978-952-15-3440-9 (printed)
ISBN 978-952-15-3464-5 (PDF)
ISSN 1459-2045

Abstract

The large number of practical applications involving digital images has motivated a significant interest towards restoration solutions that improve the visual quality of the data under the presence of various acquisition and compression artifacts. Digital images are the results of an acquisition process based on the measurement of a physical quantity of interest incident upon an imaging sensor over a specified period of time. The quantity of interest depends on the targeted imaging application. Common imaging sensors measure the number of photons impinging over a dense grid of photodetectors in order to produce an image similar to what is perceived by the human visual system. Different applications focus on the part of the electromagnetic spectrum not visible by the human visual system, and thus require different sensing technologies to form the image. In all cases, even with the advance of technology, raw data is invariably affected by a variety of inherent and external disturbing factors, such as the stochastic nature of the measurement processes or challenging sensing conditions, which may cause, e.g., noise, blur, geometrical distortion and color aberration.

In this thesis we introduce two filtering frameworks for video and volumetric data restoration based on the BM3D grouping and collaborative filtering paradigm. In its general form, the BM3D paradigm leverages the correlation present within a nonlocal *group* composed of mutually similar basic filtering elements, e.g., patches, to attain an enhanced sparse representation of the group in a suitable transform domain where the energy of the meaningful part of the signal can be

thus separated from that of the noise through coefficient shrinkage. We argue that the success of this approach largely depends on the form of the used basic filtering elements, which in turn define the subsequent spectral representation of the nonlocal group. Thus, the main contribution of this thesis consists in tailoring specific basic filtering elements to the the inherent characteristics of the processed data at hand. Specifically, we embed the local spatial correlation present in volumetric data through 3-D cubes, and the local spatial and temporal correlation present in videos through 3-D spatiotemporal volumes, i.e. sequences of 2-D blocks following a motion trajectory. The foundational aspect of this work is the analysis of the particular spectral representation of these elements. Specifically, our frameworks stack mutually similar 3-D patches along an additional fourth dimension, thus forming a 4-D data structure. By doing so, an effective group spectral description can be formed, as the phenomena acting along different dimensions in the data can be precisely localized along different spectral hyperplanes, and thus different filtering shrinkage strategies can be applied to different spectral coefficients to achieve the desired filtering results. This constitutes a decisive difference with the shrinkage traditionally employed in BM3D-algorithms, where different hyperplanes of the group spectrum are shrunk subject to the same degradation model.

Different image processing problems rely on different observation models and typically require specific algorithms to filter the corrupted data. As a consequent contribution of this thesis, we show that our high-dimensional filtering model allows to target heterogeneous noise models, e.g., characterized by spatial and temporal correlation, signal-dependent distributions, spatially varying statistics, and non-white power spectral densities, without essential modifications to the algorithm structure. As a result, we develop state-of-the-art methods for a variety of fundamental image processing problems, such as denoising, deblocking, enhancement, deflickering, and reconstruction, which also find practical applications in consumer, medical, and thermal imaging.

Foreword

In the summer of 2009, while I was preparing the last exams of the "Laurea Specialistica" degree in Computer Engineering in Politecnico di Milano, the advisor of one of my graduate courses, Giacomo Boracchi, unexpectedly asked me if I was interested in doing my M.Sc. thesis as an exchange student in Tampere University of Technology, Finland, and I accepted.

In January 2010 I arrived in Tampere. Still with the luggage in my hand, after struggling to find Tietotalo and its E corridor in the Department of Signal Processing, I was greeted by my soon-to-be supervisor Alessandro Foi who immediately offered me a tea and my first "pulla". This was a completely new setting for me. It was the first time living alone, first time living abroad, first time experiencing an everlasting minus-zero Celsius temperature, and the first time working within a research environment as well. The first dip in a frozen lake followed few weeks later. This exchange period under the guidance of Alessandro Foi and Karen Egiazarian, beyond crashing university clusters, resulted in an opportunity to continue as a doctoral student in the same department.

This thesis is the results of four years spent as a doctoral student with the Department of Signal Processing in Tampere University of Technology, from the fall of 2010 to the fall of 2014. My first thank goes to Alessandro Foi, whose invaluable advice, comments and friendly presence made possible the fulfillment of my work. Discussing any matter with Alessandro is always an enlightening experience. I also deeply thank Giacomo Boracchi for opening my path

to Finland and to research in general. I am indebted with Karen Egiazarian for welcoming me in his group during my exchange period and for always providing a positive research environment throughout my studies. I thank Vladimir Katkovnik for sharing his insights, Enrique Sánchez-Monge for his collaboration in my research, and Thrasos Pappas for giving me the opportunity to spend four months in Northwestern University (Chicago, IL, USA) as a visiting research fellow. I am grateful to the administrative staff of Tampere University of Technology for always supporting me in all daily matters, especially Ulla Siltaloppi, Virve Larmila, Noora Rotola-Pukkila, Susanna Anttila, Pirkko Ruotsalainen, Johanna Pernu, and Elina Orava.

Examining a doctoral thesis is certainly a laborious task, thus I wish to express my gratitude to the examiners and opponents of my doctoral thesis, Dr. Charles Kervrann, Prof. Marcelo Bertalmìo, and Prof. Pasi Fränti.

This research work would have not been possible without the financial support provided by Alessandro Foi and Karen Egiazarian, as well as the scholarships granted by Tampere Doctoral Programme in Information Science and Engineering (TISE) and by TUT President's Doctoral Programme. I wish to thank also KAUTE and Nokia foundation for granting me the scholarships that allowed me to support my research visit in Northwestern University, as well as to finalize my research activities.

Finland allowed me to meet an incredibly heterogeneous group of friends who have shared with me the joys and sorrows of living in Tampere. Their diverse personalities constantly motivate me to improve myself. I admire the confidence of Ugur "Ugo", the genuineness of Waqar, the wisdom of Marco "Marcone", the sarcastic take on life of Stefano "Stef", the ever-lasting positiveness of Andrea "Milo", the contagious sociability of Davide, the light-heartedness of Lucio, the absolute comedic sense of Bruno "Brno", and the dedication of Antonietta. Thank you all for being the way you are.

I would also like to mention my Evanstonian friends Graziano, Marco, and Chiara "Chiaramore", remembering our satirical anal-

ysis of the north-american culture and our raids in the canteens of Northwestern University never fails to put a smile on my face.

Being far from home has inevitably dimmed many friendships, but few others, the most important, endured the challenge. Domenico, Alberto, Andrea "Genti" and Stefano "Pionta" and I have been together since a very long time; our friendship has gone through a lot and despite living in different countries –and multiple timezones– we always manage to be part of each other's life.

My warmest thought goes to Silvia "Silli", who constantly encourages and inspires me to become a better person. Despite the large geographical distance that has been keeping us apart ever since we first met, our relationship is getting stronger with every passing day. I could not be more fortunate to have such a special person in my life. I also thank Rino, Barbara, and Greta for welcoming me in their family and making me immediately feel at home.

The final dedication goes to my family, my father Tiziano, my mother Patrizia, and my sister Francesca. I would not be who I am now without their constant, continuous, and unconditioned loving presence and support.

Matteo Maggioni
Tampere, December 2014

Contents

List of Publications

I. M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising using separable 4D nonlocal spatiotemporal transforms. In *Proceedings of the SPIE Electronic Imaging*, volume 7870, Jan. 2011

II. M. Maggioni and A. Foi. Nonlocal transform-domain denoising of volumetric data with groupwise adaptive variance estimation. In *Proceedings of the SPIE Electronic Imaging*, volume 8296, Jan. 2012

III. M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Transactions on Image Processing*, 21(9):3952–3966, Sep. 2012

IV. M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133, Jan. 2013

V. M. Maggioni, E. Sánchez-Monge, and A Foi. Joint removal of random and fixed-pattern noise through spatiotemporal video filtering. *IEEE Transactions on Image Processing*, 23(10):4282–4296, Oct. 2014

Notation and Abbreviations

For the sake of convenience, Table 1 and Table 2 contain the recurring mathematical notations and abbreviations used throughout the thesis alongside their corresponding meanings and explanation. References to relevant papers are also included when needed.

i.i.d.	independent and identical distributed
y	Noise-free data
z	Noisy data
$\hat{y}$	Estimated noise-free data
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and standard deviation σ
$\mathcal{P}(\lambda)$	Poisson distribution of parameter λ
$\mathcal{R}(\nu, \sigma)$	Rice distribution of parameters ν and σ [115]
$\mathbb{E}\{z\}$	Expected value of z
$\mathbb{E}\{z\|y\}$	Expected value of z conditioned on y
$\text{var}\{z\}$	Variance of z
$\text{var}\{z\|y\}$	Variance of z conditioned on y
$\mathcal{T}$	Transform operator
Υ	Shrinkage operator

Table 1. List of notations.

1-D, 2-D, 3-D	One-, Two-, Three- Dimensional
AWGN	Additive White Gaussian Noise
BM3D	Block-Matching and 3-D filtering [24]
CCD	Charge Coupled Semiconductor Devices
CMOS	Complementary Metal–Oxide Semiconductor
dB	Decibel
DCT	Discrete Cosine Transform
FPA	Focal Plane Arrays
FPN	Fixed-Pattern Noise
LWIR	Long Wave Infrared
MRI	Magnetic Resonance Imaging
MAD	Median Absolute Deviation [57, 41]
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NLM	Nonlocal Means [14]
PSD	Power Spectral Density
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index [124]
VST	Variance-Stabilizing Transformation [6]

Table 2. List of Abbreviations.

Chapter 1

Introduction

Imaging plays an ever increasing role for a plethora of fundamental applications in, e.g., science, engineering, and medicine. However, the raw data collected by imaging sensors is invariably contaminated by noise, blur, blocking, ringing, flickering, and other acquisition or compression artifacts. The massive growth in production as well as consumption of digital media demands for improved and more efficient acquisition processes, and thus motivates the interest in restoration or enhancement algorithms. Restoration algorithms aim to improve the quality of the observed signal in order to obtain a reliable estimate of the original (unknown) noise-free data without introducing filtering artifacts.

The statistical nature of the noise present within raw images typically depends on the particular imaging technique and sensor that has acquired the data. Currently, the most common imaging sensors are *Charge Coupled Semiconductor Devices* CCD [11] and *Complementary Metal–Oxide Semiconductor* CMOS [99] which are in essence arrays of photodetectors whose task is to accumulate and count impinging photons. Imaging techniques that detect light belonging to part of the electromagnetic spectrum not visible to the human visual system, rely on different sensing technologies or even different acquisition strategies altogether: prominent applications of this kind,

which are also of interest for the purpose of this thesis, are *Magnetic Resonance Imaging* MRI [29] and *Long Wave Infrared* LWIR thermography and hyperspectral imaging.

The difference in sensing technology, as well as in the physical processes involved, requires specific statistical models for the observed data. The physical process of photon counting in CCD and CMOS is inherently stochastic and typically modeled as a Poisson random variable whose random fluctuations are referred to as "shot noise" [67]. Different noise sources, such as thermal and electronic camera noise, can be approximated as white Gaussian random variables [60, 100]. Raw images acquired by focal plane arrays (FPA), such as *Complementary Metal–Oxide Semiconductor* CMOS [99] sensors or bolometers [110], tend to be also affected by fixed-pattern noise (FPN) because of the nonuniformities in the response of each FPA photodetector [95]. Magnetic resonance (MR) signals are detected by a system of scanner detectors that retrieve the phase and frequency information of the protons resonating after a short radio frequency pulse is emitted to the scanned body. The acquired MR signal is thus constituted by k-space samples having a real and imaginary part, and is assumed to be corrupted by signal-independent complex white Gaussian noise [55, 76, 37]. Once enough samples are acquired, the inverse Fourier transform can be eventually used to reconstruct the magnitude MR image which is assumed to be corrupted by Rician noise [73, 74].

Recently, the restoration community witnessed an exponential spread of a new breed of filtering methods relying, at some level, on the nonlocal self-similarity of small patches located at different positions within the data [62, 59, 69, 94]. The correlation of self-similar patches is then leveraged by some filtering operator to separate the noise-free portion of the observed data from the effects of the noise. The first successful nonlocal algorithm, the *NonLocal Means* (NLM) [14], depends on the nonlocal patch similarity to constitute an adaptive set of weights for the estimation of every pixel, which is thus obtained as a convex combination of, in principle, all other pixels in the

image. At the moment, the most effective filtering strategy is based on the so-called grouping and collaborative filtering paradigm [25], in which mutually similar patches extracted from the noisy data are first stacked in a nonlocal higher-dimensional structure called *group* and then filtered in transform domain. The local and nonlocal correlation of the group allows to compact the energy of the meaningful part of the signal in a small number of large-magnitude coefficients and thus the noise can be effectively filtered by thresholding the group spectrum.

1.1 Focus and Contribution of the Thesis

The focus of this thesis is the study and development of restoration frameworks designed for high-dimensional data, i.e. volumetric images and videos. The main contribution of this research work is focused on the data structures used during the filtering, as we argue that properly capturing and distinguishing the different types of correlation within the processed data is an essential factor of successful filtering methods. In particular, we embed the local spatial and temporal correlation in videos through sequences of 2-D blocks following a motion trajectory, i.e. 3-D spatiotemporal volumes; differently, we use 3-D cubes to capture the local spatial correlation in volumetric images. The 3-D cubes and volumes are the basic filtering elements in our frameworks and constitute the foundations of our filtering models. Nonlocality also plays a central role of our frameworks, as mutually similar 3-D basic filtering elements are stacked along an additional fourth dimension embodying the nonlocal correlation, thus forming 4-D data structures. The rationale behind this modeling consists in leveraging the transform-domain representations of the data such that the effects of the signal as well as those of the noise can be precisely localized along the different spectral dimensions. As a result, specific spectral hyperplanes, as well as specific

coefficients within each hyperplane, can be selectively manipulated to attain different filtering results.

The proposed frameworks have been leveraged to tackle different classic imaging processing problems such as video and volumetric data denoising, reconstruction of incomplete volumetric images, as well as the deblocking, deflickering, and enhancement of degraded sequences. As consequent contribution, we target these problems essentially using the same algorithmic structure even though the corresponding degradation models exhibit a significant difference. In particular, beside the classic case of i.i.d. white Gaussian noise, we consider noise characterized by spatial and temporal correlation, signal-dependent distributions, colored power spectral density, or spatially varying statistics.

1.2 Structure of the Thesis

Chapter 2 covers the necessary background for the remainder of the thesis by discussing the observation models for all the considered problems as well as by presenting a brief review of the current state of the art in signal restoration. In Chapter 3 we introduce the filter used for volumetric data denoising and reconstruction, whereas in Chapter 4 we present a video filtering framework and its corresponding applications. Final remarks and future research directions are eventually given in Chapter 5.

1.3 Link to Publications

The thesis encompasses five publications whose contribution is condensed in Chapter 3 and Chapter 4. In particular, the volumetric filter described in Chapter 3 has been presented in Publication IV, whereas its extension to the case of spatially varying noise has been described in Publication II. The video filter described in Chapter 4 has been originally presented in its basic form in Publication I

and later used for the filtering of different corrupting artifacts and noise characteristics in Publications III and V. The author of the thesis is the first author for all the aforementioned publications and thus in charge of the analysis, implementation, experimental evaluation, as well as the scientific writing of the presented methods. I wish especially to acknowledge the substantial contribution of Enrique Sánchez-Monge in the development and implementation of the algorithm presented in Publication V.

Chapter 2

Preliminaries

This chapter covers the concepts and notions that will serve as background for the remainder of the thesis. At first in Section 2.1 we briefly discuss the main sources of noise in digital images; then in Section 2.2 we present an overview of denoising methods, and in Section 2.4 we discuss the typical strategies employed for high-dimensional filtering. Subsequently, we dedicate Section 2.3 to the block-matching and collaborative filtering paradigm [25], and finally in Section 2.5 we briefly mention the most common metrics used for image quality assessment.

2.1 Observation Models

Digital data is acquired through a process typically (but not necessarily) involving a solid-state image sensor, such as CCD or CMOS, and an optical system of lenses. The lenses focus the light irradiated by the physical scene which is then captured by a 2-D array of opto-electronic semiconductor elements whose task is to absorb and count the impinging photons.

During the exposure, each cell in the 2-D array accumulates the electrons of the absorbed photons into electrical charge until the energy of the photons is large enough to convert such charge into an

analogue quantity (voltage). Subsequently, after such quantity is amplified, an analogue-to-digital converter discretizes (quantizes) the voltage into a digital number which finally corresponds to the raw intensity value at a determined pixel position. Raw data is invariably subjected to a variety of degradation factors; thus, a typical image processing pipeline involves denoising, demosaicing, enhancement, white balancing, gamma correction, compensation of lens distortion, as well as image compression, in order to generate the final image [108].

One of the most significant source of noise is the intrinsic randomness of the photon emission physical process: in fact, given a fixed exposure time, and even if the light irradiated from the scene is constant, there are still random fluctuations in the number of photons reaching each detector in the sensor. Furthermore, not all photons reaching the sensor are converted in electrical charge, and the percentage of those who do, i.e. the quantum efficiency, is a complex function of the photodetector depending the wavelength of the light as well as the absorption characteristics of the sensors [52]. These factors result in the so-called shot noise which is signal dependent and typically modeled by a Poisson distribution whose mean would be the ideal signal value [67].

Imperfections and physical characteristics of the imaging hardware induce other degradation in the form of thermal noise, flicker noise, readout noise and fixed-pattern noise. Specifically, thermal noise (or Johnson-Nyquist noise) consists in the erroneous accumulation of charge in the photodetectors caused by thermal agitation [60, 100].

Flicker noise (or $1/f$ noise or pink noise) is noise present in all electronic devices having a pink power spectral density spectrum, i.e. with power inversely proportional to the frequency f of the signal, and thus its effects are mostly visible in the low-frequency features of the signal [127].

Readout noise and fixed-pattern noise (FPN) are caused by spatial and temporal nonuniformities in the response of each photodetector

in the sensor, as well as imperfections in the amplifier and in the analogue-to-digital converter. FPN is a spatially correlated and temporally invariant phenomena which results in a structured pattern superimposed to the image and in extreme cases it can also manifest as impulse noise (or salt-and-pepper noise), i.e. pixels that can only correspond to either the maximum or the minimum value of the intensity range of the data [54].

A different observation model relates to compressed-sensing problems where an unknown signal of interest is observed through a limited number of linear functionals; this is particularly relevant for medical imaging applications such as computed tomography and magnetic resonance imaging (MRI). MRI uses a strong and uniform magnetic field to detect radio frequency information emitted from the scanned tissues. In particular, the hydrogen protons align along the magnetic field of the scanner and precess at a particular frequency modulated by a gradient field in the scanner. Then a radio frequency (RF) pulse excite a particular slice of protons in the scanned body, and within the scanned slice, two additional orthogonal gradients are used to encode frequency and phase information of the protons precession. The acquired MR signal is thus constituted by a set of k-space samples having a real and imaginary part, and is assumed to be corrupted by signal-independent complex white Gaussian noise. Different pulse sequences can be used to acquire the k-space samples, and such sequence is known as the k-space "trajectory" [74]. Since the acquisition process is rather slow, few k-space measurements are typically acquired. The inverse Fourier transform can be used to reconstruct the image, and the reconstructed magnitude image is thus modeled with a Rician distribution possibly having spatially varying statistics [55, 76, 37].

In this section we describe different observation models for data corrupted by Gaussian noise (Section 2.1.1), signal-dependent noise (Section 2.1.2), colored noise (Section 2.1.3), as well as the observation model for the compressed sensing problem (Section 2.1.4).

2.1.1 Gaussian Noise

Analyzing individually all different types of noise corrupting the sensed data is not practicable, however the central limit theorem (CLT) [103] states that, under mild conditions, the normalized sum of k independent random variables $Z_1, \cdots, Z_k$ each having individual mean μ_i and variance σ_i^2, defined as

$$\frac{1}{s}\sum_{i=1}^{k}\left(Z_i - \mu_i\right), \tag{2.1}$$

with $s^2 = \sum_{i=1}^{k}\sigma_i^2$, converges in distribution to the standard normal distribution $\mathcal{N}(0,1)$ as $k \to \infty$. Thus, the CLT allows us to consider the combined effects of different heterogeneous random sources as a single random variable following a Gaussian distribution. This fact, with the additional assumption of noise additivity, is the foundation of the most common observation model in the field of image processing [54], i.e. the i.i.d. additive white Gaussian noise (AWGN) model

$$z(x) = y(x) + \eta(x), \tag{2.2}$$

where z is the noisy data, y is the unknown noise-free data, $x \in X \subset \mathbb{Z}^d$ is a d-dimensional pixel coordinate denoting a position in the domain X, and $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random variable having zero mean and standard deviation σ. Estimators for the standard deviation σ have been thoroughly studied especially in the context of transform-domain representations [38, 30]. The model (2.2) has been shown to be a good approximation for the effects caused by signal-independent noise sources [98], however, in practice, the noise corrupting raw data has a form that invalidates the AWGN assumptions.

2.1.2 Signal-Dependent Noise

In this section, we explore the most common observation models characterized by noise components having distributions different than the

sole Gaussian featured in the classic AWGN model (2.2); specifically we discuss the Poissonian noise model, a mixed model comprising both Poissonian and Gaussian noise, and the Rician noise model.

Observe that signal-dependent observations can undergo a variance-stabilizing transformation (VST) which transform the variance of the noise to an almost-constant signal-independent value and thus allow the use of traditional homoskedastic algorithms [6]. Specifically, at first the noisy data is stabilized by a VST, then the stabilized data is denoised by a homoskedastic filter (e.g., designed for AWGN) using a constant value of standard deviation, and finally an inverse VST, denoted as VST^{-1}, is applied to the filtered data to obtain the final estimate. Note that VST^{-1} is not the trivial algebraic inverse of VST as it should compensate the bias induced by the nonlinearities of the forward transformation. Different unbiased VST formulae have been proposed for Poissonian noise [86, 85], Poisson-Gaussian noise [88], and Rician noise [47].

Poissonian Noise

A classic model alternative to (2.2) ignores the signal-independent corrupting sources and assumes that all the noise within the raw data is due to signal-dependent factors following a Poissonian distribution. Formally, each observation $z(x)$ in the noisy signal can be defined as an independent random variable drawn from a Poissonian distribution

$$z(x) \sim \mathcal{P}\big(y(x)\big), \tag{2.3}$$

where the noise-free value $y(x) \geq 0$ is the distribution parameter. The aim of restoration algorithms is to estimate the parameter $y(x)$, i.e. the expected value of the Poissonian variable (2.3). However, the expected value of (2.3) is also equal to its variance

$$\mathbb{E}\big\{z(x)|y(x)\big\} = \mathrm{var}\big\{z(x)|y(x)\big\} = y(x),$$

thus showing that the Poissonian noise

$$\eta(x) = z(x) - \mathbb{E}\big\{z(x)|y(x)\big\}$$

is signal dependent. From the statistics of (2.3), $\mathbb{E}\{\eta(x)|y(x)\} = 0$ and $\text{var}\{\eta(x)|y(x)\} = y(x)$, we note that the signal-to-noise ratio grows with the square root of the signal: thus, despite the standard deviation grows with the signal, the effects of the noise becomes relatively weaker as the signal intensity increases (and vice versa).

Poissonian noise might be modeled with a special signal-dependent Gaussian distribution $\mathcal{N}(0, y(x))$; this approximation is rather accurate because, whenever the photon count is large enough, a Poissonian distribution approaches a Gaussian [51].

Poissonian-Gaussian Noise

A more accurate observation model should simultaneously consider both signal-dependent and signal-independent noise sources. This is accomplished by combining two mutually independent components: a multiplicative Poissonian variable describing the shot noise, and an additive Gaussian variable capturing the effects of all other noise factors [72, 51]. Formally the Poissonian-Gaussian model is denoted as [51]

$$z(x) = \chi p(x) + \eta_{\mathcal{N}}(x), \tag{2.4}$$

where each observation $z(x)$ is the realization of a random Poissonian process $p(x) \sim \mathcal{P}(y(x))$, whose expectation is the noise-free value $y(x)$, scaled by a positive gain value $\chi > 0$ and corrupted by Gaussian noise $\eta_{\mathcal{N}}(\cdot) \sim \mathcal{N}(0, \sigma^2)$. The total variance for (2.4) is signal dependent and has an affine relation with the unknown parameters χ and σ^2 which depends on the hardware characteristic of the imaging sensor and on the acquisition settings such as quantum efficiency, pedestal parameter, analog gain, and ISO value [51]. The affine parameters can be robustly estimated from a single raw image [51].

We can note that the signal-to-noise ratio of $z(x)$ grows linearly with the signal, and thus, in the case of small signal values, it is bounded by the signal-independent components whereas the signal-dependent one dominates at large intensities. A more sophisticated

version of (2.4) also takes into account the effects of clipping, i.e. under- and over-exposures of the signal caused by the limited dynamic range of the sensor [51, 46].

Rician Noise

Raw complex MR data acquired in transform domain (k-space) can be modeled as a complex Gaussian distribution with a diagonal covariance matrix; the MR image can be obtained by applying an inverse Fourier transform, which, being linear and orthonormal, preserves the i.i.d. Gaussianity. However, the extraction of the magnitude of the complex MR image involves nonlinear operations and hence changes the distribution of the noise. Specifically, noise in magnitude MR images is assumed to follow a Rician distribution [55, 76, 37] whose observation model can be formally defined as [47]

$$z(x) = \sqrt{\left(c_r y(x) + \sigma(x)\eta_r(x)\right)^2 + \left(c_i y(x) + \sigma(x)\eta_i(x)\right)^2}, \qquad (2.5)$$

where x is again a d-dimensional coordinate belonging to the domain X of the image, c_r and c_i are arbitrary constants such that $0 \leq c_r, c_i \leq 1 = c_r^2 + c_i^2$, and $\eta_r(\cdot), \eta_i(\cdot) \sim \mathcal{N}(0,1)$ are i.i.d. random variables following the standard normal distribution. In this way, $z(x) \sim \mathcal{R}\left(y(x), \sigma(x)\right)$ represents the raw magnitude of the data modeled as a Rician distribution $\mathcal{R}$ of parameters y and $\sigma : X \to \mathbb{R}^+$, which denote the (unknown) original noise-free signal and the spatially varying standard deviation, respectively. A thorough presentation of the Rice distribution in the context of MRI together with derivations of estimators of the distribution parameters can be found e.g. in [115, 47, 87, 88, 37].

2.1.3 Colored Noise

So far the noise considered in all observation models, despite having different distributions, has always been characterized as white, i.e. its power spectral density (PSD) is always assumed to be constant in

frequency. The term "white" echoes the physical definition of white light, as white light contains nearly every wavelength of the visible electromagnetic spectrum in equal proportion.

Formally, the PSD is defined as the frequency representation (e.g., Fourier domain) of the autocorrelation function, and thus describes how the energy of the signal is distributed in frequency domain. In general, if the signal is correlated, its PSD exhibits some structure; conversely, the more unpredictable is the process, the more spread is its PSD. A random process is called white noise if its PSD is flat or, equivalently, its autocorrelation function is shaped as a Dirac delta. The AWGN model (2.2) includes the even stronger assumptions of statistical independence which thus implies uncorrelated noise. Differently, the presence of statistical dependencies between noise samples results in a different autocorrelation functions and hence a non-constant PSDs. Non-constant PSDs are helpful to model the spatial correlation characteristics caused by pink noise, fixed-pattern noise, or even post-processing techniques such as interpolation, demosaicing, enhancement, and compression. A precise modeling of non-constant noise PSDs needs to be taken into account in order to properly match the underlying sensor characteristics and/or the reconstruction processes forming the image [54]. Therefore, observation models featuring colored PSDs are needed by denoising methods to effectively handle the correlation within the noise.

The PSD is commonly estimated resorting to parametric or non-parametric (also known as classical) approaches. The former strategies hypothesize a model for the data in order to establish a parametric formulation for its spectrum which thus should ease the estimation task. The latter ones rely directly on the distribution of the power of the signal over frequency to estimate the PSD without any assumption on the structure of the data. The spectral density is typically estimated from the squared magnitude of Fourier coefficients. A complete overview of spectral estimation can be found, e.g., in [119]. However, when estimating the PSD of the noise from noisy signals, one should take into account that the estimated PSD would

not only capture the power spectrum of the noise but would be also affected by the structures (e.g., edges and textures) of the underlying noise-free data, thus yielding biased results. In such cases, the estimation algorithms should make use of uniform segments of patches containing only noise or, whenever such patches are unavailable or not in sufficient number, should leverage multiscale transforms to selectively extract the information better suited to describe the noise [106, 102].

2.1.4 Compressed Sensing

Compressed sensing studies the conditions under which a signal can be perfectly reconstructed using a fewer number of samples than what is required by the Nyquist-Shannon sampling theorem. This is motivated by the desire to minimize the number of acquired samples only during the acquisition stage, instead of wastefully acquiring all redundant information from the original signal. This is particularly relevant in applications where data acquisition is expensive or time consuming. In this context, we are primarily interested in an observation model

$$\theta = \mathcal{T}(f) + \eta, \tag{2.6}$$

where the representations θ of the original signal f depend on a linear sensing operator $\mathcal{T}$, being η the corrupting noise in the system. The model (2.6) allows to describe different sensing modalities such as the MRI acquisition process which considers $\mathcal{T}$ as the Fourier transform [74]. Essentially, a measurement is not a single point sample, but corresponds to some linear functional of the signal. Let Ω be the support of the available portion of θ. We define a sensing 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

$$\theta = \underbrace{S \cdot \theta}_{\theta_1} + \underbrace{(1 - S) \cdot \theta}_{\theta_2}, \tag{2.7}$$

where θ_1 and θ_2 are the observed (known) and unobserved (unknown) portion of the spectrum θ, respectively. Our goal is to recover (reconstruct) an estimate $\tilde{f}$ of the unknown f from the observed incomplete and noisy measurements θ_1. Note that if we had the complete spectrum θ, we could trivially obtain $\tilde{f}$ by applying the inverse transformation on the complete noisy spectrum as $|\mathcal{T}^{-1}(\theta)|$.

In general a direct application of the inverse operator $\mathcal{T}^{-1}$ cannot reconstruct the original signal because we consider cases where the available data is much smaller than what is required according to standard sampling techniques. However, stable (and even exact) reconstruction is made possible by assuming that the signal can be sparsely represented with respect to some suitable basis, and that the measurement basis is sufficiently "incoherent", i.e. radically different, with respect to the basis in which the signal is sparse. Thus, the signal of interest should be compactly represented in a suitable domain, and, within the same domain, the representation of sensing operator, unlike that of the signal, should be extremely dense; in fact, if the two basis are somehow too closely correlated, it would not be possible to recover the signal from few measurements. For example, the time-frequency domain pair enjoys maximum incoherence.

Compressed sensing is a two-step procedure, at first the information of the signal is compressed in few coefficients by using a stable sensing basis, and then a reconstruction procedure is used to recover the original signal from such sparse measurements. The reconstruction is an underdetermined problem admitting infinite solutions that can be however solved exactly with high probability through the fundamental constraint that the original signal is sparse in a domain incoherent with respect to the measurement basis. The smaller the coherence between sensing and representation basis, the fewer samples are needed [17, 39]. The solution of such underdetermined system can be found by minimizing the number of its non-zero components, i.e. the ℓ_0-norm of the solution which however is numerical unstable and computationally intractable problem. Thus, minimization of the ℓ_1-norm is typically used because such formulation can be efficiently

solved by linear programming being a convex optimization problem and it has proven to lead to the exact recovery of sparse signals with high probability [17].

2.2 Overview of Denoising Methods

In this section we discuss the denoising techniques proposed in the literature relevant in the scope of this thesis. The focus is here mainly given to images because we are interested in the foundational aspects of the denoising problem, but let us note that similar methods can be applied to higher-dimensional signals as well; in particular, following the categorization of [62], we discuss local and nonlocal methods (Section 2.2.1), transform-domain filtering (Section 2.2.2), and multipoint estimation (Section 2.2.3). We finally conclude with a brief discussion on the optimal performance bounds of the image denoising problem (Section 2.2.4).

2.2.1 Local and Nonlocal Filtering

A denoising algorithm is called *local* if an estimate of the noise-free signal is obtained through a local combination of the noisy data using weights which depends on some relation between the estimation point and the observation points [97, 126]. This strategy is incarnated in its basic form by, e.g., a limited bandwidth Gaussian smoothing kernel. A different strategy relies on modeling local image neighborhoods with adaptive local polynomial approximation (LPA) kernels [20, 4]. A prominent example of local denoising algorithm is the bilateral filter [120], which combines both the structure information and the photometric similarity during the filtering thus awarding larger weights to the most similar pixels. Local techniques do not take into account the information of the data falling outside the support of the chosen denoising kernel, and thus are not able to exploit the high degree of auto-correlation and repeated patterns at different location within natural signals [111, 117]. Conversely, imaging methods are

called *nonlocal* whenever the redundancy and self-similarity of the data is leveraged during the denoising [62, 69, 94].

The nonlocal paradigm, pioneered within the context of texture synthesis in [43], in the recent past has been one of the most influential ideas in the field of image restoration, as more than a thousand papers on this subject can be currently found in the literature [69]. The idea of reducing noise from the self-similarity of the data has been briefly discussed in the technical report [35], but the first algorithm for image denoising embedding the nonlocality principle is considered to be *NonLocal Means* (NLM) [14]. A method embodying the same essential principles has been independently presented in [5] where the authors propose to restore images using the similarity of the content within different image patches. Intuitively, NLM replaces the values in a noisy observation at a given reference pixel by an adaptive convex combination including –in principle– all pixels in the image, and the weights of the combination depend on the similarity between local patches associated to the reference and target pixels. In particular, the similarity is measured as a windowed quadratic point-by-point distance, and, naturally, the higher is the similarity the larger are the corresponding weight. This strategy allows all pixels to contribute during the estimation of every other point in the image, and even distant pixels can have a large contribution to the final estimate provided that they exhibit a sufficient similarity with the reference one. In practice, the nonlocal search is restricted to smaller neighborhoods because, beside increasing the computational cost, an exhaustive search might even lead to performance losses [113]. Many modifications and extensions of NLM have been proposed. In particular, adaptive mechanisms based on local image statistics can be used to estimate the aggregation weights and the size of the search neighborhoods [63, 64], as well as the shape of the patches [36].

2.2.2 Transform-Domain Filtering and Sparse Representation

A significant interest has been given to approaches that are able to compact the redundancy and self-similarity of natural images by disassembling the data with respect to a specific set of elementary basis functions. In fact, signals that admit a sparse representation within a suitable transform domain can be entirely described using a small set of transform coefficients. Popular transform operators decompose the data into oscillatory waveforms which eventually allow to provide a sparse representation for certain class of signals. For example the DCT or Fourier transform are efficient in describing uniformly regular signals, and the Wavelet transform can also sparsely represent localized and/or transient phenomena [34, 89].

The sparsity induced by a decorrelating transformation is commonly exploited by thresholding the spectrum of the noisy data in transform domain. Such strategy is composed of three steps: at first a forward transformation is applied to the noisy data, then the spectrum coefficients are thresholded following a particular nonlinear shrinkage operator, and finally the inverse transformation is applied to the thresholded spectrum. The complete process can be formally defined as

$$\hat{y} = \mathcal{T}^{-1}\Big(\Upsilon\big(\mathcal{T}(z)\big)\Big), \tag{2.8}$$

being z and $\hat{y}$ the noisy and estimated data, $\mathcal{T}$ a chosen decorrelating transform operator, and Υ a shrinkage operator. The threshold operator should preserve the energy of the signal while at the same time suppressing that of the noise. This is achieved by using a decorrelating transform $\mathcal{T}$ which should compact the significant information of the signal in a small number of large magnitude coefficients, and spread the effect of the noise in the the remaining coefficients having as small magnitude as possible. This is often referred to as energy compaction.

In [40, 38, 41], multiscale wavelet transforms are used to decorre-

late images corrupted by Gaussian noise, and the filtering is achieved by hard- or soft-thresholding the transformed spectrum. The choice of the threshold value has a significant impact in the efficacy on the denoising procedure, and, if the exact form of neither the underlying noise-free signal nor the statistics of the noise are known, its setting becomes a non-trivial task. Several approaches have been proposed to select a proper threshold, one of the most widely adopted is the "universal threshold" $\sigma\sqrt{2\log(n)}$ [40], being σ the standard deviation of the Gaussian noise and n the size of the data, which has been proven to be very close to an ideal solution [40]. Another popular shrinkage operator, optimal in the MSE sense and widely used in the remainder of this thesis, is the empirical Wiener filter defined as

$$\Upsilon\Big(\mathcal{T}\big(z\big)\Big) = \mathcal{T}(z) \cdot \frac{|\mathcal{T}(\hat{y})|^2}{|\mathcal{T}(\hat{y})|^2 + \sigma^2},$$

where z is the noisy data, $\hat{y}$ is an empirical noise-free estimate obtained, e.g., by prefiltering z, $\mathcal{T}$ is a transform operator, σ^2 is the variance of the corrupting additive noise, and $\cdot$ denotes element-wise multiplication [130].

Different approaches apply the transform operator on local image patches rather than globally on the whole image. In this context, the DCT transform is a well established tool because of its near-optimal decorrelating properties which are even close to those of the Karhunen-Loève transform (KLT). The KLT is a linear transform having basis functions that adapt to the statistical properties of the data, but, despite being optimal in the sense of signal decorrelation and energy compaction, its usage is limited because the KLT is not separable and the computation of its basis is very demanding. Thus, other transforms, such as the DCT or the Fourier transform, are more practicable alternatives. Particularly, in [56] the DCT is applied in a sliding window fashion on every $N \times N$ blocks in the image, and then each DCT-block spectrum is separately filtered in transform domain eventually providing an estimate of the block. However the performance decays in the presence of discontinuities in the data which are

not effectively described by the block DCT, thus in [49] the authors overcome this problem by utilizing a shape-adaptive DCT transform applied on blocks having anisotropic support determined by the local features in the image.

There is no single transform general enough to guarantee a good sparsification for all kinds of signal. A solution can be thus adapting the transformation to the known features of the processed data. This idea is leveraged in [45], where the authors present a technique based on dictionaries trained from either the noisy image or a database of natural images. The filtering is implemented within a Bayesian formulation whose prior consists in the existence of a representation of every patch in the noisy image which is sparse with respect to the trained dictionary. The method iteratively finds the optimal description of each patch as a sparse combination of atoms in the dictionary, and consequently updates the atoms in the same dictionary using the singular value decomposition (SVD) to better fit the data.

Finally, we cite the sophisticated strategy presented in [107], where the image is first disassembled in a set of subbands by a multi-scale multi-oriented wavelet transform, and then local neighboring coefficients within each subband are modeled as a scale mixture of Gaussians [123]. Hence, assuming AWGN, denoising of the transform coefficients is operated by Wiener filtering within a Bayesian least-squares formulation. Once all transform coefficients are estimated, the final noise-free image estimate can be reconstructed by inverting the original transform operator.

2.2.3 Multipoint Filtering

In general, a denoising algorithm uses a set of observation points during the estimation process at a any given (reference) position. Such algorithm is called a pointwise estimator if, despite using multiple points, the result of each estimation process consists of one single (reference) point. An example is NLM [14], which uses a set of similar patches to produce the estimation of a single pixel. Conversely, a

filtering algorithm is called multipoint if it gives an estimate for all points involved in the estimation process. A typical example is [56] which filters the data as image blocks in DCT domain and returns an estimate for all pixels in the transformed block. In other words, for each estimation process, multipoint methods return a set of estimates, whereas pointwise approaches return the estimate of the a single (reference) point [62].

Observe that, in the multipoint approach, the estimation for different reference points are likely to use overlapping sets of observation points. Thus, a typical multipoint filtering paradigm is composed of three steps: at first some kind of data windowing is applied through spatial or transform domain analysis, then multipoint data estimation is performed within each window, and finally an aggregation function is used to fuse all overlapping estimates [62]. This redundancy typically yields to more accurate estimation results because, in principle, it allows to overcome the filtering artifacts due to singularities in the signal, without necessarily resorting to shape-adaptive techniques, translational invariant filtering such as cycle spinning [21], nor transforms tailored to specific nonuniformities in the data (e.g., directional wavelets [2], curvelets [16, 118], etc.).

The design of an optimal aggregation function, which combines different estimates into a single final value, is not a trivial task. Typically, the final estimate for each point in the data consists in a convex combination of all the available estimates for that point. The easiest formulation for the weights in the convex combination, leveraged by many works in the literature [62], simply awards equal weights to all contributions, however a significant advantage can be achieved by promoting the values originating by the most reliable estimates [101, 56]. Hence an effective aggregation strategy, also used in the remainder of this thesis, assigns weights inversely proportional to the total mean variance of the estimate, which is approximated from the spectrum coefficients retained after shrinkage [62]. For example, in the case of hard thresholding, the variance of the estimate can be approximated from the number of the retained non-zero coefficients.

2.2.4 Optimal Denoising Bounds

Multipoint methods based on patch-wise processing seem to provide the best denoising results. In particular, the current state of the art in image denoising have been even shown to achieve near-optimal theoretical denoising results [19, 70, 69, 94]. The fundamental question that these papers aim to answer also titles [19]: "Is Denoising Dead?". In order to provide an answer, in [70] the authors designs an algorithm based on NLM that uses a database of 10^{10} patches, previously extracted from thousands of different natural images, instead of the image itself only. Through extensive experimentation on such patch space, the authors evaluate the best-possible estimation error attainable by any patch-based denoising methods. The final claim of [70] states that the current state-of-the-art patch-based methods are close to optimality; conversely in [19] it is shown that there is still room for improvement. However such analysis does not take into account aggregation strategies that are used to combine overlapping estimates of different patches [69], even tough this practice is proven to provide a substantial improvement in the denoising performance as discussed in Section 2.2.3. In conclusion, a complete axiomatic theory for the image restoration problem is extremely hard (and may be not possible) to formulate. Notwithstanding that, according to [19, 70] the current state of the is arguably very close to achieve optimum performance.

2.3 Block-Matching and Collaborative Filtering

In this section we discuss the Block-Matching and Collaborative Filtering (BM3D) paradigm [25], being a foundational aspect for the remainder of this thesis. The BM3D paradigm encompasses three steps, namely grouping, collaborative filtering, and aggregation. These steps are performed for every (reference) patch in the image. The

rationale behind BM3D consists in exploiting the local and nonlocal correlation within the data to generate an enhanced sparse representation in transform domain, and then leveraging the overcompleteness of the estimated data to eventually produce the final estimate.

2.3.1 Grouping

During the grouping, $d+1$-dimensional data structures, i.e. "groups", are built from mutually similar d-dimensional patches (e.g., 2-D blocks) extracted from the noisy data. The groups are characterized by local correlations between the pixels within each patch, as well as nonlocal correlation between corresponding pixels of different patches. These groups are obtained by a nonlocal matching procedure which evaluates the similarity between the reference patch and any other patch in the data. The similarity is typically measured as the ℓ^2-norm of the patch difference but other metrics are of course admissible.

2.3.2 Collaborative Filtering

The correlation within and among the grouped patches enables an enhanced sparse representation of the group in transform domain. Thus, as we have already discussed for (2.8), denoising can be effectively achieved via coefficients shrinkage in the sparsifying transform domain. In BM3D, this is referred to as collaborative filtering, and consists of three steps: application of a linear $d+1$-dimensional transform to the group, thresholding of the group spectrum by coefficients shrinkage, and application of the inverse $d+1$-dimensional transform to obtain an estimate of all the grouped patches.

Collaborative filtering reveals fine details shared by the grouped patches while preserving their individual features because each of the grouped patches influences the filtering of all the others. The linear transform employed by the collaborative filtering is built as a separable decomposition of lower-dimensional linear transforms. The performance is greatly influenced by the choice of the used transform

operators, and those should include a constant basis function, i.e. the DC term [25]. After the coefficients shrinkage, only a small number of coefficients remain in the thresholded spectrum, and most of them are concentrated around the DC.

2.3.3 Aggregation

The collection of d-dimensional patch estimates is an overcomplete representation of the original data because estimates belonging to different groups, as well as estimates within each group, are likely to overlap. The redundancy is not predictable as it depends on the grouping and on the data, thus in order to compute a final estimate of the original signal, the overlapping patch estimates originating from all $(d+1)$-dimensional groups need to be aggregated. The aggregation is performed via a convex combination with adaptive weights depending on the total residual variance of the group, as motivated in Section 2.2.3. Intuitively, the sparser is the shrunk spectrum, the larger is the corresponding weight in the combination.

2.3.4 Implementation

The BM3D algorithm is implemented with two cascading stages, each including the aforementioned grouping, collaborative filtering, and aggregation [25]. In the first stage, the grouping of mutually similar patches is performed within the noisy data and coefficients shrinkage is implemented as a hard-thresholding operator with threshold value depending on the variance of the noise standard deviation. After aggregating all filtered groups, a basic estimate is produced. In the second stage, the basic estimate resulting from the first stage of filtering is used to refine the matching and build new groups: the grouping is repeated by testing the patch similarity within the basic estimate, and the coordinates of similar patches are used to build two groups, one formed by patches extracted from the noisy data and the other by patches extracted from the basic estimate. Collaborative

filtering is then implemented as an empirical Wiener filter applied on the noisy group using the corresponding groups extracted from the basic estimate as pilot signal. After aggregation of the overlapping estimates of the patches, the final denoised image is eventually obtained.

The BM3D paradigm, originally presented in [25], has also later extended to allow for shape-adaptive patches and improved transform operators based on principal component analysis [28]. This strategy has proven state-of-the-art performance and even near-optimal theoretical denoising results [19, 70]. The BM3D image model has also been applied to the raw data denoising [9, 33], color filtering [25], video denoising [24], joint image sharpening and denoising [26] via "alpha rooting" [1], image and video super-resolution [31], image deblurring [27, 32], and also noise estimation [30].

2.4 High-Dimensional Filtering

We focus on the problem on restoring high-dimensional imaging data such as 3-D volumetric images and videos. The degradation factors introduced in the beginning of this chapter still apply to this case, but the additional dimension of the data introduces further artifacts which thus need to be properly modeled and targeted during the filtering.

2.4.1 Volumetric Filtering

The literature on volumetric filtering is mainly focused on MR image restoration, as MRI is among the most prominent applications using volumetric data. One of the earliest approach in MRI denoising simply relies on Gaussian filtering [3], but has the obvious drawback of removing the high-frequency features of the signal together with the noise. Alternative classical approaches embed anisotropic diffusion filters [53] in the presence of either Guassian or Rician noise [68].

Transform-domain techniques have also been studied in the context of multiscale wavelet representation [104] or local sliding-window transforms [56]. In particular, the method in [56] uses a sliding overcomplete linear transforms, such as the DCT, and coefficient thresholding to estimate the noise-free image within local patches of the data. Recent advances in volumetric denoising combine and extend the approaches presented in [130, 56, 14] to nonlocal 3-D filtering as well as non-Gaussian noise removal. In particular, the most successful approaches in volumetric denoising embed the nonlocal paradigm [14], leveraging the self-similarity of higher-dimensional patches [129, 23, 90], and include a mechanism to correct the bias caused by the asymmetry of the Rician distribution in order to effectively handle Rician noise in the data. A similar technique has also been extended for the case of spatially varying noise levels, i.e. noise with non-uniform statistics [91]. In [22], the amount of filtering is adapted to the particular image content by aggregating a set of estimates obtained using different filtering parameters in multiscale transform domain. In [90], the authors use both voxel value and local mean to assess the similarity of 3-D patches, thus allowing for an efficient nonlocal matching procedure which is also rotationally invariant and resistant to noise.

2.4.2 Video Filtering

Video denoising filters exploit the spatiotemporal redundancy between consecutive frames [58, 125]. The similarity along the motion trajectories is typically much stronger than the nonlocal similarity existing within an individual frame because of the strong temporal smoothness present in videos [9]. Thus, we argue that the temporal dimension should be explicitly considered during the filtering, and failing to do so would result in suboptimal denoising results, or even generate temporal filtering artifacts especially whenever the same moving feature is inconsistently estimated along time [12].

Different strategies have been proposed to exploit the redundancy

present along the temporal dimension and typically include a motion estimator to compensate the data and a filter acting along the motion trajectories in spatial [42, 71] or transform domain [84, 61]. Particularly, in [71] denoising is achieved by integrating an optical flow operation with the nonlocal paradigm in spatiotemporal domain, whereas wavelet [61] or adaptive transforms [84] are used to induce sparsity and denoise the data in transform domain. A motion detection technique can be used to trigger spatial (e.g., frame-by-frame) filtering, if the temporal information is insufficient or not reliable enough [105]. We reckon that the estimation of motion is a hard and computationally intensive problem and it is further complicated by imperfections of the motion model, temporal discontinuities (e.g., occlusions in the scene), and by the presence of the noise [7]. Therefore, methods that do not explicitly account for motion information have also been investigated in, e.g., [66, 13, 112, 10, 24], where local spatial or spatiotemporal 3-D patches within the video are adaptively filtered in spatial or transform domain using the local and nonlocal information in the video.

The nonlocal paradigm is leveraged in video filtering by first finding mutually similar patches within a spatiotemporal search neighborhood, and then by estimating the noise-free data exploiting the information within matched patches. The size and shape of the spatiotemporal neighborhoods which in turn define the nonlocal weights in the combination [14] can be also adaptive [10]. Nonlocal self-similarity is embedded in the recent filter [59] where the noise is removed from a stack of similar patches through a low rank matrix completion problem solved with a nuclear norm minimization [15]. This approach has the advantage to require minimal assumptions on the corrupting noise, which can thus deviate from the classical white Gaussian distribution.

At the moment, one of the most successful strategies is provided by the V-BM3D algorithm [24], which is a filter based on the BM3D filtering paradigm introduced in Section 2.3. The innovative idea of V-BM3D lies in the grouping step, where the search for similar

blocks is not restricted only to a single image but it also covers several consecutive frames in order to simultaneously exploit spatial and temporal redundancy. In particular, since an exhaustive spatiotemporal search would be computationally not feasible, V-BM3D uses a technique based on a data-adaptive predictive-search block-matching procedure which progressively refines the position and size of the search neighborhoods using the information of the blocks matched in the previous frames.

2.5 Assessing Image Quality

In this thesis we mainly restrict to the peak signal-to-noise ratio (PSNR), being widely-used in the field of image restoration and thus allowing for an easy comparison with respect to methods proposed in the literature. The PSNR is formally defined in logarithmic scale as

$$PSNR = 10 \log_{10} \frac{I_{\max}^2}{MSE}, \tag{2.9}$$

being $I_{\max}$ the maximum intensity value of the signal, hence expressing the ratio between the maximum possible power of the signal versus the power of the corrupting noise as measured by the mean squared error (MSE)

$$MSE = \frac{1}{|X|} \sum_{x \in X} \left(y(x) - \hat{y}(x) \right)^2,$$

which corresponds to the dissimilarity magnitude between the original signal y and the estimated one $\hat{y}$ averaged over all image domain X, being $|X|$ the cardinality of X.

However, a high PSNR (or low MSE) value does not always correspond to a signal with perceptually high quality. Image quality assessment (IQA) aims at measuring the quality of a given image using objective metrics designed to agree with human visual judgment. This is by itself a difficult problem and still an open research

topic, and thus many IQA algorithms have been proposed by many researchers [18], with the final goal to define a procedural metric able to objectively measure the quality of different image estimates while also providing a quality assessment that correlates to human perception. In the remainder of this thesis, we make also use of objective metrics that are expected to be more consistent with human judgment, i.e. the structural similarity SSIM index [124] and the motion-based video integrity evaluation MOVIE index [114].

Chapter 3

Volumetric Filtering

In this chapter we introduce a denoising filter for volumetric data based on the BM3D filtering paradigm [25]. In the proposed algorithm, denoted BM4D, we naturally utilize cubes of voxels as basic filtering elements, and hence we form 4-D groups by stacking together mutually similar cubes. The fourth dimension, along which the cubes are stacked, embodies the nonlocal correlation across the data. The groups are collaboratively filtered by simultaneously exploiting the local correlation present among voxels in each cube as well as the nonlocal correlation between the corresponding voxels of different cubes. Thus, the spectrum of the group is highly sparse, leading to a very effective separation of signal and noise by coefficient shrinkage. After inverse transformation, we obtain the estimates of each grouped cube, which are then aggregated at their original locations using adaptive weights.

We apply the BM4D algorithm for noisy data corrupted by Gaussian as well as Rician noise, leveraging the VST approach proposed in [47]. Adaptive noise variance estimation is also implemented by exploiting the sparsity of the representation of the group in transform domain, where the local groupwise standard deviation is accurately estimated from the outcome of robust median operations applied to the coefficients of the group spectrum [30].

Additionally, we apply BM4D as a regularizer operator for the reconstruction of incomplete volumetric data. In several inverse imaging applications, and particularly in MRI, the observed (acquired) measurements are a severe subsample of a transform-domain representation of the original unknown signal. 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 [39, 74, 75, 122]. Differently, the proposed procedure addresses the reconstruction of volumetric data having non-zero phase from a set of incomplete and noisy transform-domain measurements, replacing the common parametric modeling of the solution with a nonparametric one implemented by the use of a spatially adaptive denoising filter. 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.

In Section 3.1 we will first introduce the formalization and implementation of the basic BM4D volumetric filter, and then in Section 3.2 and 3.3 we will present its application in volumetric data denoising and reconstruction, respectively.

3.1 Basic Algorithm

The basic BM4D algorithm comprises grouping, collaborative filtering and aggregation [25], with an optional additional step for the groupwise noise variance estimation, which is enabled whenever the

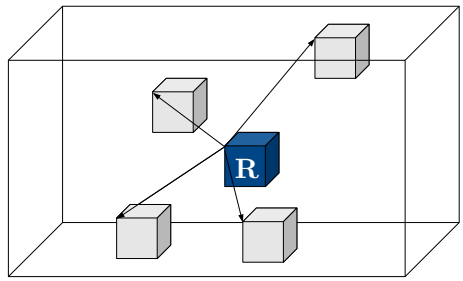


Figure 3.1. Schematic illustration of the BM4D grouping procedure. The reference cube "R" is shown in blue.

variance of the noise is unknown. In what follows, we describe the general steps of the algorithm for the filtering of data corrupted by either Gaussian (2.2) or Rician noise (2.5). Note that, the noise variance estimation can be also used to change the filtering strength in the presence of spatially varying noise, as the estimation is performed in a groupwise fashion and thus adapts to the local characteristics of the noise.

3.1.1 Grouping

In the grouping step, any given reference 3-D cube $C(x_R)$ of 3-D spatial coordinate $x_R \in X \subset \mathbb{Z}^3$ are extracted from the noisy data z and then tested for similarity against all cubes within a local 3-D neighborhood around the reference voxel x_R. The similarity between two blocks is typically measured using a distance metric, e.g., the ℓ_2-norm of the cubes difference, and two blocks are considered similar if such distance is smaller than or equal to a predefined threshold. A schematic illustration of the grouping is provided in Fig. 3.1.

As a result, for each reference cube $C(x_R)$, a group $G(x_R)$ is build by stacking together mutually similar 3-D cubes along an additional fourth dimension, hence creating a 4-D group.

3.1.2 Adaptive Groupwise Noise Variance Estimation

Assuming that the noise variance is slowly varying, and since the grouped cubes have typically nearby coordinates, we can reasonably treat the noise level within each group as a constant. Therefore, only a single noise variance estimate is needed for each group. A precise estimation of the variance is a crucial task, because the amount of filtering operated on the noisy observations is proportional to the strength of the corrupting noise.

After the application of a sparsifying $\mathcal{T}_{4\mathrm{D}}$ transform, the energy of the signal and that of the noise are well localized in the low- and high-frequencies portions of the group spectrum, respectively. Thus, in the case of Gaussian noise, an accurate groupwise variance estimation can be directly obtained from the median of absolute deviation [57, 40] (MAD) of the high-frequencies coefficients of the group spectrum [30]. Differently, if the noise follows a Rician distribution, we first need to estimate the mean-variance pair of the median value of the underlying noise-free group so that we can univocally and directly obtain a robust estimate of the scale parameter of the Rician noise in (2.5) [47].

3.1.3 Collaborative Filtering

Before the collaborative filtering, if the noise is Rician, a VST specifically designed for the Rice distribution [47] is applied to the group in order to remove the dependencies between the noise and the underlying data [6]. In this way, the stabilized group can be filtered using the constant standard deviation value induced by the VST.

During collaborative filtering, the group is first transformed by a decorrelating separable four-dimensional transform $\mathcal{T}_{4D}$, then the coefficients of the so-obtained spectrum are thresholded through a coefficient shrinkage operator (e.g., hard thresholding or Wiener filtering) scaled by the noise standard deviation level. An estimate of

the group is eventually produced by inverting the original 4-D transform, and therefore contains the individual estimates of each grouped cube.

Finally, in case of Rician noise, the filtered group undergoes the exact unbiased inverse VST [47] that simultaneously inverts the VST and produces an unbiased estimate of the underlying noise-free data. Observe that, in the case of Rician noise with uniform (non-spatially varying) standard deviation, equivalent results can be produced by first applying the VST globally on the volumetric image before the denoising, and then inverting the VST after the final estimate is obtained from the denoising filter.

3.1.4 Aggregation

Since the cubes in the different group estimates (as well as the cubes within the same group) are likely to overlap, we may have multiple estimates for the same voxel. Therefore the final volumetric estimate is obtained through a convex combination as explained in Section 2.3.3.

3.2 Volumetric Data Denoising

The denoising performance of BM4D are evaluated using a synthetic BrainWeb phantom [121] corrupted by synthetic noise having uniform or spatially varying variance accordingly to (2.2) and (2.5). In the case of spatially varying noise, we multiply such uniform noise realization by a volumetric noise modulation map [91]. Noise-free and noisy phantoms with uniform and spatially varying Gaussian noise are shown in Fig. 3.2, Fig. 1 (p. 97) in Publication II, and Fig. 3 (p. 121) in Publication IV.

Real cross-sectional MR data from the Open Access Series of Imaging Studies (OASIS) database [92] is also considered. The noise in this case is assumed to be Rician-distributed and its standard deviation, estimated as described in [47], is approximately $\sigma \approx 4\%$ of the

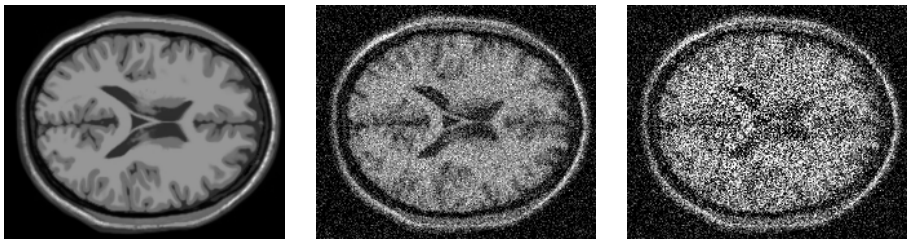


Figure 3.2. Noise-free (left), noisy cross-section of the BrainWeb phantom [121] corrupted by Gaussian noise with uniform standard deviation $\sigma = 15\%$ (center), and spatially varying standard deviation σ ranging between 15% and 45% (right). The standard deviation is defined with respect to the maximum intensity value of the noise-free data.

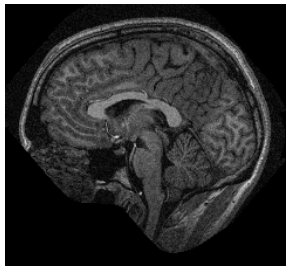


Figure 3.3. Cross-section of the OASIS phantom [92]. The noise is Rician-distributed and has approximately standard deviation $\sigma \approx 4\%$ of the maximum intensity value of the data.

maximum intensity value of the data. A cross-sections of such OASIS phantom is shown in Fig. 3.3 and Fig. 3 (p. 121) in Publication IV.

In Fig. 3.4, Fig. 2 (p. 98) in Publication II, and Fig. 4 (p. 123) in Publication IV we show the denoising results provided by BM4D for the denoising of the BrainWeb phantom corrupted by Gaussian noise having uniform or spatially varying statistics, as well as for the denoising of the OASIS phantom. Specifically, we use the BrainWeb phantom corrupted by synthetic Gaussian noise and the real OASIS

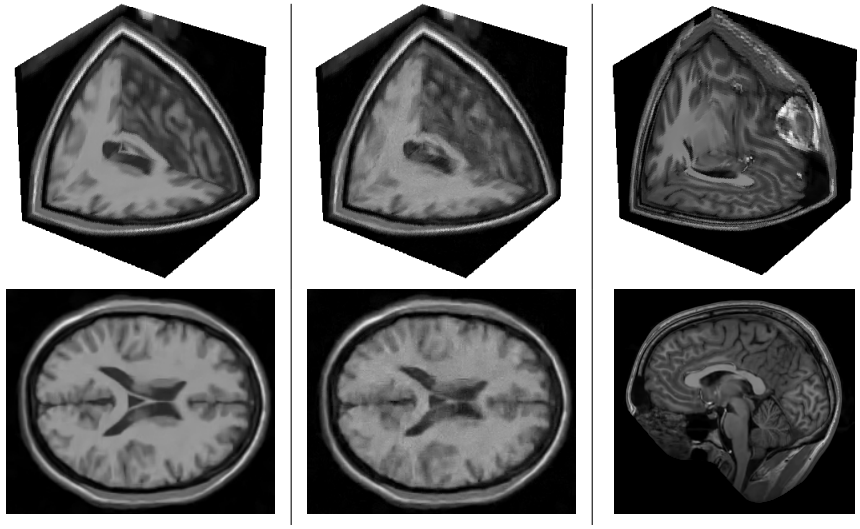


Figure 3.4. Denoising results of BM4D applied to the BrainWeb phantom corrupted by Gaussian noise with standard deviation $\sigma = 15\%$ (left), BrainWeb phantom corrupted by spatially varying Gaussian noise with standard deviation $\sigma \in [15\% \sim 45\%]$ (center), and the OASIS phantom corrupted by Rician noise with standard deviation $\sigma \approx 4\%$ (right).

phantom corrupted by noise assumed to follow a Rician distribution. 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.

The proposed BM4D has been proven to be the state of the art in volumetric filtering under the presence of either Rician and Gaussian noise with uniform or spatially varying statistics. Subjective and objective results, measured as the PSNR (2.9) and an extension of SSIM [124] to 3-D data [90], consistently provide the best visual and numeric performances for BM4D in all considered cases, as shown by Table II (p. 122) in Publication IV and Table 2 (p. 98) in Publication II.

3.3 Volumetric Data Reconstruction

The proposed BM4D filter can be leveraged as a regularizer operator for the reconstruction of data observed as incomplete and noisy measurements acquired in transform domain. The general form of the observation model follows (2.6), and for our purposes it can be specialized by setting $f = ye^{\imath\phi}$ being θ the transform-domain $\mathcal{T}$ representations of the unknown data having magnitude y and absolute (unwrapped) phase ϕ, and $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$ being i.i.d. complex Gaussian noise with zero mean and standard deviation σ. In practice, the transform operator $\mathcal{T}$ is the Fourier transform.

The reconstruction is carried out within an iterative process where an estimate of the unobserved portion of the spectrum is improved via a stochastic search driven by the action of the BM4D denoising filter [44, 32]. We recall from (2.7) that the only available data is the spectrum portion θ_1 measured through an operator S which acts as a MR k-space sensing trajectory.

The initial estimate of the unobserved portion of the spectrum θ_2 is set to zero, thus the initial estimate is generated by back-projection. Subsequently, for each iteration (k), the reconstruction is carried out through three cascading steps: noise addition (excitation), volumetric filtering, and data reconstruction. The iterative procedure can be either stopped after a pre-specified number of iterations, or whenever the current magnitude estimate does not differ significantly from that obtained in the previous iteration.

3.3.1 Noise Addition

The estimate of the unobserved portion of the spectrum $\hat{\theta}_2^{(k)}$ is first extracted from the denoised magnitude and regularized phase produced in the previous iteration. Then, we excite the unobserved portion of the spectrum by injecting i.i.d. complex Gaussian noise having zero mean and standard deviation $\sigma_{\text{excite}}^{(k)}$ leaving θ_1 unaltered. The standard deviation $\sigma_{\text{excite}}^{(k)}$ typically has an exponential decay with re-

spect to the iteration number k and should converge to the standard deviation of the noise σ in the initial measurements.

3.3.2 Volumetric Filtering

The coefficients of the excited spectrum $\hat{\theta}^{(k)}_{\text{excite}}$ are then modified by independently denoising its magnitude and phase, thus obtaining $\hat{y}^{(k)}$ and $\hat{\phi}^{(k)}$, respectively. 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 is distributed accordingly to the Rician observation model (2.5) from the fact the noise in the corresponding excited spectrum is i.i.d. complex Gaussian. Conversely, for the sake of simplicity, the phase is assumed to obey the Gaussian observation model (2.2) with standard deviation equal to that of the excitation noise.

3.3.3 Data Reconstruction

The sequence of filtering estimates might get trapped in local optima because the data piloting the regularization, i.e. the available spectrum θ_1, is corrupted by noise. Thus, in order to escape from possible degenerate solutions, we aggregate all estimates in a complex recursive convex combination $\tilde{y}^{(k)}e^{\imath\tilde{\phi}^{(k)}}$ which fuses all $\hat{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ using weights inversely proportional to the variances of the corresponding excitation noise.

3.3.4 Discussion

Observe that the sequence of estimates produce by the denoising filter $\hat{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ is not convergent, but it approaches the sample mean of $\hat{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ over k, and 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. Thus, $\tilde{y}^{(k)}e^{\imath\tilde{\phi}^{(k)}}$

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 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 of the noisy phantom is completely available and the excitation noise has constant value of standard deviation for all k, the reconstruction algorithm coincides with a one-time application of the denoising filter, because the inputs of each iteration do not vary with k.

Thus, the proposed algorithm generalizes both the iterative reconstruction algorithm implemented in [44, 32] to the case of noisy observations, as well as the BM4D filter to the case of incomplete measurements.

3.3.5 Results

The 3-D sampling operator S 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 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. Fig. 6 (p. 128) of Publication IV illustrates different examples of k-space trajectories.

In Table IV (p. 128) and Fig. 9 (p. 130) of Publication IV, we present the objective and subjective reconstruction performance after 1000 iterations from a set of incomplete noisy or noise-free k-space measurements with either zero or non-zero phase ϕ illustrated in Fig. 5 (p. 127) of Publication IV. 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

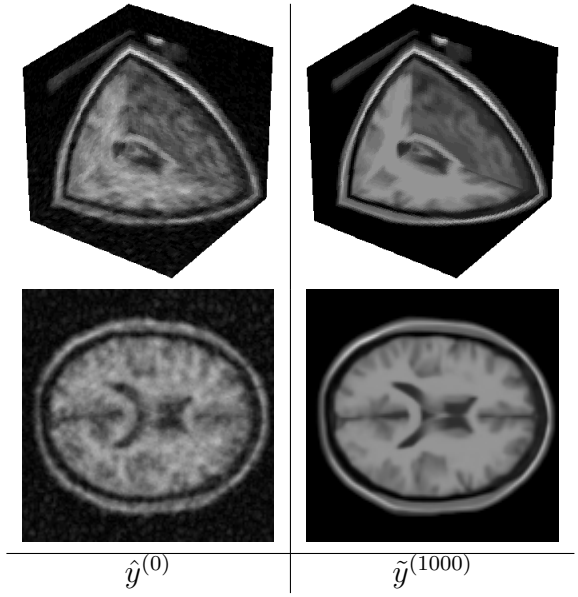


Figure 3.5. Initial back-projections (left) and final magnitude estimate (right) after 1000 iterations of the BrainWeb phantom reconstructed from noisy measurements ($\sigma = 5\%$) sensed with *Radial* trajectory and sampling ratio 30%.

erroneous jumps. In Fig. 3.5 we show the reconstructed BrainWeb magnitude after 1000 iterations in the case of initial noisy k-space measurements and non-zero phase sampled with *Radial* trajectory.

Fig. 8 (p. 129) of Publication IV gives a deeper insight on the PSNR progression with respect to the number of iterations. 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. The figure empirically shows that the ratio between the PSNR of $\tilde{y}^{(k)}$ and $\hat{y}^{(k)}$ approaches one, as motivated in Section 3.3.4.

Chapter 4

Video Filtering

In this chapter we introduce a powerful video filtering framework based on an overcomplete nonlocal representation of the video using motion-compensated 3-D spatiotemporal volumes as basic filtering elements. The filter is designed to sparsify these volumes in spatiotemporal transform domain leveraging the redundancy of the data in a fashion similar to the BM3D algorithm [25]. The spatiotemporal volumes are 3-D structure formed by a sequence of blocks following a specific motion trajectory obtained, for example, by concatenation of motion vectors along time [61]. Then, a nonlocal search procedure matches and subsequently stacks together mutually similar spatiotemporal volumes into 4-D groups. The group is transformed by a 4-D separable spatiotemporal transform leveraging the local spatial correlation between pixels in each block of a volume, local temporal correlation between blocks of each volume, as well as the nonlocal spatial and temporal correlation between volumes of the same group. The 4-D spectrum conveniently describes the characteristics of the grouped data, allowing to adapt the filtering by coefficient shrinkage with respect to the peculiar frequency information encoded within each 4-D spectrum coefficient. The richness of our spectral description is the fundamental building block in the development of the proposed video restoration framework.

The remainder of this chapter is organized as follows. At first in Section 4.1 we define the core elements of our spatiotemporal framework. Then, in Section 4.2 we describe the proposed V-BM4D filter and its implementation for the denoising, deblocking, sharpening, and deflickering of grayscale and color videos. Finally, in Section 4.3 we describe a second filter, termed RF3D, which targets the problem of denoising videos corrupted by spatially and temporally correlated noise.

4.1 Basic Algorithm

In this section we introduce the general spatiotemporal framework that stands as foundation of the proposed video filters. We denote a noisy video as $z : X \times T \to \mathbb{R}$

$$z(\mathbf{x}, t) = y(\mathbf{x}, t) + \eta(\mathbf{x}, t) \tag{4.1}$$

where y is the original (unknown) video, η is the noise, and $(\mathbf{x}, t)$ is a 3-D voxel coordinates belonging to the spatial domain $X \subset \mathbb{Z}^2$ and time domain $T \subset \mathbb{Z}$, respectively.

4.1.1 Spatiotemporal Volumes

The spatiotemporal volumes are built as a sequence of 2-D $N \times N$ blocks following a motion trajectory of the video, which is essentially a set of spatiotemporal coordinates defining blocks that supposedly contain the same moving feature of the video along time. Each block is extracted from a different frame and all frames spanned by the trajectory are consecutive in time. Thus, assuming that the trajectory for any given reference block $B(\mathbf{x}_R, t_R)$ is known, one can easily define the corresponding motion-compensated spatiotemporal volume as a 3-D structure $V(\mathbf{x}_R, t_R)$ composed by the (reference) blocks defining the motion trajectory.

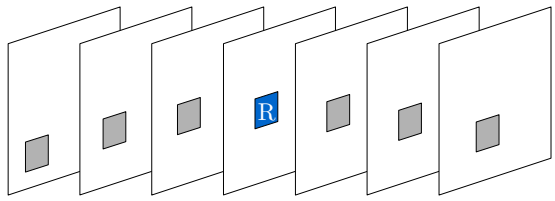


Figure 4.1. Schematic illustration of a spatiotemporal volume. The blocks of the volume are grey with the exception of the reference block "R", which is blue.

An accurate motion estimation strategy is of paramount interest to our framework, because the temporal correlation in the spatiotemporal volumes is vital to provide a sparse spectral description of the data which in turn translates to improved filtering results. The motion trajectories can be either known *a-priori*, or built in-loop, e.g., from the motion information produced by a coding module [61]. The motion estimation technique needs to be also tolerant to noise [7, 10, 71]. In Fig. 4.1, we show a schematic illustration of a spatiotemporal volume. In the figure, the reference block $B(\mathbf{x}_R, t_R)$ is shown in blue and occupies the middle position, the other blocks of the volume are shown in grey.

4.1.2 Grouping

Each (reference) spatiotemporal volume $V(\mathbf{x}_R, t_R)$ in the video is tested for similarity against all volumes within a nonlocal search neighborhood using a distance operator such as the ℓ_2-norm of the volumes difference. The group $G(\mathbf{x}_R, t_R)$ associated to the reference volume $V(\mathbf{x}_R, t_R)$ is a 4-D data structure composed by stacking together mutually similar 3-D volumes along an additional fourth dimension, and thus the groups constitute the nonlocal elements of the framework. Fig. 4.2 shows an example of spatiotemporal volume (left) and group (right).

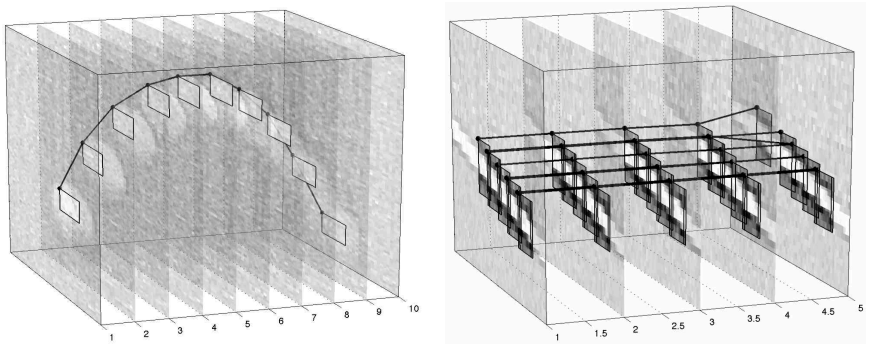


Figure 4.2. Illustration of a spatiotemporal volume (left), and a group of mutually similar volumes (right).

The spatiotemporal volumes have a fixed temporal extent H; however shorter volumes are also formed whenever an occlusion or a scene change arise in the video. We implicitly deal with such cases by requiring each block in the volume to provide at least a minimum similarity value with respect to the reference one. The trajectories are stopped either when the maximum temporal extent is reached or whenever it is not possible to find a block with the required minimum similarity. Consequently, different volumes might have different temporal extents. However, since the volumes within each group must have the same temporal extent, during the grouping we only consider volumes having extent greater than or equal to the one of the reference volume and then we extract from the longer volumes a subvolume having length equal to the one of the reference one. There are many ways to extract such subvolumes, and for the sake of simplicity we constrain that each volume in the group to be temporally synchronized with all the others.

4.1.3 Spatiotemporal Filtering

During the spatiotemporal filtering, similarly to (2.8), the group $G(\mathbf{x}_R, t_R)$ is first transformed via a decorrelating separable linear transform $\mathcal{T}_{4D}$, then a shrinkage operator Υ modifies the magnitude of the spectrum coefficients to attenuate the noise. This strategy leverages the sparse spectral description of the 4-D group induced by $\mathcal{T}_{4D}$. An estimate of the noise-free data $\hat{G}(\mathbf{x}_R, t_R)$ is eventually obtained after inverting the transform $\mathcal{T}_{4D}$ on the thresholded spectrum. The data is collaboratively filtered thus generating individual estimates of each noise-free 3-D volume (and 2-D block) in the group. The transform $\mathcal{T}_{4D}$ is a 4-D separable composition of a spatial $\mathcal{T}_{2D}$ transform, a $\mathcal{T}_{1D}$ transform in the temporal dimension, and an additional $\mathcal{T}_{1D}$ in the fourth (grouping) dimension.

In particular, the spatiotemporal volume is characterized by local spatial correlation within each block and temporal correlation along its third dimension. The 3-D spectrum of the spatiotemporal volume is obtained by applying a 2-D spatial transform to each patch in the volume followed by a 1-D temporal transform along the third (temporal) dimension. Thus, the temporal DC plane encodes the features shared among the blocks in the volume. The nonlocal correlation is localized along the grouping dimension of the group with respect to the ulterior 1-D linear transform. Consequently, the 4-D spectrum is structured according to the four dimensions of the corresponding group, i.e. two local spatial, one local temporal, and one for the nonlocal similarity. In particular, it includes a 2-D plane corresponding to the DC terms of the two 1-D transforms used for decorrelating the temporal and non-local dimensions of the group, and 3-D volume corresponding to the DC term of the 1-D temporal transform.

4.1.4 Aggregation

The groups constitute a very redundant representation of the video, consequently the overlapping estimates are aggregated through the usual convex combination as described in Section 2.3.3.

Figure 4.3. From top to bottom: noise-free frame, noisy frame, and V-BM4D denoising results for the grayscale and color sequence *Foreman* and *Tennis*. The sequences have intensity range of [0,255] and are corrupted by i.i.d. additive white Gaussian noise with standard deviation $\sigma = 40$.

4.2 Filtering in 4-D Transform Domain

We originally implemented the video denoising framework as a grayscale denoising filter V-BM4D and then, leveraging the spatiotemporal modeling, we have extended V-BM4D to allow for video deblocking, sharpening as well as multi-channel (color) filtering.

4.2.1 Denoising

The proposed V-BM4D, as described in the previous section, can be directly applied for the denoising of videos corrupted as in (2.2), i.e. specifying the noise in 4.1 as $\eta(\cdot,\cdot) \sim \mathcal{N}(0,\sigma^2)$. Multi-channel (e.g., RGB) videos corrupted by AWGN can be also filtered in luminance-chrominance color space. The motion estimation and grouping information of the luminance channel are reused within the chrominance channels to increase the efficiency of the filter [25].

We compare the V-BM4D grayscale and color denoising performances against those of the state-of-the-art V-BM3D algorithm [24]. Objective results in term of PSNR (2.9) and MOVIE index [114] demonstrate that V-BM4D outperforms V-BM3D with a substantial improvement in nearly every experiment as reported by Table II (p. 109) in Publication III. Subjective visual results for grayscale and color denoising are shown in Fig. 4.3, Fig. 6 (p. 111) and Fig. 10 (p. 112) in Publication III, and Fig. 6 (p. 89) in Publication I substantiate the excellent numeric performances: as a subjective quality assessment, V-BM4D better preserves textures, without introducing disturbing artifacts in the restored video, even under the presence of high level of noise.

4.2.2 Deblocking

Most video compression algorithms, such as MPEG-4 [116] or H.264 [128], make use of block-transform coding and thus may suffer, especially at low bitrates, from several compression artifacts due to the motion compensation and the coarse quantization of the block-transform coefficients. Inspired by [49] we treat the blocking artifacts as additive noise. This choice allows us to model the compressed video z as AWGN data (2.2), with y now corresponding to the uncompressed video, and η being noise with variance σ^2 describing the compression artifacts. In practice, we relate σ to the actual compression artifacts parametrized by the average bit-per-pixel (bpp) rate of the compressed video and the quantization parameter q [116]. Ob-

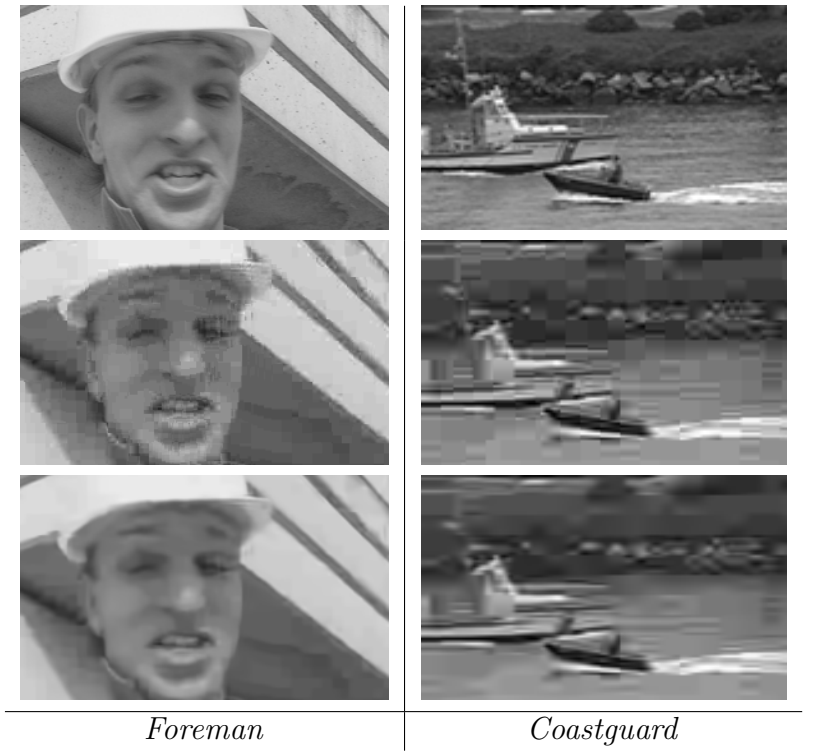


Figure 4.4. From top to bottom: original frame, compressed frame, and deblocking results of V-BM4D for the sequence *Foreman* and *Coastguard* compressed using the MPEG-4 encoder with quantization parameter $q = 25$.

serve that such value of σ is not an estimate of the noise variance in the compressed videos, but it is the assumed value of the variance of an hypothetical Gaussian noise η which would be filtered using the same level σ necessary to remove the blocking artifacts.

We compare the V-BM4D deblocking filter against the *MPlayer accurate deblocking filter*, as, to the best of our knowledge, this is one of the most effective deblocking algorithm. Numerical results reported in Table III (p. 110) of Publication III show that V-BM4D

significantly outperforms *Mplayer* in all the experiments in terms of PSNR, whereas interestingly the MOVIE index [114] often prefers the non-filtered blocky sequences over the deblocked counterparts thus showing a general preference towards piecewise smooth images, a behavior that contradicts its purpose of acting in agreement with human visual judgement. Fig. 4.4 and Fig. 7 (p. 111) of Publication III presents the V-BM4D deblocking visual results, showing a significant improvement of the image quality.

4.2.3 Enhancement

Enhancement techniques are used to improve the image quality by sharpening the image features characterized by low contrast. A critical issue of enhancement algorithms is the amplification of the noise together with the sharpening of image details [93, 1], an effect that becomes more severe as the amount of applied sharpening increases. In order to overcome this problem, a joint application of a denoising and sharpening filter is often recommendable, and in particular this practice has been investigated in [26].

Among the existing enhancement techniques, we choose the so-called alpha-rooting [1], which induces sharpening by scaling the large coefficients relatively to the small ones by raising the magnitude of each spectrum coefficient to a power $1/\alpha$, with $\alpha > 1$. We combine the V-BM4D denoising with the alpha-rooting operator, in order to simultaneously reduce the noise and sharpen the original signal [26, 82]. The V-BM4D sharpening filter includes the hard-thresholding stage only, and the alpha-rooting is operated on the spectrum coefficients right after the thresholding. Fig. 4.5 and Fig. 8 (p. 111) of Publication III show the visual performances of the joint application of denoising and sharpening, demonstrating a good detail enhancement together with an excellent noise suppression using different values of the parameter α.

The 4-D spectral representation induced by the 3-D spatiotemporal volumes can be exploited to selectively process different portions

Figure 4.5. Top row: noise-free (left) and noisy (right) frame of the sequence *Bus* having intensity range of [0,255] corrupted by i.i.d. additive white Gaussian noise with standard deviation $\sigma = 25$. Bottom row: sharpening results of V-BM4D using $\alpha = 1.1$ (left) and $\alpha = 1.25$ (right).

of the 4-D spectrum. Hence, the value of α can be decreased for the coefficients that belong to the temporal AC coefficients, in order to attenuate the temporal flickering artifacts. In Fig. 9 (p. 111) of Publication III, we show the enhancement results of V-BM4D applied to the test sequence *Miss America*. We use either a unique value $\alpha_{\mathrm{DC}} = \alpha_{\mathrm{AC}} = 1.25$ for the whole 4-D spectrum, or different values $\alpha_{\mathrm{DC}} = 1.25$ and $\alpha_{\mathrm{AC}} = 0.625$ to apply a different level of sharpening of the temporal DC and AC coefficients. One can clearly notice that the sequence processed using $\alpha_{\mathrm{DC}} \neq \alpha_{\mathrm{AC}}$ is dramatically less affected by flickering artifacts because the intensities of the background in the temporal difference are extremely smooth. Thus, a non-uniform sharpening of the 4-D spectrum allows V-BM4D to significantly attenuate the flickering, yet maintaining excellent enhancement (sharpening) and noise reduction properties.

4.2.4 Discussion

The improved effectiveness of V-BM4D indicates the importance of separately treating spatial and temporal correlation, and, in particular, of explicitly accounting the motion information. Let us analyze the PSNR performances of the algorithms when a temporal-based or nonlocal-based grouping is encouraged. In Fig. 11 (p. 115) of Publication III we present the PSNR of the results provided by V-BM4D using different combinations of grouping parameters. The analysis empirically demonstrates that the nonlocal spatial correlation does not dramatically affect the global performances, and highlights the dramatic improvement as the size of the spatiotemporal volume, and thus the amount of temporally correlated data, increases. Thus we pinpoint the importance of the spatiotemporal volumes and temporal correlation in conjunction with our filtering framework as a basic filtering elements during video restoration.

4.3 Random and Fixed-Pattern Noise Removal

In this section, we present a denoising algorithm for videos jointly corrupted by spatially correlated (i.e. non-white) random noise and spatially correlated fixed-pattern noise. Thus, for the case considered, the noise in the generic observation model (4.1) becomes

$$\eta\left(\mathbf{x},t\right)=\eta_{\mathrm{RND}}\left(\mathbf{x},t\right)+\eta_{\mathrm{FPN}}\left(\mathbf{x},t\right), \tag{4.2}$$

where η_{RND} and η_{FPN} are colored Gaussian random and fixed-pattern noise having individual non-uniform PSDs σ^2_{RND} and σ^2_{FPN} defined with respect to a 2-D spatial transform $\mathcal{T}_{\mathrm{2D}}$. The PSDs σ^2_{RND} and σ^2_{FPN} can be separated into their normalized time-invariant counterparts Ψ_{RND}, Ψ_{FPN}, which are are assumed to be known and fixed, and their corresponding unknown time-variant scaling factors $\varsigma^2_{\mathrm{RND}}$ and $\varsigma^2_{\mathrm{FPN}}$. This model can be practically used to describe the raw output of microbolometers LWIR cameras.

Our approach, denoted as RF3D, is essentially based on the spatiotemporal filtering framework described in Section 4.1 but, based on the analysis in Section 4.2.4, we disable the nonlocal feature, i.e. the grouping, and we only use the volumes as basic elements for the filtering; as a result the 4-D transform used in V-BM4D, reduces to a 3-D transform in RF3D. However, let us note that RF3D can be easily adapted to the general case of 4-D filtering by first grouping mutually similar spatiotemporal volumes and then decorrelating them through an additional transform along the fourth dimension. In order to address the spatial and temporal correlation of the noise in (4.2), the coefficient shrinkage in RF3D relies on a 3-D array of variances to be used as threshold parameters during the collaborative filtering. The idea of applying different shrinkage strategies within different hyperplanes can be traced back already in [82] where a similar to that described in Section 4.2.3 is leveraged to improve the contrast and reduce the noise in videomicroscopy sequences. RF3D is implemented in two cascading stages, namely the hard-thresholding and the Winer-filtering stage.

4.3.1 Noise Estimation

Assuming that the fixed-pattern noise (FPN) is roughly constant in time, a spatial high-pass filtering of the video captures both random and fixed-pattern noise components, whereas a temporal high-pass filter captures only the random one. Thus, we can estimate the PSDs by applying the median absolute deviation (MAD) [57, 41] of all $\mathcal{T}_{2D}$ high-frequency block coefficients of every frame within a specified temporal window. Once an estimate of the global (i.e. random and fixed-patter noise) PSD and the PSD of the random noise are obtained, we can calculate the scaling factors ς^2_{RND} and ς^2_{FPN} of the two noise components using the estimated PSDs and the known Ψ_{RND}, Ψ_{FPN} through a non-negative least-squares optimization.

4.3.2 Motion-Adaptive 3-D Spectrum Variances

The shrinkage operator Υ in the collaborative filtering modulates the applied filtering strength relying on the variances of the $\mathcal{T}_{3D}$-spectrum coefficients. However, due to the presence of the FPN, the relative spatial alignment of the blocks in the filtered volume has an impact on the variance of the spectrum coefficients and thus needs to be taken into account for the design of the threshold coefficients. Thus, we use a 3-D threshold array of variances defined accordingly to the characteristics of the spatiotemporal volume.

If all blocks are perfectly overlapping, the FPN component, being the same across all blocks, accumulates through averaging in the 2-D temporal DC plane of the 3-D volume spectrum. Thus the variances of the temporal DC plane contain the contributions of the random noise and the accumulated FPN, whereas the AC coefficients contain only the random noise. Conversely, if all blocks have different spatial positions and their relative displacement is such that the FPN exhibits uncorrelated patterns over different blocks, then, restricted to the volume, the FPN behaves just like another random component and the variances of the coefficients can be simply obtained as the sum of the two noise components. All the intermediate cases for which any number of blocks in the volume are aligned or partially aligned with any of the others are approximated with an interpolation formula.

4.3.3 Enhanced Fixed-Pattern Suppression

We also propose an enhancement of RF3D, denoted E-RF3D, in which the fixed-pattern (FP), i.e. the actual realization of the FPN, is first progressively estimated using the data previously filtered, and then subtracted from the subsequent noisy frames. The flowchart of both RF3D and E-RF3D is shown in Fig. 4 (p. 137) of Publication V.

According to the additive model (4.1) and noise (4.2), also assuming that $\hat{y}$ is a perfect estimate of y and that the FPN is time-invariant

within any short temporal extent, the FP can be simply estimated by averaging the noise residuals obtained from a set of consecutive filtered frames. However, the FP estimate is still corrupted by a new random noise component and, thus, a new estimation of the standard deviation and the PSD of the updated FPN component becomes necessary. We model the PSD of the updated FPN as a convex combination of the original PSDs Ψ_{RND} and Ψ_{FPN}, then we estimate the scaling factors of the mixed PSDs as the solutions of a non-negative least-squares problem similar to that used in the case without FP subtraction. Finally, the estimated scaling factors are used to compute the parameter used in the convex combination determining the contributions of the original PSDs.

4.3.4 Results

In our experiments, both videos corrupted by synthetic noise (4.2) and real LWIR thermography sequences acquired using a FLIR Tau 320 camera are considered. In Table I and Table II (p. 142) in Publication V, we report the objective experimental evaluation of the proposed RF3D and E-RF3D compared with the same algorithm using different *a-priori* assumptions on the corrupting noise, as well as against state-of-the-art denoising algorithms BM4D presented in Chapter 3 and V-BM3D [24]. Numeric performances demonstrate that RF3D and E-RF3D consistently outperform the results obtained by the compared methods with a substantial PSNR improvement in nearly every experiment.

In Fig. 4.6 and Fig. 11 (p. 144) of Publication V, we show the visual denoising results obtained by the proposed method applied on data corrupted by synthetic noise having $\varsigma_{\text{RND}} = \varsigma_{\text{FPN}} = 15$ over an intensity range of [0,255]. As one can see, RF3D and E-RF3D generate more visually pleasant images, as the artifacts of the FPN are dramatically reduced and many high-frequency features are nicely preserved. We also use the proposed method for the denoising of two real LWIR thermography sequences acquired using a FLIR Tau 320

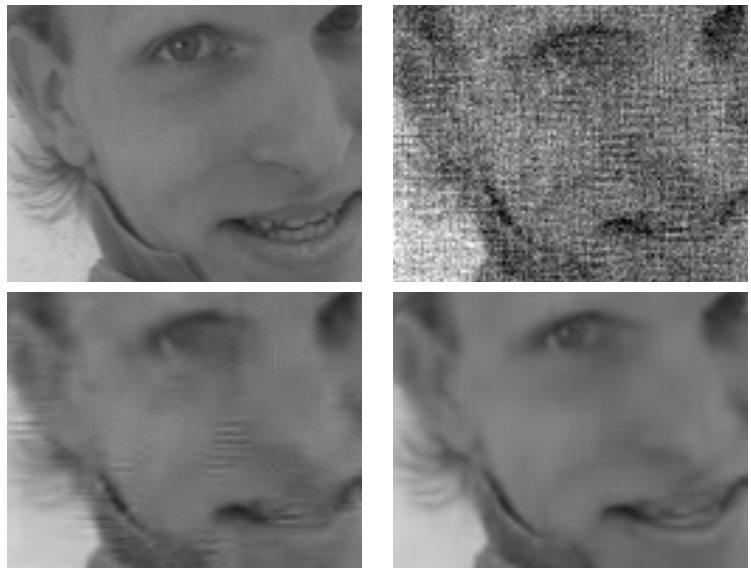


Figure 4.6. Top row: noise-free (left) and noisy (right) frame of *Foreman* corrupted by random and fixed-pattern noise having $\varsigma_{\mathrm{RND}} = \varsigma_{\mathrm{FPN}} = 15$ over an intensity range of [0,255]. Bottom row: denoised results of RF3D (left) and E-RF3D (right).

camera. The noise in the acquired data corresponds to $\varsigma_{\mathrm{RND}} \approx 6.5$ and $\varsigma_{\mathrm{FPN}} \approx 4.3$ for a $[0, 255]$ range. The visual denoising results of real LWIR data, shown in Fig. 14 (p. 146) of Publication V, confirm the previous analysis.

Chapter 5

Conclusions

5.1 Summary of the thesis

We introduced two restoration frameworks for high-dimensional data. The main contribution of the thesis consists in establishing a design for the patches used as basic filtering elements, and in leveraging such design during the filtering. In the context of volumetric data and video filtering, we characterize the basic elements to be 3-D cubes and 3-D spatiotemporal volumes respectively. Specifically, the cubes are a natural extension of the concept of block in 3-D domain, whereas the volumes are defined as a sequences of blocks following a motion trajectory in the video. In doing so, the local spatial correlations within the cubes or the local spatial and temporal correlation within the volumes, combined with the nonlocal correlation provided by the grouping of mutually similar elements, allows a decorrelating transformation to disclose a highly descriptive spectral representation of the filtering structure at hand.

In this thesis we demonstrate that the spectral representation of groups based on cubes or volumes can be used to selectively manipulate the transform coefficients along the different spectral dimensions. In particular, the shrinkage strategies can be adapted with respect to the spectral hyperplane as well as to the specific coefficients

Table 5.1. Summary of the contributions of this thesis in term of foundational aspects and developed algorithms.

Foundational Aspects
High-dimensional filtering model based on 4-D groups of mutually similar 3-D cubes and 3-D spatiotemporal volumes.
Adaptive shrinkage of the 4-D groups in transform domain using the information coded within specific spectral hyperplanes.
Filtering model allowing for noise characterized by spatial and temporal correlation.

Developed Algorithms
Denoising of volumetric data with adaptive noise estimation.
Reconstruction of volumetric data from incomplete and noisy transform-domain measurements.
Denoising, deblocking, deflickering, and enhancement for grayscale and color videos.

within each hyperplane: this allows to consider heterogenous observation models for the corrupted data featuring noise characterized by signal-dependent distributions, spatial or temporal correlation, non-white power spectral densities, or spatially varying statistics. Thus, our modeling is leveraged to develop a wide range of algorithms – summarized in Table 5.1– which are among the state of the art for several fundamental image processing problems.

The developed algorithms have already found practical use in several imaging fields. In particular, the volumetric filter is naturally employed in medical [65, 132] and hyperspectral imaging [109] as a

powerful filtering tool to improve the data for subsequent processing tasks. The video filter has been successfully applied in thermal imaging [50] where the spatial correlation is a primal characteristic of the corrupting noise, as well as in biomedical imaging to facilitate the tracking of features in videomicroscopy sequences corrupted by heavy noise and flickering [82].

5.2 Future Research Directions

As covered in this thesis, assessing image redundancy and nonlocal self-similarity is of fundamental importance for several image processing applications, however traditional metrics are not always consistent with the human visual system as images perceived as identical by a human observer can have a large point-by-point difference; this is especially true in the case of textured content. Textures are ubiquitous in natural signals, and can be loosely defined as an almost identical repetition of elementary components within an almost regular pattern at approximately the same scale. The preliminary study on the relation of texture and noise in [80] shows the potential of using statistical features to assess patch similarity in the presence of noise, hence a future research direction focuses on embedding elements of human perception theory into restoration algorithms. Recent studies tackle this problem through the nonlocal paradigm by exploiting external databases, i.e. sets of noise-free natural patches, to denoise structured, e.g., textured, patches characterized by strong signal features [96]. Differently, in [131] the authors model natural stochastic textures as a Gaussian self-similar process to form a prior for different imaging application such as super-resolution and denoising.

The mismatch between traditional patch-matching strategies and the human visual system can be also addressed by embedding foveation principles during the filtering. Foveated imaging is an image processing technique that takes into account the inability of the human visual system to perceive high resolution visual signals outside the fovea, i.e.

the fixation point of the eye. Recently, foveation has been studied in the context of image denoising as a tool to define a similarity metric which computes the distance between foveated patches [48]; this is implemented by using point spread functions of spread that increases accordingly to the spatial distance from the center of the patch. The foveated distances used in place of the common point-by-point distance in NLM [14] has lead to remarkable improvements in the final image estimate. Thus, another research direction focuses on embedding foveated imaging within the collaborative filtering frameworks, further adapting it to high-dimensional patches.

Furthermore, the proposed frameworks can be extended to the direct filtering of complex-valued data which is not natively handled in our current approaches. Our methods extract magnitude and phase from the complex data, and independently process each component; this strategy might be improved by embedding complex-domain operations, such as the Fourier transform, throughout the filtering. Along a similar line of research, in the context of processing signals with non-zero phase, the work [8] has shown the potential of prefiltering the data before performing phase unwrapping, which is an operation of paramount interest for several imaging applications such as for example tomography, spectroscopy, interferometry, and MRI. The problem is to reconstruct the absolute phase from the wrapped measurements caused by 2π discontinuities in the acquired data at the extreme values $-\pi$ and π. Thus, the proposed volumetric filtering framework can be also leveraged for the prefiltering and the reconstruction of data characterized by wrapped phase. Additionally, the same filter can be adapted for higher-dimensional applications, such as videos of 3-D volumetric data, diffusional MRI, as well as multispectral imaging.

On a more practical side, given the size of the data processed by the proposed method and the high computational demand of few operations therein (e.g., block matching), efficient implementations exploiting many-core architectures (such as GPUs) are of paramount interest for both desktop and mobile applications.

Bibliography

[1] S. Aghagolzadeh and O.K. Ersoy. Transform image enhancement. *Optical Engineering*, 31(3):614–626, 1992.

[2] J.P. Antoine, P. Vandergheynst, and R. Murenzi. Two-dimensional directional wavelets in image processing. *International Journal of Imaging Systems and Technology*, 7(3):152–165, 1996.

[3] J. Ashburner and K.J. Friston. Voxel-based morphometry – the methods. *NeuroImage*, 11(6):805–821, 2000.

[4] J. Astola, V. Katkovnik, and K. Egiazarian. *Local Approximation Techniques in Signal and Image Processing*. SPIE Press, 2006.

[5] S.P. Awate and R.T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):364–376, Mar. 2006.

[6] S.K. Bar-Lev and P. Enis. On the construction of classes of variance stabilizing transformations. *Statistics & Probability Letters*, 10(2):95–100, 1990.

[7] M. Bertero, T.A. Poggio, and V. Torre. Ill-posed problems in early vision. *Proceedings of the IEEE*, 76(8):869–889, Aug. 1988.

[8] J. Bioucas-Dias, V. Katkovnik, J. Astola, and K. Egiazarian. Adaptive local phase approximations and global unwrapping. In *Proceedings of the 3DTV Conference*, pages 253–256, May 2008.

[9] G. Boracchi and A. Foi. Multiframe raw-data denoising based on block-matching and 3-D filtering for low-light imaging and stabilization. In *Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing*, 2008.

[10] J. Boulanger, C. Kervrann, and P. Bouthemy. Space-time adaptation for patch-based image sequence restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1096–1102, Jun. 2007.

[11] W.S. Boyle and G.E. Smith. Charge coupled semiconductor devices. *Bell System Technical Journal*, 49(4):587–593, 1970.

[12] J.C. Brailean, R.P. Kleihorst, S. Efstratiadis, A.K. Katsaggelos, and R.L. Lagendijk. Noise reduction filters for dynamic image sequences: a review. *Proceedings of the IEEE*, 83(9):1272–1292, Sep. 1995.

[13] A. Buades, B. Coll, and J.M. Morel. Denoising image sequences does not require motion estimation. In *Proceedings of the Conference on Advanced Video and Signal Based Surveillance*, pages 70–74, Sep. 2005.

[14] A. Buades, B. Coll, and J.M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490–530, 2005.

[15] J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[16] E. J. Candès and D.L. Donoho. Curvelets: a surprisingly effective nonadaptive representation of objects with edges. In *Proceedings of the International Conference of Curves and Surface Fitting*, pages 0–82651357. University Press, 2000.

[17] E.J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, Feb. 2006.

[18] D.M. Chandler, M.M. Alam, and T.D. Phan. Seven challenges for image quality research. In *Proceedings of the SPIE Human Vision and Electronic Imaging*, volume 9014, pages 901402–901402–14, 2014.

[19] P. Chatterjee and P. Milanfar. Is denoising dead? *IEEE Transactions on Image Processing*, 19(4):895–911, Apr. 2010.

[20] W.S. Cleveland and S.J. Devlin. Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.

[21] R.R. Coifman and D.L. Donoho. Translation-invariant denoising. In *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 125–150. Springer New York, 1995.

[22] P. Coupé, J.V. Manjón, M. Robles, and D.L. Collins. Adaptive multiresolution non-local means filter for three-dimensional magnetic resonance image denoising. *IET Image Processing*, 6(5):558–568, Jul. 2012.

[23] 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*, 27(4):425–441, Apr. 2008.

[24] K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3D transform-domain collaborative filtering. In *Proceedings of the European Signal Processing Conference*, Sep. 2007.

[25] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, Aug. 2007.

[26] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Joint image sharpening and denoising by 3D transform-domain collaborative filtering. In *Proceedings of the International Workshop on Spectral Methods and Multirate Signal Processing*, 2007.

[27] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image restoration by sparse 3D transform-domain collaborative filtering. In *Proceedings of the SPIE Electronic Imaging*, volume 6812-07, Jan. 2008.

[28] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. BM3D image denoising with shape-adaptive principal component analysis. In *Proceedings of the International Workshop on Signal Processing With Adaptive Sparse Structured Representations*, 2009.

[29] R. Damadian. Tumor detection by nuclear magnetic resonance. *Science*, 171(3976):1151–1153, 1971.

[30] A. Danielyan and A. Foi. Noise variance estimation in nonlocal transform domain. In *Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing*, pages 41–45, Aug. 2009.

[31] 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.

[32] A. Danielyan, V. Katkovnik, and K. Egiazarian. Image deblurring by augmented Lagrangian with BM3D frame prior. In *Proceedings of the Workshop on Information Theoretic Methods in Science and Engineering*, Aug. 2010.

[33] A. Danielyan, M. Vehvilainen, A. Foi, V. Katkovnik, and K. Egiazarian. Cross-color bm3d filtering of noisy raw data. In *Proceedings of the International Workshop on Local and Non-Local Approximation in Image Processing*, pages 125–129, Aug. 2009.

[34] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

[35] J.S. De Bonet. Noise reduction through detection of signal redundancy. Technical report, Rethinking Artificial Intelligence, MIT AI Lab, 1997.

[36] C.A. Deledalle, V. Duval, and J. Salmon. Non-local methods with shape-adaptive patches (NLM-SAP). *Journal of Mathematical Imaging and Vision*, 43(2):103–120, 2012.

[37] A.J. den Dekker and J. Sijbers. Data distributions in magnetic resonance images: A review. *Physica Medica*, 2014.

[38] D.L. Donoho. De-noising by soft thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995.

[39] D.L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, Apr. 2006.

[40] D.L. Donoho, I. Johnstone, and I.M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1993.

[41] D.L. Donoho and I.M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association*, 90(432):1200–1224, Dec. 1995.

[42] E. Dubois and S. Sabri. Noise reduction in image sequences using motion-compensated temporal filtering. *IEEE Transactions on Communications*, 32(7):826–831, Jul. 1984.

[43] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1033–1038, 1999.

[44] K. Egiazarian, A. Foi, and V. Katkovnik. Compressed sensing image reconstruction via recursive spatially adaptive filtering. In *Proceedings of the International Conference on Image Processing*, volume 1, pages 549–552, Oct. 2007.

[45] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, Dec. 2006.

[46] A. Foi. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 89(12):2609–2629, Dec. 2009.

[47] A. Foi. Noise estimation and removal in MR imaging: the variance-stabilization approach. In *Proceedings of the International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1809–1814, Chicago, IL, USA, Mar. 2011.

[48] A. Foi and G. Boracchi. Foveated self-similarity in nonlocal image filtering. In *Proceedings of the SPIE Human Vision and Electronic Imaging*, volume 8291, pages 829110–829110–12, 2012.

[49] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of

grayscale and color images. *IEEE Transactions on Image Processing*, 16(5):1395–1411, May 2007.

[50] A. Foi and M. Maggioni. Methods and systems for suppressing noise in images, Patent Application US 13/943,035, Filed Jul. 2013.

[51] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, Oct. 2008.

[52] B. Fowler, A. El Gamal, D. Yang, and H. Tian. A method for estimating quantum efficiency for CMOS image sensors. In *Proceedings of SPIE*, 1998.

[53] G. Gerig, O. Kubler, R. Kikinis, and F.A Jolesz. Nonlinear anisotropic filtering of MRI data. *IEEE Transactions on Medical Imaging*, 11(2):221–232, Jun. 1992.

[54] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, 3 edition, 2007.

[55] H. Gudbjartsson and S. Patz. The Rician distribution of noisy MRI data. *Magnetic resonance in medicine*, 34(6):910–914, Dec. 1995.

[56] O.G. Guleryuz. Weighted averaging for denoising with overcomplete dictionaries. *IEEE Transactions on Image Processing*, 16(12):3020–3034, Dec. 2007.

[57] F.R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, Jun. 1974.

[58] H. Hang, Y. Chou, and S. Cheng. Motion estimation for video coding standards. *Journal of VLSI Signal Processing Systems*, 17(2/3):113–136, 1997.

[59] H. Ji, S. Huang, Z. Shen, and Y. Xu. Robust video restoration by joint sparse and low rank matrix approximation. *SIAM Journal on Imaging Sciences*, 4(4):1122–1142, Nov. 2011.

[60] J.B. Johnson. Thermal agitation of electricity in conductors. *Physical Review*, 32:97–109, Jul. 1928.

[61] L. Jovanov, A. Pižurica, S. Schulte, P. Schelkens, A. Munteanu, E. Kerre, and W. Philips. Combined wavelet-domain and motion-compensated video denoising based on video codec motion estimation methods. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(3):417–421, Mar. 2009.

[62] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola. From local kernel to nonlocal multiple-model image denoising. *International Journal of Computer Vision*, 86:1–32, Jan. 2010.

[63] C. Kervrann and J. Boulanger. Optimal spatial adaptation for patch-based image denoising. *IEEE Transactions on Image Processing*, 15(10):2866–2878, Oct. 2006.

[64] C. Kervrann and J. Boulanger. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.

[65] J.H. Kim, I.J. Ahn, W.H. Nam, Y. Chang, and J.B. Ra. Post-filtering of PET image based on noise characteristic and spatial sensitivity distribution. In *Proceedings of the Nuclear Science Symposium and Medical Imaging Conference*, pages 1–3, Oct. 2013.

[66] R.P. Kleihorst, R.L. Lagendijk, and J. Biemond. Noise reduction of image sequences using motion compensation and signal decomposition. *IEEE Transactions on Image Processing*, 4(3):274–284, 1995.

[67] P. Koczyk, P. Wiewiór, and C. Radzewicz. Photon counting statistics – undergraduate experiment. *American Journal of Physics*, 64(3):240–245, 1996.

[68] K. Krissian and S. Aja-Fernández. Noise-driven anisotropic diffusion filtering of MRI. *IEEE Transactions on Image Processing*, 18(10):2265–2274, Oct. 2009.

[69] M. Lebrun, M. Colom, A. Buades, and J.M. Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, May 2012.

[70] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2833–2840, Jun. 2011.

[71] C. Liu and W.T. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *Proceedings of the European conference on Computer vision*, pages 706–719, 2010.

[72] C. Liu, R. Szeliski, S.B. Kang, C.L. Zitnick, and W.T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):299–314, Feb. 2008.

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

[74] M. Lustig, D.L. Donoho, J.M. Santos, and J.M. Pauly. Compressed sensing MRI. *IEEE Signal Processing Magazine*, 25(2):72–82, Mar. 2008.

[75] M. Lustig and J.M. Pauly. SPIRiT: Iterative self-consistent parallel imaging reconstruction from arbitrary k-space. *Magnetic Resonance in Medicine*, 64(2):457–471, 2010.

[76] A. Macovski. Noise in MRI. *Magnetic Resonance in Medicine*, 36(3):494–497, 1996.

[77] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising using separable 4D nonlocal spatiotemporal transforms. In *Proceedings of the SPIE Electronic Imaging*, volume 7870, Jan. 2011.

[78] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Transactions on Image Processing*, 21(9):3952–3966, Sep. 2012.

[79] M. Maggioni and A. Foi. Nonlocal transform-domain denoising of volumetric data with groupwise adaptive variance estimation. In *Proceedings of the SPIE Electronic Imaging*, volume 8296, Jan. 2012.

[80] M. Maggioni, G. Jin, A. Foi, and T.N. Pappas. Structural texture similarity metric based on intra-class variances. In *Proceedings on the International Conference on Image Processing*, Oct. 2014.

[81] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133, Jan. 2013.

[82] M. Maggioni, R. Mysore, E. Coffey, and A. Foi. Four-dimensional collaborative denoising and enhancement of time-lapse imaging of mCherry-EB3 in hippocampal neuron growth cones. In *Proceedings of the BioPhotonics and Imaging Conference*, Oct. 2010.

[83] M. Maggioni, E. Sánchez-Monge, and A Foi. Joint removal of random and fixed-pattern noise through spatiotemporal video

filtering. *IEEE Transactions on Image Processing*, 23(10):4282–4296, Oct. 2014.

[84] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241, 2008.

[85] M. Mäkitalo and A. Foi. A closed-form approximation of the exact unbiased inverse of the Anscombe variance-stabilizing transformation. *IEEE Transactions on Image Processing*, 20(9):2697–2698, Sep. 2011.

[86] M. Mäkitalo and A. Foi. Optimal inversion of the Anscombe transformation in low-count Poisson image denoising. *IEEE Transactions on Image Processing*, 20(1):99–109, Jan. 2011.

[87] M. Mäkitalo and A. Foi. Poisson-Gaussian denoising using the exact unbiased inverse of the generalized Anscombe transformation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 1081–1084, Mar. 2012.

[88] M. Mäkitalo and A. Foi. Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise. *IEEE Transactions on Image Processing*, 22(1):91–103, Jan. 2013.

[89] S.G. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, 3 edition, December 2008.

[90] 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*, 16(1):18–27, 2012.

[91] 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*, 31:192–203, 2010.

[92] 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*, 22(12):2677–2684, 2010.

[93] J.H. McClellan. Artifacts in alpha-rooting of images. In *Proceedings on the International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 449–452, Apr. 1980.

[94] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, Jan. 2013.

[95] A.F. Milton, F.R. Barone, and M.R. Kruer. Influence of nonuniformity on infrared focal plane array performance. *Optical Engineering*, 24(5):245855–245855, Aug. 1985.

[96] I Mosseri, M. Zontak, and M. Irani. Combining the power of internal and external denoising. In *Proceedings of the International Conference on on Computational Photography*, pages 1–9, Apr. 2013.

[97] É.A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.

[98] J. Nakamura. *Image Sensors and Signal Processing for Digital Still Cameras*. CRC Press, 2005.

[99] P.J.W. Noble. Self-scanned silicon image detector arrays. *IEEE Transactions on Electron Devices*, 15(4):202–209, Apr. 1968.

[100] H. Nyquist. Thermal agitation of electric charge in conductors. *Physical Review*, 32:110–113, Jul 1928.

[101] H. Oktem, V. Katkovnik, K. Egiazarin, and J. Astola. Local adaptive transform based image denoising with varying window

size. In *Proceedings on the International Conference on Image Processing*, volume 1, pages 273–276, 2001.

[102] H. Olkkonen. *Discrete Wavelet Transforms - Algorithms and Applications*. InTech, 2011.

[103] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill College, 3 edition, 1991.

[104] A Pižurica, W. Philips, I Lemahieu, and M. Acheroy. A versatile wavelet domain noise filtration technique for medical imaging. *IEEE Transactions on Medical Imaging*, 22(3):323–331, Mar. 2003.

[105] A. Pižurica, V. Zlokolica, and W. Philips. Combined wavelet domain and temporal video denoising. In *Proceedings of the Conference on Advanced Video and Signal Based Surveillance*, pages 334–341, Jul. 2003.

[106] J. Portilla. Full blind denoising through noise covariance estimation using Gaussian scale mixtures in the wavelet domain. In *Proceedings of the International Conference on Image Processing*, volume 2, pages 1217–1220, Oct. 2004.

[107] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338–1351, Nov. 2003.

[108] R. Ramanath, W.E. Snyder, Y. Yoo, and M.S. Drew. Color image processing pipeline. *IEEE Signal Processing Magazine*, 22(1):34–43, Jan. 2005.

[109] B. Rasti, J.R. Sveinsson, M.O. Ulfarsson, and J.A Benediktsson. Hyperspectral image denoising using first order spectral roughness penalty in wavelet domain. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2458–2467, Jun. 2014.

[110] P.L. Richards. Bolometers for infrared and millimeter waves. *Journal of Applied Physics*, 76(1):1–24, 1994.

[111] D.L. Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517–548, Nov. 1994.

[112] D. Rusanovskyy and K. Egiazarian. Video denoising algorithm in sliding 3D DCT domain. In *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 618–625, 2005.

[113] J. Salmon. On two parameters for denoising with non-local means. *IEEE Signal Processing Letters*, 17(3):269–272, Mar. 2010.

[114] K. Seshadrinathan and A.C. Bovik. Motion tuned spatio-temporal quality assessment of natural videos. *IEEE Transactions on Image Processing*, 19(2):335–350, Feb. 2010.

[115] J. Sijbers, A.J. den Dekker, P. Scheunders, and D. Van Dyck. Maximum-likelihood estimation of Rician distribution parameters. *IEEE Transactions on Medical Imaging*, 17(3):357–361, Jun. 1998.

[116] T. Sikora. The MPEG-4 video standard verification model. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(1):19–31, Feb. 1997.

[117] E.P. Simoncelli and B. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, May 2001.

[118] J.L. Starck, E.J. Candès, and D.L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):670–684, Jun. 2002.

[119] P. Stoica and R. Moses. *Spectral Analysis of Signals*. Prentice Hall, 1 edition, 2005.

[120] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the International Conference on Computer Vision*, pages 839–846, Jan. 1998.

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

[122] M.J. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transaction on Information Theory*, 55(5):2183–2202, May 2009.

[123] M.J. Wainwright and E.P. Simoncelli. Scale mixtures of Gaussians and the statistics of natural images. In *Proceedings of Advances in Neural Information Processing Systems*, volume 12, pages 855–861, Cambridge, MA, May 2000. MIT Press.

[124] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, Apr. 2004.

[125] Z. Wang and Q. Li. Statistics of natural image sequences: temporal motion smoothness by local phase correlations. In *Proceedings of the SPIE Human Vision and Electronic Imaging*, volume 7240, pages 1–12, Jan. 2009.

[126] G.S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 26(4):359–372, Dec. 1964.

[127] M.B. Weissman. $\frac{1}{f}$ noise and other slow, nonexponential kinetics in condensed matter. *Reviews of Modern Physics*, 60:537–571, Apr. 1988.

[128] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE*

Transactions on Circuits and Systems for Video Technology, 13(7):560–576, Jul. 2003.

[129] 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 International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 171–179, 2008.

[130] L.P. Yaroslavsky. *Digital Picture Processing.* Springer Press, 1985.

[131] I. Zachevsky and Y.Y. Zeevi. On the statistics of natural stochastic textures and their application in image processing. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 5829–5833, May 2014.

[132] E. Ziegler, M. Rouillard, E. André, T. Coolen, J. Stender, E. Balteau, C. Phillips, and G. Garraux. Mapping track density changes in nigrostriatal and extranigral pathways in Parkinson's disease. *NeuroImage*, 99(0):498–508, Oct. 2014.

Publication I

M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising using separable 4D nonlocal spatiotemporal transforms. In *Proceedings of the SPIE Electronic Imaging*, volume 7870, Jan. 2011, DOI: 10.1117/12.872569

Video denoising using separable 4-D nonlocal spatiotemporal transforms

Matteo Maggioni°, Giacomo Boracchi⋆, Alessandro Foi°, Karen Egiazarian°

°Department of Signal Processing, Tampere University of Technology, Finland;
⋆Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

ABSTRACT

We propose a powerful video denoising algorithm that exploits temporal and spatial redundancy characterizing natural video sequences. The algorithm implements the paradigm of nonlocal grouping and collaborative filtering, where a higher-dimensional transform-domain representation is leveraged to enforce sparsity and thus regularize the data. The proposed algorithm exploits the mutual similarity between 3-D spatiotemporal volumes constructed by tracking blocks along trajectories defined by the motion vectors. Mutually similar volumes are grouped together by stacking them along an additional fourth dimension, thus producing a 4-D structure, termed group, where different types of data correlation exist along the different dimensions: local correlation along the two dimensions of the blocks, temporal correlation along the motion trajectories, and nonlocal spatial correlation (i.e. self-similarity) along the fourth dimension. Collaborative filtering is realized by transforming each group through a decorrelating 4-D separable transform and then by shrinkage and inverse transformation. In this way, collaborative filtering provides estimates for each volume stacked in the group, which are then returned and adaptively aggregated to their original position in the video. Experimental results demonstrate the effectiveness of the proposed procedure which outperforms the state of the art.

Keywords: Video denoising, nonlocal methods, adaptive transforms, motion estimation

1. INTRODUCTION

The large number of practical applications involving digital videos has motivated a significant interest in denoising solutions, and the literature contains a plethora of such algorithms (see[1,2] for a comprehensive overview). At the moment, the most effective approach in restoring images or videos exploits the redundancy given by the *nonlocal* similarity between patches at different locations within the data.[3] Algorithms based on this approach have been proposed for various signal processing problems, and mainly for denoising.[2–12] Among these methods, we especially mention the BM3D algorithm,[7] which represents the state of the art in image denoising. BM3D relies on the so-called grouping and collaborative filtering paradigm: the observation is processed in a blockwise manner and mutually similar 2-D image blocks are stacked into a 3-D group (grouping), which is then filtered through a transform-domain shrinkage (collaborative filtering), simultaneously providing different estimates for each grouped block. These estimates are then returned to their respective locations and eventually aggregated into the estimate of the image. In doing so, BM3D leverages the spatial correlation of natural images both at the nonlocal and local level, due to the abundance of mutually similar patches and to the high correlation of image data within each patch, respectively. The BM3D filtering scheme has been applied successfully to video denoising (V-BM3D),[8] as well as to several other applications including image and video super-resolution,[11–13] image sharpening,[10] and image deblurring.[14]

In V-BM3D, groups are 3-D arrays of mutually similar blocks extracted from a set of consecutive frames of the video sequence. A group may include multiple blocks from the same frame, naturally exploiting in this way the nonlocal similarity. However, it is typically along the temporal dimension that most mutually similar blocks can be found. It is well known that motion-compensated videos[15] are extremely smooth along the temporal axis and this fact is exploited by nearly all modern video-coding techniques. As shown by the experimental analysis

This work was supported by the Academy of Finland (project no. 213462, Finnish Programme for Centres of Excellence in Research 2006-2011, project no. 118312, Finland Distinguished Professor Programme 2007-2010, and project no. 129118, Postdoctoral Researcher's Project 2009-2011).

in,[9] even when motion is present, the similarity along the motion trajectories is much stronger than the nonlocal similarity existing within an individual frame. In spite of this, in V-BM3D the blocks are grouped regardless of whether their similarity is due to the tracking of motion along time or to the nonlocal spatial self-similarity within each frame. In other words, the filtering in V-BM3D is not able to distinguish between temporal versus spatial nonlocal similarity. We recognize it as a conceptual as well as practical weakness of the algorithm: as simple experiments can demonstrate, increasing the number of spatially self-similar blocks in a V-BM3D group does not lead to an improvement in the final result and instead it most often leads to a systematic degradation.

This work proposes V-BM4D, a novel video-denoising approach that, to overcome the above weaknesses, separately exploits the temporal and spatial redundancy in the video sequence. For the sake of clarity and because of space limitation, we present V-BM4D for denoising only, although it can be implemented for a variety of other video filtering applications. The core element of V-BM4D is the spatiotemporal volume, a 3-D structure formed by a sequence of blocks extracted from the noisy video following a specific trajectory (obtained, for example, by concatenating motion vectors along time).[16,17] Thus, contrary to V-BM3D, V-BM4D does not group blocks, but mutually similar spatiotemporal volumes according to a nonlocal search procedure. Hence, these groups are 4-D stacks of 3-D volumes and the collaborative filtering is then performed via a separable 4-D spatiotemporal transform. The transform takes advantage of the following three types of correlation that characterize natural video sequences:

- the local spatial correlation between pixels in each block of a volume;
- the local temporal correlations between blocks of each volume;
- the nonlocal spatial and temporal correlation between grouped volumes.

The 4-D group spectrum is thus highly sparse, which makes the shrinkage more effective than in V-BM3D and results in the superior performance of V-BM4D in terms of noise reduction.

The paper is organized as follows: Section 3 presents a formal definition of the fundamental steps of the algorithm, while Section 4 describes the implementation aspects, with particular attention to the computation of motion vectors; experiments are illustrated and discussed in Section 5.

2. OBSERVATION MODEL

We consider the observed video as a noisy image sequence $z : X \times T \to \mathbb{R}$ defined as

$$z(\mathbf{x}, t) = y(\mathbf{x}, t) + \eta(\mathbf{x}, t), \qquad x \in X,\, t \in T, \tag{1}$$

where y is the original video, $\eta(\cdot, \cdot) \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. Gaussian noise, and $(\mathbf{x}, t)$ are the 3-D spatiotemporal coordinates belonging to the spatial domain $X \subset \mathbb{Z}^2$ and time domain $T \subset \mathbb{Z}$, respectively. The frame of the video z at time index t is denoted by $z(X, t)$.

3. BASIC ALGORITHM

The aim of the proposed algorithm is to provide an estimate $\hat{y}$ of the original video y from the observed data z. According to the BM3D paradigm, the V-BM4D algorithm comprises three fundamental steps, specifically grouping, collaborative filtering and aggregation. These steps are performed for every spatiotemporal volume of the video.

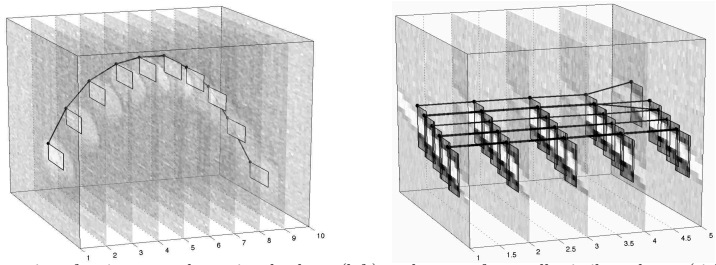


Figure 1. Illustration of trajectory and associated volume (left), and group of mutually similar volumes (right) calculated for the sequence *Tennis* corrupted by white Gaussian noise with $\sigma = 20$.

3.1 Spatiotemporal Volumes

Let $B_z(\mathbf{x}_0, t_0)$ denote a square block of fixed size $N \times N$ extracted from the noisy video z; without loss of generality, the coordinates $(\mathbf{x}_0, t_0)$ identify the top-left pixel of the block in the frame $z(X, t_0)$. A spatiotemporal volume is the 3-D sequence of blocks built following a specific trajectory along time. The trajectory associated to $(\mathbf{x}_0, t_0)$ is defined as

$$\mathrm{Traj}(\mathbf{x}_0, t_0) = \Big\{(\mathbf{x}_j, t_0 + j)\Big\}_{j=-h^-}^{h^+}, \tag{2}$$

where the elements $(\mathbf{x}_j, t_0 + j)$ are time-consecutive coordinates, each of these represents the position of the reference block $B_z(\mathbf{x}_0, t_0)$ within the neighboring frames $z(X, t_0+j)$, $j = -h^-, \ldots, h^+$. For the sake of simplicity, in this section it is assumed $h^- = h^+ = h$ for all $(\mathbf{x}, t) \in X \times T$ and the considerations concerning the general case are postponed in Section 4.

The trajectories can be either computed from the noisy video (as shown in Section 4.1), or, when given a coded video, they can be obtained by concatenating motion vectors. In what follows we assume that, for each $(\mathbf{x}_0, t_0) \in X \times T$, a trajectory $\mathrm{Traj}(\mathbf{x}_0, t_0)$ is given and thus the 3-D spatiotemporal volume in $(\mathbf{x}_0, t_0)$ can be determined as

$$V_z(\mathbf{x}_0, t_0) = \big\{B_z(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \mathrm{Traj}(\mathbf{x}_0, t_0)\big\}, \tag{3}$$

where the subscript z specifies that the volumes are extracted from the noisy video. The length of a volume $V_z(\mathbf{x}_i, t_i)$ is defined as

$$L_i = h^- + h^+ + 1. \tag{4}$$

3.2 Grouping

Groups are stacks of mutually similar volumes and constitute the nonlocal element of V-BM4D. Mutually similar volumes are determined with a nonlocal search procedure as in.[7] Let $\mathrm{Ind}(\mathbf{x}_0, t_0)$ be the set of indexes identifying volumes that, according to a distance operator δ^{v}, are similar to $V_z(\mathbf{x}_0, t_0)$:

$$\mathrm{Ind}(\mathbf{x}_0, t_0) = \big\{(\mathbf{x}_i, t_i) : \delta^{\mathrm{v}}(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)) < \tau_{\mathrm{match}}\big\}. \tag{5}$$

The parameter $\tau_{\mathrm{match}} > 0$ controls the minimum degree of similarity among volumes; the distance δ^{v} is typically the ℓ^2-norm of the difference between two volumes.

The group associated to the reference volume $V_z(\mathbf{x}_0, t_0)$ is then

$$G_z(\mathbf{x}_0, t_0) = \big\{V_z(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \mathrm{Ind}(\mathbf{x}_0, t_0)\big\}. \tag{6}$$

In (6), we implicitly assume that the 3-D volumes are stacked along a fourth dimension, and hence the groups are 4-D data structures. Note that since $\delta^{\mathrm{v}}(V_z, V_z) = 0$, every group $G_z(\mathbf{x}_0, t_0)$ contains, at least, its reference volume $V_z(\mathbf{x}_0, t_0)$. Figure 1 shows examples of trajectories, volumes and groups.

3.3 Collaborative Filtering

In the general formulation of the grouping and collaborative-filtering approach for a d-dimensional signal,[7] groups are $(d+1)$-dimensional structures of similar d-dimensional elements, which are then jointly filtered. In particular, each of the grouped elements influences the filtered output of all the other elements of the group: this is the basic idea of collaborative filtering. It is typically realized with the following steps: firstly a $(d+1)$-dimensional separable linear transform is applied to the group, then the transformed coefficients are shrunk, for example by hard-thresholding or by Wiener filtering, and finally the $(d+1)$-dimensional transform is inverted to obtain an estimate for each grouped element.

The core elements of V-BM4D are the spatiotemporal volumes ($d = 3$), and thus the collaborative filtering performs a 4-D separable linear transform $\mathcal{T}_{4D}$ on each 4-D group $G_z(\mathbf{x}_0, t_0)$, and provides an estimate for each grouped volume V_z:

$$\hat{G}_y(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{-1}\big(\Upsilon\left(\mathcal{T}_{4D}\left(G_z(\mathbf{x}_0, t_0)\right)\right)\big), \tag{7}$$

where Υ denotes a generic shrinkage operator. The filtered 4-D group $\hat{G}_y(\mathbf{x}_0, t_0)$ is composed of volumes $\hat{V}_y(\mathbf{x}, t)$

$$\hat{G}_y(\mathbf{x}_0, t_0) = \big\{\hat{V}_y(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \mathrm{Ind}(\mathbf{x}_0, t_0)\big\}, \tag{8}$$

with each $\hat{V}_y$ being an estimate of the corresponding volume V_y extracted from the original video y.

3.4 Aggregation

The groups $\hat{G}_y$ constitute a very redundant representation of the video, because in general the volumes $\hat{V}_y$ overlap and, within the overlapping parts, the collaborative filtering provides multiple estimates at the same coordinates $(\mathbf{x}, t)$. For this reason, the estimates are aggregated through a convex combination with adaptive weights. In particular, the estimate $\hat{y}$ of the original video is computed as

$$\hat{y} = \frac{\sum_{(\mathbf{x}_0,t_0)\in X\times T}\big(\sum_{(\mathbf{x}_i,t_i)\in\mathrm{Ind}(\mathbf{x}_0,t_0)} w_{(\mathbf{x}_0,t_0)}\hat{V}_y(\mathbf{x}_i,t_i)\big)}{\sum_{(\mathbf{x}_0,t_0)\in X\times T}\big(\sum_{(\mathbf{x}_i,t_i)\in\mathrm{Ind}(\mathbf{x}_0,t_0)} w_{(\mathbf{x}_0,t_0)}\chi_{(\mathbf{x}_i,t_i)}\big)}, \tag{9}$$

where we assume $\hat{V}_y(\mathbf{x}_i, t_i)$ to be zero-padded outside its domain, $\chi_{(\mathbf{x}_i,t_i)} : X \times T \to \{0, 1\}$ is the characteristic function (indicator) of the support of the volume $\hat{V}_y(\mathbf{x}_i, t_i)$, and the aggregation weights $w_{(\mathbf{x}_0,t_0)}$ are different for different groups. The particular choice of the aggregation weights depends on the result of shrinkage in the collaborative filtering: typically the weights are defined so that the sparser is the shrunk 4-D spectrum $\hat{G}_y(\mathbf{x}_0, t_0)$, the larger is the weight $w_{(\mathbf{x}_0,t_0)}$. In particular, the weights can be effectively defined to be inversely proportional to the total sample variance of the estimate of the corresponding groups.[7]

4. IMPLEMENTATION ASPECTS

4.1 Computation of the Trajectories

In our implementation, we construct trajectories by concatenation of motion vectors which are defined as follows.

4.1.1 Similarity criterion

Motion of a block is generally tracked by identifying the most similar block in the subsequent (and precedent) frame. However, since we deal with noisy signals, prior information about motion smoothness can be exploited to improve the tracking. In particular, provided a rough guess $\hat{\mathbf{x}}_i(t_j)$ of the future (or past) location of the block $B_z(\mathbf{x}_i, t_i)$ at the time $t_j = t_i + 1$ ($t_j = t_i - 1$), we define the similarity between $B_z(\mathbf{x}_i, t_i)$ and $B_z(\mathbf{x}_j, t_j)$, through a penalized quadratic difference

$$\delta^{\mathrm{b}}\big(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_j, t_j)\big) = \frac{||B_z(\mathbf{x}_i, t_i) - B_z(\mathbf{x}_j, t_j)||_2^2}{N^2} + \gamma_d\,||\hat{\mathbf{x}}_i(t_j) - \mathbf{x}_j||_2\,, \tag{10}$$

where $\hat{\mathbf{x}}_i(t_j)$ is the predicted position of $B_z(\mathbf{x}_i, t_i)$ in the frame $z\,(X, t_j)$, and $\gamma_d \in \mathbb{R}^+$ is the penalization parameter. Whenever $\hat{\mathbf{x}}_i(t_j)$ is not available, we consider the lack of motion as the most likely condition and we set $\hat{\mathbf{x}}_i(t_j) = \mathbf{x}_i$.

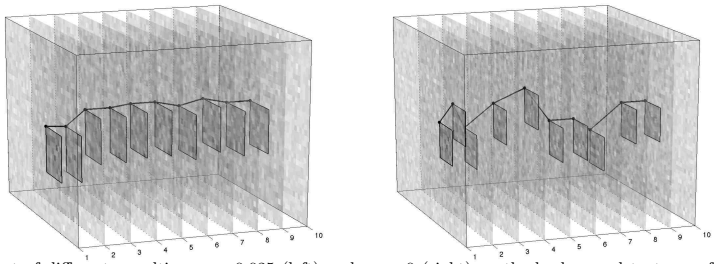


Figure 2. Effect of different penalties $\gamma_d = 0.025$ (left) and $\gamma_d = 0$ (right) on the background textures of the sequence *Tennis* corrupted by Gaussian noise with $\sigma = 20$. The initial positions at time $t = 1$ are equal in both experiments.

V-BM4D repeatedly uses the minimization of (10) to construct the trajectory (2). Formally, the motion of $B_z(\mathbf{x}_i, t_i)$ from time t_i to $t_i + 1$ is determined by the position $\mathbf{x}_j$ that minimizes (10)

$$\mathbf{x_j} = \underset{x_k \in \mathcal{N}}{\arg\min} \left\{ \delta^{\mathrm{b}}\big(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_k, t_i + 1)\big) < \tau_{\mathrm{traj}} \right\}, \tag{11}$$

where $\mathcal{N}$ is a restriction in the frame $z(X, t_i + 1)$ applied by an adaptive search neighborhood (further details are given in Section 4.1.3). Nevertheless, even though a minimizer for (10) can always be found, we interrupt the trajectory whenever the corresponding minimum distance δ^{b} exceeds a fixed parameter $\tau_{\mathrm{traj}} \in \mathbb{R}^+$, which determines the minimum accepted similarity along spatiotemporal volumes, to effectively deal with occlusions and changes of scene. Figure 2 illustrates trajectories estimated using different penalization parameters. Observe that the penalization term is essential when tracking blocks belonging to areas covered by homogeneous texture, in fact, as shown in the right image of Figure 2, without a position-dependent distance metric, the trajectories would be mainly determined by noise, and, for this reason, the collaborative filtering would be less effective.

4.1.2 Location prediction

As soon as the motion of a block at two consecutive spatiotemporal locations $(\mathbf{x}_{i-1}, t_i - 1)$ and $(\mathbf{x}_i, t_i)$ has been determined, we can define the motion vector (velocity) $\mathbf{v}(\mathbf{x}_i, t_i) = \mathbf{x}_{i-1} - \mathbf{x}_i$. Hence, under the assumption of smooth motion, we define the guess $\hat{\mathbf{x}}_i(t_i + 1)$ as

$$\hat{\mathbf{x}}_i(t_i + 1) = \mathbf{x}_i + \gamma_p \cdot \mathbf{v}(\mathbf{x}_i, t_i), \tag{12}$$

where $\gamma_p \in [0, 1]$ is a weighting factor of the prediction. Analogous prediction can be made for $\hat{\mathbf{x}}_{i-1}(t_i - 1)$, when we look for precedent blocks in the sequence.

4.1.3 Search neighborhood

Because of the penalty term $\gamma_d \, ||\hat{\mathbf{x}}_i(t_j) - \mathbf{x}_j||_2$, the minimizer of (10) is likely close to $\hat{\mathbf{x}}_i(t_j)$. We therefore restrict the minimization of (10) to a spatial search neighborhood $\mathcal{N}$ centered at $\hat{\mathbf{x}}_i(t_j)$. The size $N_{PR} \times N_{PR}$ of this neighborhood can be adapted based on the velocity (magnitude of motion vector) of the tracked block by setting

$$N_{PR} = N_S \cdot \left(1 - \gamma_w \cdot e^{-\frac{||\mathbf{v}(\mathbf{x}_i, t_i)||_2^2}{2 \cdot \sigma_w^2}}\right), \tag{13}$$

where N_S is the maximum size of $\mathcal{N}$, $\gamma_w \in [0, 1]$ is a scaling factor and $\sigma_w > 0$ is a tuning parameter. As the velocity increases, N_{PR} approaches N_S accordingly to σ_w; conversely, when the velocity is zero $N_{PR} = N_S(1 - \gamma_w)$. By setting a proper value of σ_w we can control how fast the exponential term approaches zero, or, in other words, how permissive is the window shrinkage with respect to the velocity of the tracked block. For instance, considering the same velocity $\mathbf{v}$ for a given block and using increasing values of σ_w in (13), we would obtain smaller windows, because the decay of the function would be slower.

4.2 Sub-volume Extraction

So far, the number of frames spanned by all the trajectories has been assumed fixed. However, because of occlusions, scene changes or heavy noise, any trajectory $\mathrm{Traj}(\mathbf{x}_i, t_i)$ can be interrupted at any time, as determined by the parameter τ_{traj}. Thus, if $\left[t_i - h_i^-, t_i + h_i^+\right]$ is the temporal extent of the trajectory $\mathrm{Traj}(\mathbf{x}_i, t_i)$, we have that

$$0 \leq h_i^- \leq h\,, \quad 0 \leq h_i^+ \leq h, \tag{14}$$

where h denotes the maximum forward and backward extent of trajectories (and thus volumes) allowed in the algorithm.

As a result, during grouping, V-BM4D may stack together volumes having different lengths. Nevertheless, because of the separability of the transform $\mathcal{T}_{4D}$, every group $G_z(\mathbf{x}_i, t_i)$ has to be composed of volumes of equal length. Thus, in the current implementation of grouping we consider, for each reference volume $V_z(\mathbf{x}_0, t_0)$, only volumes $V_z(\mathbf{x}_i, t_i)$ such that $t_i = t_0$, $h_i^- \geq h_0^-$ and $h_i^+ \geq h_0^+$. In this case, V-BM4D extracts from $V_z(\mathbf{x}_i, t_i)$ the sub-volume with temporal extent $[t_0 - h_0^-, t_0 + h_0^+]$, denoted as $\mathcal{E}_{L_0}\big(V_z(\mathbf{x}_i, t_i)\big)$. There are obviously many other, less restrictive, possibilities for extracting sub-volumes of length L_0 from longer volumes, however, the one we implemented aims at limiting the complexity while maintaining a high correlation within the grouped volumes.

In the grouping, the distance operator δ^{v} is the ℓ^2-norm of the difference between time-synchronous volumes normalized with respect to their lengths

$$\delta^{\mathrm{v}}\big(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)\big) = \big|\big|V_z(\mathbf{x}_0, t_0) - \mathcal{E}_{L_0}\big(V_z(\mathbf{x}_i, t_i), t_0\big)\big|\big|_2^2 / L_0, \tag{15}$$

thus providing larger weight to the volumes belonging to groups having sparser representation in $\mathcal{T}_{4D}$ domain.

4.3 Two-Stage Implementation with Collaborative Wiener Filtering

The general procedure described in Section 3 is implemented in two cascading stages, both composed of the grouping, collaborative filtering and aggregation steps.

4.3.1 Hard-thresholding stage

In the first stage, volumes are extracted from the noisy video z, and groups are then formed using the similarity measure δ^{v}-operator (15), and the predefined threshold $\tau_{\mathrm{match}}^{\mathrm{ht}}$. Collaborative filtering is realized by hard thresholding in 4-D transform domain each group $G_z(\mathbf{x}, t)$:

$$\hat{G}_y^{\mathrm{ht}}(\mathbf{x}, t) = \mathcal{T}_{4D}^{\mathrm{ht}^{-1}}\left(\Upsilon^{\mathrm{ht}}\left(\mathcal{T}_{4D}^{\mathrm{ht}}\left(G_z(\mathbf{x}_0, t_0)\right)\right)\right),\ (\mathbf{x}, t) \in X \times T, \tag{16}$$

where $\mathcal{T}_{4D}^{\mathrm{ht}}$ is the 4-D transform and Υ^{ht} is the hard-threshold operator with threshold $\sigma\lambda_{4D}$.

The outcome of hard-thresholding stage, $\hat{y}^{\mathrm{ht}}$, is obtained by aggregation of all the estimated groups $\hat{G}_y^{\mathrm{ht}}(\mathbf{x}, t)$. The weights $w_{(\mathbf{x}_0, t_0)}^{\mathrm{ht}}$ in the aggregation (9) are inversely proportional to the number $N_{(\mathbf{x}_0, t_0)}^{\mathrm{ht}}$ of non-zero coefficients of the corresponding hard-thresholded group $\hat{G}_y^{\mathrm{ht}}(\mathbf{x}_0, t_0)$:

$$w_{(\mathbf{x}_0, t_0)}^{\mathrm{ht}} = \frac{1}{N_{(\mathbf{x}_0, t_0)}^{\mathrm{ht}}}. \tag{17}$$

4.3.2 Wiener filtering stage

In the second stage, new trajectories $\mathrm{Traj}_{\hat{y}^{\mathrm{ht}}}$ are extracted from the basic estimate $\hat{y}^{\mathrm{ht}}$, and the grouping is performed on the new volumes $V_{\hat{y}^{\mathrm{ht}}}$. Volume matching is still performed through the δ^{v}-distance, but using a different threshold $\tau_{\mathrm{match}}^{\mathrm{wie}}$. The set of volume indexes $\mathrm{Ind}_{\hat{y}^{\mathrm{ht}}}(\mathbf{x}, t)$ resulting from similarity search are used to construct two sets of groups G_z and $G_{\hat{y}^{\mathrm{ht}}}$, composed by volumes extracted from the noisy video z and from the estimate y^{ht}, respectively.

Table 1. Parameter settings of V-BM4D for the first (hard-thresholding) and the second (Wiener-filtering) stage. The parameters γ_d, τ_{traj} and τ_{match} vary according to the noise, as shown in Figure 3.

Stage	N	N_S	N_G	h	M	λ_{4D}	γ_p	γ_w	σ_w	N_{step}	γ_d	τ_{traj}	τ_{match}
Hard thr.	8	11	19	4	32	2.7	0.3	0.5	1	6	$\gamma_d(\sigma)$	$\tau_{\text{traj}}(\sigma)$	$\tau_{\text{match}}(\sigma)$
Wiener filt.	7		27		8	*Unused*				4	0.005	1	13.5

Collaborative filtering is hence performed using an empirical Wiener filter in $\mathcal{T}_{4D}^{\text{wie}}$ transform domain, whose shrinkage coefficients are computed from the energy of the 4-D spectrum of the basic estimate group $G_{\hat{y}^{\text{ht}}}$

$$\mathbf{W}(\mathbf{x}_0, t_0) = \frac{\left|\mathcal{T}_{4D}^{\text{wie}}\left(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)\right)\right|^2}{\left|\mathcal{T}_{4D}^{\text{wie}}\left(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)\right)\right|^2 + \sigma^2}, \tag{18}$$

Shrinkage is realized as element-by-element multiplication between the 4-D transform coefficients of the group $G_z(\mathbf{x}_0, t_0)$ extracted from the noisy video z and the Wiener coefficients $\mathbf{W}(\mathbf{x}_0, t_0)$. Subsequently, we obtain the group of volumes estimates by inverting the 4-D transform as

$$\hat{G}_y^{\text{wie}}(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{\text{wie}^{-1}}\left(\mathbf{W}(\mathbf{x}_0, t_0) \cdot \mathcal{T}_{4D}^{\text{wie}}\left(G_z(\mathbf{x}_0, t_0)\right)\right). \tag{19}$$

The global final estimate $\hat{y}^{\text{wie}}$ is computed by the aggregation (9), using the weights

$$w_{(\mathbf{x}_0, t_0)}^{\text{wie}} = ||\mathbf{W}(\mathbf{x}_0, t_0)||_2^{-2}. \tag{20}$$

5. EXPERIMENTS

In this section we present the experimental results obtained with a C/MATLAB implementation of the V-BM4D algorithm, and we compare it against V-BM3D*, as it represents the state of the art in video denoising. Observations z are obtained by synthetically adding Gaussian noise to greyscale image sequences, according to (1). The denoising performance is measured using the PSNR as a global measure for the whole processed video:

$$\text{PSNR} = -10 \log_{10}\left(255^{-2}|X||T| \sum_{(\mathbf{x},t)\in X\times T} \left(y((\mathbf{x},t)) - \hat{y}(\mathbf{x},t)\right)^{-2}\right), \tag{21}$$

where $|X|$ and $|T|$ stand for the cardinality of X and T, respectively.

The transforms employed in the collaborative filtering are similar to those in:[7,8] in the hard-thresholding stage $\mathcal{T}_{4D}^{\text{ht}}$ is a 4-D separable composition of 1-D biorthogonal wavelet in both spatial dimensions, 1-D DCT in the temporal dimension, and 1-D Haar wavelet in the fourth (grouping) dimension while, in the Wiener filtering stage, $\mathcal{T}_{4D}^{\text{wie}}$ uses a 2-D DCT for the spatial dimension. Note that, because of the Haar transform, the cardinality M of each group must be a power of 2. In order to reduce the complexity of the grouping phase, we restrict the search of similar volumes within a $N_G \times N_G$ neighborhood centered around the coordinates of the reference volume, moreover, to lighten the computational complexity of the grouping, a step of $N_{\text{step}} \in \mathbb{N}$ pixels in both horizontal and vertical directions separates each processed volume. Notwithstanding the trajectory of every possible volume in the video must be computed beforehand, because any volume is a potential candidate element of every group.

The two stages share some of the parameters such as: the search neighborhoods for the trajectory calculation N_S, the temporal extent h, the weights γ_p of (12) and γ_w, σ_w of (13), while the block size N, the grouping window N_G, the group size M, and the processing step N_{step} are different, and λ_{4D} is used in the first stage only. Observe that we restrict the volumes contained in the groups to be the largest power of 2 smaller than or equal to the minimum value between the original cardinality of the groups and M itself.

*Matlab code at `http://www.cs.tut.fi/~foi/GCF-BM3D/`.

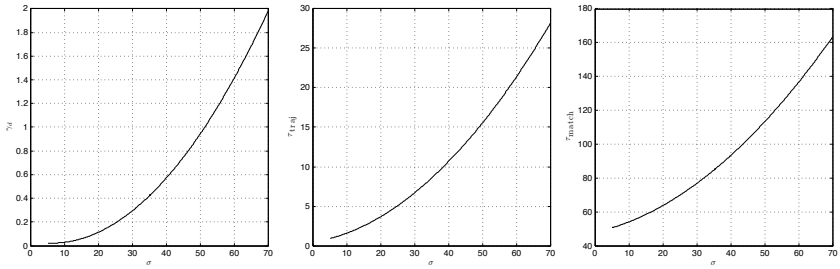
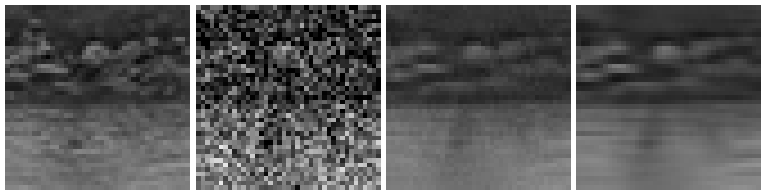


Figure 3. Parameters depending on σ in the hard-thresholding stage. The functions are the quadratic polynomials approximation of the optimum parameters obtained from the Nelder-Mead simplex direct search algorithm applied on a set of test sequences corrupted by white Gaussian noise having different values of σ. The functions are built such that their coefficients maximize the average PSNR of the test sequences along each value of σ. In particular we use *Salesman*, *Tennis*, *Flower Garden Miss America*, *Coastguard*, *Foreman*, *Bus*, and *Bicycle*.



(a) Original y (b) Noisy z (c) Result y^{ht} of the first stage (d) Result y^{wie} of the second stage

Figure 4. Visual comparison of the sequence *Coastguard* corrupted by white Gaussian noise with standard deviation $\sigma = 40$, denoised after the first and second stage of V-BM4D.

The parameters involved in the motion estimation and in the grouping, that is γ_d, τ_{traj} and τ_{match}, vary with σ. Intuitively, in order to compensate the effects of the noise, the larger σ is, the larger the thresholds controlling blocks and volumes matching become. The behavior of such parameters w.r.t. σ is determined following an empirical approach. First we compute the parameters that maximize the V-BM4D restoration performance (PSNR) on a set of sequences, where σ is known. Then the restoration performance is maximized using the Nelder-Mead simplex direct search algorithm[18,19] in a multivariate space, thus finding the optimum value of the triplet (γ_d, τ_{traj}, τ_{match}) for eight test video corrupted by i.i.d. white Gaussian noise having eight different value of σ, ranging from 5 to 70. Subsequently, we approximate the behavior of the three parameters as a function of σ using a quadratic polynomial for each variable in the domain (γ_d, τ_{traj}, τ_{match}) maximizing the total PSNR of the test sequences. The resulting fit is

$$\gamma_d(\sigma) = 0.0005 \cdot \sigma^2 - 0.0059 \cdot \sigma + 0.0400, \tag{22}$$

$$\tau_{\text{traj}}(\sigma) = 0.0047 \cdot \sigma^2 + 0.0676 \cdot \sigma + 0.4564, \tag{23}$$

$$\tau_{\text{match}}(\sigma) = 0.0171 \cdot \sigma^2 + 0.4520 \cdot \sigma + 47.9294. \tag{24}$$

The above functions are shown in Figure 3: experimentally they were found to be a good approximation of the optimum (γ_d, τ_{traj}, τ_{match}). Note that during the second stage such parameters can be considered constants independent of σ, because in the processed sequence $\hat{y}^{\text{ht}}$ the noise is considerably lower than in the observation z; this is evident when looking at the second and third image of Figure 4. Moreover, since in this stage the

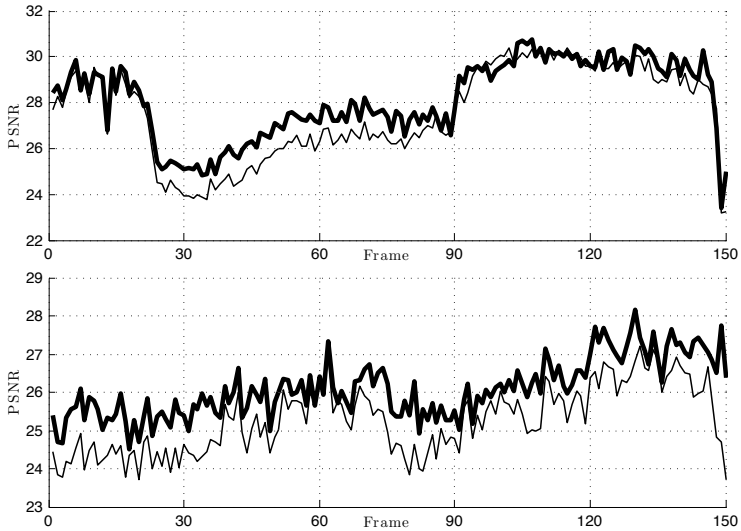


Figure 5. Frame-by-frame PSNR output of *Tennis* (top) and *Bus* (bottom) denoised by V-BM4D (thick line), and V-BM3D (thin line). The sequences are corrupted by i.i.d. white Gaussian noise with standard deviation $\sigma = 20$.

Table 2. Comparison between the PSNR (dB) outputs obtained from the proposed V-BM4D algorithm (top number in each cell), and the V-BM3D algorithm tuned with its default parameters[8] (bottom number in each cell). The test sequences are corrupted by i.i.d. Gaussian noise with zero mean and three different standard deviations σ.

σ	Video:	*Salesm.*	*Tennis*	*Fl. Gard.*	*Miss Am.*	*Coastg.*	*Foreman*	*Bus*	*Bicycle*
	Res.:	288×352	240×352	240×352	288×360	144×176	288×352	288×352	576×720
	Frames:	50	150	150	150	300	300	150	30
10	V-BM4D	**37.30**	**35.22**	**32.81**	**40.09**	**35.54**	**36.94**	**34.26**	**37.66**
	V-BM3D	37.21	34.68	32.11	39.61	34.78	36.46	33.32	37.62
20	V-BM4D	33.79	**31.59**	**28.63**	**37.98**	**31.94**	**33.67**	**30.26**	34.10
	V-BM3D	**34.04**	31.20	28.24	37.85	31.71	33.30	29.57	**34.18**
40	V-BM4D	**30.35**	**28.49**	**24.60**	**35.47**	**28.54**	**30.52**	**26.72**	**30.10**
	V-BM3D	29.93	27.99	24.33	35.45	28.27	29.97	26.28	30.02

trajectories and the grouping are determined from the basic estimate $\hat{y}^{\mathrm{ht}}$, there is no a straightforward relation with σ, the standard deviation of the noise corrupting the observation z.

The comparison against V-BM3D[8] is carried out using the set of parameters reported in Table 1. Table 2 compares the denoising performance in terms of PSNR of the two algorithms, applied to a set of standard video sequences corrupted by white Gaussian noise with increasing standard deviation $\sigma = \{10, 20, 40\}$, which is assumed known. Further details concerning the original sequences, such as the resolution and number of frames, are shown in the header of the table. As one can see, V-BM4D outperforms V-BM3D in nearly all the experiments, with PSNR improvement of up to 1 dB. It is particularly interesting to observe that V-BM4D handles effectively the sequences characterized by rapid motion and frequent scene changes, especially under

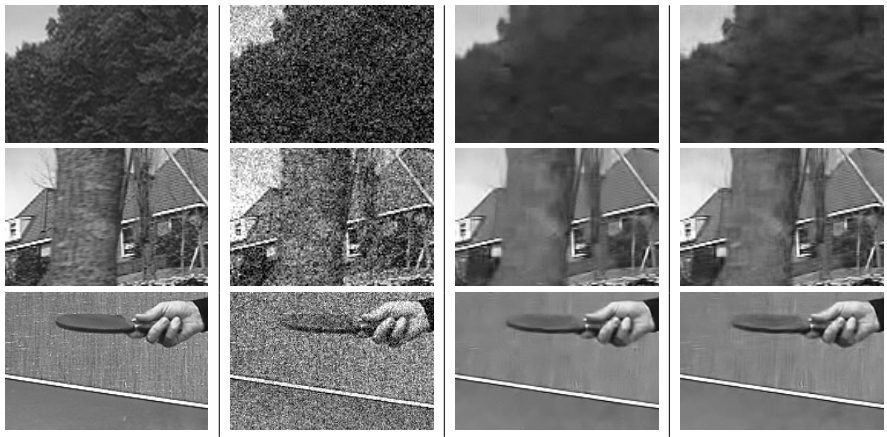


Figure 6. Visual comparison of the sequences, from top to bottom, *Bus*, *Flower Garden* and *Tennis* corrupted by white Gaussian noise with standard deviation $\sigma = 40$, denoised by the proposed algorithm V-BM4D and the V-BM3D algorithm.

heavy noise, as *Tennis*, *Flower Garden*, *Coastguard* and *Bus*. In particular, Figure 5 shows that as soon as the sequence presents a significant change in the scene, the denoising performance decreases significantly for the two algorithms, but, in these situations, V-BM4D requires much less frames to recover high PSNR values, as shown by the lower peaks at frame 30 and 90 of *Tennis* and around frame 75 of *Bus*.

Figure 6 offers a visual comparison of the performance of the two algorithms. As a subjective quality assessment, V-BM4D better preserves the textures, without introducing significant artifacts in the restored video: this is clearly visible in the tree leaves of the *Bus* sequence.

6. DISCUSSION AND CONCLUSIONS

Experiments show that V-BM4D outperforms V-BM3D in terms of measured performance (PSNR), and of visual appearance (Figure 6), thus achieving state-of-the-art results in video denoising. In particular, V-BM4D can restore much better than V-BM3D fine image details, even in sequences corrupted by heavy noise ($\sigma = 40$): this difference is clearly visible in the processed frames shown in Figure 6. Moreover, the comparison between V-BM3D and V-BM4D highlights that the temporal correlation is a key element in video denoising, and that it has to be adequately handled when designing nonlocal video restoration algorithms. We wish to remark that the computational complexity in V-BM4D is obviously higher than in V-BM3D, mainly because V-BM4D processes higher-dimensional arrays. Thus, V-BM4D can be a viable alternative to V-BM3D especially in applications where the highest restoration quality is paramount. Ongoing work addresses the parallelization of V-BM4D, leveraging GPU hardware.

REFERENCES

[1] Protter, M. and Elad, M., "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing* **18**(1), 27–35 (2009).

[2] Ghoniem, M., Chahir, Y., and Elmoataz, A., "Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs," *Signal Processing* **90**(8), 2445–2455 (2010). Special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.

[3] Katkovnik, V., Foi, A., Egiazarian, K., and Astola, J., "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision* **86**(1), 1–32 (2010).

[4] Buades, A., Coll, B., and Morel, J.-M., "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation* **4**(2), 490–530 (2005).

[5] Buades, A., Coll, B., and Morel, J.-M., "Nonlocal image and movie denoising," *Int. Journal of Computer Vision* **76**(2), 123–139 (2008).

[6] Li, X. and Zheng, Y., "Patch-based video processing: a variational bayesian approach," *IEEE Transactions on Circuits and Systems for Video Technology* **29**, 27–40 (January 2009).

[7] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.* **16** (August 2007).

[8] Dabov, K., Foi, A., and Egiazarian, K., "Video denoising by sparse 3D transform-domain collaborative filtering," in [*Proc. 15th European Signal Processing Conference, EUSIPCO*], (September 2007).

[9] Boracchi, G. and Foi, A., "Multiframe raw-data denoising based on block-matching and 3-D filtering for low-light imaging and stabilization," in [*Proceedings of LNLA 2008, the International Workshop on Local and Non-Local Approximation in Image Processing, 22 - 24 August, 2008 Lausanne, Switzerland*], (2008).

[10] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Joint image sharpening and denoising by 3D transform-domain collaborative filtering," in [*Proc. 2007 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2007*], (2007).

[11] Danielyan, A., Foi, A., Katkovnik, V., and Egiazarian, K., "Image and video super-resolution via spatially adaptive block-matching filtering," in [*Proc. Int. Workshop on Local and Non-Local Approx. in Image Process., LNLA 2008, Lausanne, Switzerland*], (August 2008).

[12] Danielyan, A., Foi, A., Katkovnik, V., and Egiazarian, K., "Image upsampling via spatially adaptive block-matching filtering," in [*Proc. of 16th European Signal Processing Conference, EUSIPCO 2008*], (2008).

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

[14] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image restoration by sparse 3D transform-domain collaborative filtering," in [*Proc. SPIE Electronic Imaging, San Jose (CA), USA*], **6812** (January 2008).

[15] Hang, H.-M., Chou, Y.-M., and Cheng, S.-C., "Motion estimation for video coding standards," *Journal of VLSI Signal Processing Systems* **17**(2/3), 113–136 (1997).

[16] Megret, R. and Dementhon, D., "A survey of spatio-temporal grouping techniques," tech. rep. (2002).

[17] Basharat, A., Zhai, Y., and Shah, M., "Content based video matching using spatiotemporal volumes," *Comput. Vis. Image Underst.* **110**(3), 360–377 (2008).

[18] Nelder, J. A. and Mead, R., "A simplex method for function minimization," *Computer Journal* **7**, 308–313 (1965).

[19] Lagarias, J. C., Reeds, J. A., Wright, M. H., and Wright, P. E., "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal of Optimization* **9**, 112–147 (1998).

Publication II

M. Maggioni and A. Foi. Nonlocal transform-domain denoising of volumetric data with groupwise adaptive variance estimation. In *Proceedings of the SPIE Electronic Imaging*, volume 8296, Jan. 2012, DOI: 10.1117/12.912109

Nonlocal transform-domain denoising of volumetric data with groupwise adaptive variance estimation

Matteo Maggioni and Alessandro Foi

Department of Signal Processing, Tampere University of Technology, Finland
firstname.lastname@tut.fi

ABSTRACT

We propose an extension of the BM4D volumetric filter to the denoising of data corrupted by spatially non-uniform noise. BM4D implements the grouping and collaborative filtering paradigm, where similar cubes of voxels are stacked into a four-dimensional "group". Each group undergoes a sparsifying four-dimensional transform, that exploits the local correlation among voxels in each cube and the nonlocal correlation between corresponding voxels of different cubes. Thus, signal and noise are effectively separated in transform domain. In this work we take advantage of the sparsity induced by the four-dimensional transform to provide a spatially adaptive estimation of the local noise variance by applying a robust median estimator of the absolute deviation to the spectrum of each filtered group. The adaptive variance estimates are then used during coefficients shrinkage. Finally, the inverse four-dimensional transform is applied to the filtered group, and each individual cube estimate is adaptively aggregated at its original location.

Experiments on medical data corrupted by spatially varying Gaussian and Rician noise demonstrate the efficacy of the proposed approach in volumetric data denoising. In case of magnetic resonance signals, the adaptive variance estimate can be also used to compensate the estimation bias due to the non-zero-mean errors of the Rician-distributed data.

Keywords: Volumetric data denoising, nonlocal methods, adaptive transforms, non-uniform noise, variance estimation, magnetic resonance imaging

1. INTRODUCTION

The most powerful methods for image restoration rely on the self-similarity and nonlocality characteristics of natural images. The state-of-the-art BM3D image denoising algorithm[1] couples the nonlocal filtering paradigm proposed in[2,3] with the grouping and collaborative filtering approach. The method leverages an enhanced sparse representation in transform domain enabled by the grouping of similar 2-D image patches into 3-D data arrays which are called "groups". Collaborative filtering includes three successive steps: 3-D transformation of a group, shrinkage of transform spectrum, and inverse 3-D transformation. Due to the similarity between the grouped blocks, the transform can achieve a highly sparse representation of the true signal so that the noise can be effectively attenuated by shrinkage. In this way, the collaborative filtering reveals even the finest details shared by grouped fragments and at the same time it preserves the essential unique features of each individual fragment.

The grouping and collaborative paradigm can be also effectively exploited in volumetric data restoration and, in particular, it is the foundation of recently proposed BM4D volumetric denoising algorithm.[4] Instead of using blocks of pixels as basic data patches, BM4D naturally utilizes similar 3-D cubes of voxels which are stacked together to form the 4-D group. The local correlation present among voxels in each cube as well as the nonlocal correlation between the corresponding voxels of different cubes induce a sparse representation of the group in transform domain. After collaborative filtering and inverse transformation, we obtain individual estimates of the grouped cubes, which are then aggregated at their original locations using adaptive weights.

This work was supported by the Academy of Finland (project no. 213462, Finnish Programme for Centres of Excellence in Research 2006-2011, project no. 129118, Postdoctoral Researcher's Project 2009-2011, and project no. 252547, Academy Research Fellow 2011-2016), and by Tampere Graduate School in Information Science and Engineering (also known as TISE).

The original BM4D volumetric denoising algorithm has been utilized in the denoising of magnetic resonance (MR) images corrupted by either Gaussian- or Rician-distributed noise having uniform variance.[4] However, in some applications, e.g., parallel acquisition techniques such as sensitivity encoding (SENSE)[5] or generalized autocalibrating partially parallel acquisitions (GRAPPA),[6] the noise corrupting the observed data is characterized by a spatially varying variance. In literature, the approaches addressing this problem generally adhere to the following scheme: at first, the variance of the noise is locally estimated, then, a filtering technique, adjusted depending on the strength of the estimated noise, is adaptively applied to the data. For example, in[7] the variance is estimated using the noise distribution map and the denoising is performed via anisotropic diffusion kernels. A different approach, presented in,[8] relies on the high-frequency subband of the wavelet coefficients to estimate the variance, and on a coefficients shrinkage in transform domain to filter the noisy observation. A limitation of both approaches is the assumption of data corrupted by additive zero-mean Gaussian noise. The optimized 3-D nonlocal means filter proposed in[9] also addresses the Rice distribution, and it proposes to estimate the variance from the minimum distance between the high-pass components of noisy patches. This approach exploits the relation between the expectation of the squared ℓ^2-distance and the variance of the noise.[3]

In this work, we present an extension of the BM4D denoising algorithm to data corrupted by either Gaussian or Rician noise having spatially varying variance. Exploiting the sparsity of the representation of the group in transform domain, the noise variance is accurately estimated from the outcome of robust median operations applied to the spectrum coefficients. Subsequently, the estimate is used during the collaborative filtering and the aggregation to calibrate the amount of coefficients shrinkage and the adaptive weights, respectively. Experimental results on volumetric data from the BrainWeb database demonstrate the state-of-the-art denoising performance of the proposed algorithm. In particular, our filter outperforms the method proposed in,[9] which is currently, to the best of our knowledge, the best-performing denoising method for volumetric data corrupted by spatially varying noise.

The remainder of paper is organized as follows. In Section 2 we define the adopted observation models, for both Gaussian and Rician noisy observations. Section 3 is devoted to the formal description of the fundamental steps of the algorithm, together with the techniques used to estimate the variance of the noise of both distributions. The implementation of the spatially adaptive BM4D algorithm is then formalized in Section 4. The results of the experimental validation of the proposed method are reported in Section 5, and the final discussions and general conclusions are summarized in Section 6.

2. OBSERVATION MODELS

2.1 Gaussian-Distributed Noise

We consider the noisy volumetric Gaussian observation $z_{\mathcal{N}} : X \to \mathbb{R}$ as

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

where $x = (x_1, x_2, x_3)$ is a 3-D coordinate belonging to the domain $X \subset \mathbb{Z}^3$, y is the (unknown) original noise-free signal, and $\eta(x) \sim \mathcal{N}\left(0, \sigma^2(x)\right)$ is independent additive white Gaussian noise having spatially varying standard deviation $\sigma : X \to \mathbb{R}^+$.

2.2 Rician-Distributed Noise

The observation model of a Rician observation $z_{\mathcal{R}} : X \to \mathbb{R}^+$ is

$$z_{\mathcal{R}}(x) = \sqrt{\left(c_r y(x) + \sigma(x)\eta_r(x)\right)^2 + \left(c_i y(x) + \sigma(x)\eta_i(x)\right)^2}, \quad x \in X, \tag{2}$$

where $x = (x_1, x_2, x_3)$ is again a 3-D coordinate belonging to the domain $X \subset \mathbb{Z}^3$, c_r and c_i are constants such that $0 \leq c_r, c_i \leq 1 = c_r^2 + c_i^2$, 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, $z_{\mathcal{R}}(x) \sim \mathcal{R}\left(y(x), \sigma(x)\right)$ represents the raw magnitude MR data, modeled as a Rician distribution $\mathcal{R}$ of parameters y and $\sigma : X \to \mathbb{R}^+$, which denote the (unknown) original noise-free signal and the spatially varying standard deviation, respectively.

3. BASIC ALGORITHM

The aim of the proposed algorithm is to provide an estimate $\hat{y}$ of the original volumetric signal y from the observed data $z_{\mathcal{N}}$ or $z_{\mathcal{R}}$. The proposed adaptive BM4D algorithm comprises the grouping, collaborative filtering and aggregation step as in,[4] with an additional step performed after the grouping, devoted to the groupwise estimation of the noise variance.

3.1 Grouping

Let $C^z_{x_R}$ denote a cube of $L \times L \times L$ voxels, with $L \in \mathbb{N}$, extracted from the generic observation z at the 3-D coordinate $x_R \in X$, which identifies its top-left-front corner. The 4-D groups are formed by stacking together, along an additional fourth dimension, 3-D cubes similar to $C^z_{x_R}$. Specifically, the similarity between two cubes is measured via the squared ℓ^2-norm of the intensities difference of their voxels, normalized with respect to the size of the cube:

$$d\big(C^z_{x_i}, C^z_{x_j}\big) = \frac{\big|\big|C^z_{x_i} - C^z_{x_j}\big|\big|^2_2}{L^3}. \tag{3}$$

The set containing the indices of the cubes extracted from z that are similar to $C^z_{x_R}$ is defined as

$$S^z_{x_R} = \Big\{x_i \in X : d(C^z_{x_R}, C^z_{x_i}) \leq \tau_{\text{match}}\Big\}, \tag{4}$$

thus, two cubes are considered similar if their distance (3) is smaller or equal than a predefined threshold τ_{match}. The set $S^z_{x_R}$ is consequently used to build the group associated to the reference cube $C^z_{x_R}$ as the disjoint union of the matched cubes

$$\mathbf{G}^z_{S^z_{x_R}} = \coprod_{x_i \in S^z_{x_R}} C^z_{x_i}. \tag{5}$$

Observe that each set $\mathbf{G}^z_{x_R}$ necessarily contains the reference cube $C^z_{x_R}$ because $d(C^z_{x_R}, C^z_{x_R}) = 0$.

3.2 Groupwise Variance Estimation

We assume that the noise level in the groups (5) can be treated as constant. This is a reasonable assumption since the map σ is typically a slowly varying function, and the grouped cubes have usually nearby coordinates. Consequently, only a single standard deviation estimate $\hat{\sigma}_{x_R}$ is needed for each group. We remark that a precise estimation of the variance is a crucial step during the denoising, because the amount of filtering operated on the noisy observations is proportional to the strength of the corrupting noise.

The groups are sparsely represented in transform domain as the energies of the signal and the noise are well localized in the low- and high-frequencies portions of the spectrum, respectively. Thus, an accurate groupwise variance estimation can be obtained from the median of absolute deviation[10,11] (MAD) of the high-frequencies coefficients in the 4-D group spectrum.[12]

3.2.1 Gaussian-distributed data

In case of Gaussian-distributed data ($z \equiv z_{\mathcal{N}}$), we apply an orthonormal separable 4-D transform $\mathfrak{T}_{4D}$ to the group (5), obtaining

$$\Phi_{S^{z_{\mathcal{N}}}_{x_R}} = \mathcal{H}\left(\mathfrak{T}_{4D}\left(\mathbf{G}^{z_{\mathcal{N}}}_{S^{z_{\mathcal{N}}}_{x_R}}\right)\right), \tag{6}$$

where $\mathcal{H}$ is a high-pass filter that discards the DC hyperplane of the transform applied to the fourth dimension of the group.

A robust estimate $\hat{\sigma}_{x_R}$ of the standard deviation is consequently calculated as

$$\hat{\sigma}^{z_{\mathcal{N}}}_{x_R} = \frac{1}{0.6745} \cdot \text{MAD}\left(\Phi_{S^{z_{\mathcal{N}}}_{x_R}}\right) = \frac{1}{0.6745} \cdot \text{median}\left(\left\{\left|\phi_k - \text{median}\big(\Phi_{S^{z_{\mathcal{N}}}_{x_R}}\big)\right|\right\}\right), \ \phi_k \in \Phi_{S^{z_{\mathcal{N}}}_{x_R}}, \tag{7}$$

where ϕ_k is the k^{th} coefficient of the high-passed spectrum $\Phi_{S^{z_{\mathcal{N}}}_{x_R}}$. The orthonormality of $\mathfrak{T}_{4D}$ ensures that the noise standard deviation in transform and spatial domain coincide. Even though this would strictly require the independence of the data, i.e. non overlapping cubes,[12] we have experimentally found that the potential underestimation due to overlaps does not significantly affect the final denoising quality.

3.2.2 Rician-distributed data

If the data follows the Rician distribution ($z \equiv z_{\mathcal{R}}$), we first estimate the mean-variance pair (μ_{x_R}, $s^2_{x_R}$) of the median value of y over the Rician group $\mathbf{G}^{z_{\mathcal{R}}}_{S^{z_{\mathcal{R}}}_{x_R}}$ as

$$\hat{\mu}_{x_R} = \text{median}\left(\mathbf{G}^{z_{\mathcal{R}}}_{S^{z_{\mathcal{R}}}_{x_R}}\right), \tag{8}$$

$$\hat{s}_{x_R} = \frac{1}{0.6745} \cdot \text{MAD}\left(\Phi_{S^{z_{\mathcal{R}}}_{x_R}}\right), \tag{9}$$

where $\Phi_{S^{z_{\mathcal{R}}}_{x_R}}$ is the 4-D spectrum calculated as in (6). It can be shown that from the pair ($\hat{\mu}_{x_R}$, $\hat{s}^2_{x_R}$) one can univocally and directly obtain a robust estimate $\hat{\sigma}^{z_{\mathcal{R}}}_{x_R}$ of the parameter σ in (2).

3.3 Collaborative Filtering

The first phase of collaborative filtering, executed on Rician observations only, is the application of a variance stabilization transform (VST) specifically designed for the Rice distribution,[13] in order to remove the dependencies between the noise and the underlying grouped data. In this way, the stabilized data can be filtered using the constant standard deviation value $c > 0$ induced by the VST.

During collaborative filtering, each group is first transformed by a decorrelating separable four-dimensional transform $\mathcal{T}_{4D}$, then the coefficients of the so-obtained spectrum are thresholded through a generic shrinkage operator Υ (e.g., hard thresholding or Wiener filtering) parametrized by the estimated noise level s. The filtered group $\hat{\mathbf{G}}^{y}_{S^{z}_{x_R}}$ is eventually produced by inverting the original four-dimensional transform as

$$\mathcal{T}^{-1}_{4D}\left(\Upsilon_s\left(\mathcal{T}_{4D}\left(\mathbf{G}^{z}_{S^{z}_{x_R}}\right)\right)\right) = \hat{\mathbf{G}}^{y}_{S^{z}_{x_R}} = \coprod_{x_i \in S^{z}_{x_R}} \hat{C}^{y}_{x_i}, \tag{10}$$

where $\mathcal{T}_{4D}$ is the combination of four 1-D linear transform that are separately applied to each dimension of the group, and $s = \hat{\sigma}^{z_{\mathcal{N}}}_{x_R}$ or $s = c$ if the noise is Gaussian- or Rician-distributed, respectively. The shrinkage is never applied on the DC coefficient of the 4-D spectrum, in order to preserve the mean value of the group. Each $\hat{C}^{y}_{x_i}$ is an estimate of the original $C^{y}_{x_i}$ extracted from the unknown volumetric data y.

Finally, in case of Rician noise, the filtered group undergoes the exact unbiased inverse variance stabilization transform as in[13] that simultaneously inverts the VST and produces an unbiased estimate for the underlying y.

3.4 Aggregation

Since the cubes in the different group estimates $\hat{\mathbf{G}}^{y}_{S^{z}_{x_R}}$ (as well as the cubes within the same group) are likely to overlap, we may have multiple estimates for the same voxel. Therefore the final volumetric estimate $\hat{y}$ is obtained through a convex combination with adaptive weights formulated as

$$\hat{y} = \frac{\sum_{x_R \in X}\left(\sum_{x_i \in S^{z}_{x_R}} w_{x_R} \hat{C}^{y}_{x_i}\right)}{\sum_{x_R \in X}\left(\sum_{x_i \in S^{z}_{x_R}} w_{x_R} \chi_{x_i}\right)}, \tag{11}$$

where each cube estimate $\hat{C}^{y}_{x_i}$ is assumed to be zero-padded outside its domain, and $\chi_{x_i} : X \to \{0, 1\}$ denotes the characteristic function of the domain of a cube $\hat{C}^{y}_{x_i}$ located at x_i. In other words, $\chi_{x_i} = 1$ over the coordinates of the voxels of $\hat{C}^{y}_{x_i}$ and $\chi_{x_i} = 0$ elsewhere. The aggregation weights w_{x_R} are defined to be the reciprocal of the total residual noise variance in the estimate of the corresponding groups.

4. IMPLEMENTATION

The general procedure described in Section 3 is implemented in two cascading stages, each composed of the grouping, noise variance estimation, collaborative filtering and aggregation steps.

4.1 Hard-Thresholding Stage

In the first stage, the cubes are extracted from the generic observation z, and the group $\mathbf{G}^z_{S^z_{x_R}}$ is then formed testing the similarity measure (3) with a predefined threshold $\tau^{\text{ht}}_{\text{match}}$. After the standard deviation $\hat{\sigma}^{\text{ht}}_{x_R}$ of the noise is estimated from the group $\mathbf{G}^z_{S^z_{x_R}}$ as described in Section 3.2, collaborative filtering is realized by hard thresholding the coefficients of the spectra in (10) with an adaptive threshold value $\hat{\sigma}_{x_R} \cdot \lambda_{4D}$ in case of Gaussian noise, or $c \cdot \lambda_{4D}$ in case of Rician noise, being c the value of the stabilized standard deviation. In the latter case, the group undergoes a forward and inverse VST before and after the filtering (10), respectively.

The outcome of hard-thresholding stage, $\hat{y}^{\text{ht}}$, is obtained by aggregating the estimated cubes obtained obtained from collaborative filtering via the convex combination (11). The adaptive weights w_{x_R} in (11) are reciprocal to the residual noise variance in the estimate which, in case of hard thresholding is approximated with the number $N^{\text{ht}}_{x_R}$ of coefficients retained after thresholding times the estimated variance as

$$w^{\text{ht}}_{x_R} = \hat{\sigma}^{-2}_{x_R} {N^{\text{ht}}_{x_R}}^{-1}, \tag{12}$$

thus penalizing groups having higher estimated variance of the corrupting noise, as well as rewarding sparser groups. Note that $N^{\text{ht}}_{x_R} \geq 1$, since at least the DC coefficients is retained.

4.2 Wiener-Filtering Stage

In the Wiener-filtering stage, the grouping is performed within the hard-thresholding estimate $\hat{y}^{\text{ht}}$, thus for each reference cube $C^{\hat{y}^{\text{ht}}}_{x_R}$ with $x_R \in X$ we look for similar cubes in $\hat{y}^{\text{ht}}$ via (4) using a similarity threshold $\tau^{\text{wie}}_{\text{match}}$. Since the noise is considerably reduced after the first stage, the cube-matching in $\hat{y}^{\text{ht}}$ is far more accurate. The improved correlation properties of the group are consequently beneficial to collaborative filtering because they enable a better sparsification of the data in transform domain.

The set of coordinates $S^{\hat{y}^{\text{ht}}}_{x_R}$ is used to form two groups: one from the observation z, and the other from the basic estimate $\hat{y}^{\text{ht}}$, termed $\mathbf{G}^z_{S^{\hat{y}^{\text{ht}}}_{x_R}}$ and $\mathbf{G}^{\hat{y}^{\text{ht}}}_{S^{\hat{y}^{\text{ht}}}_{x_R}}$, respectively. The standard deviation $\hat{\sigma}_{x_R}$ of the noise is estimated from the noisy data grouped in $\mathbf{G}^z_{S^{\hat{y}^{\text{ht}}}_{x_R}}$, and collaborative filtering is consequently realized through an empirical Wiener filter. Element by element, the group spectrum is multiplied by the Wiener shrinkage coefficients, defined from the energy of the transformed spectrum of the basic estimate group as

$$\mathbf{W}_{S^{\hat{y}^{\text{ht}}}_{x_R}} = \frac{\left|\mathcal{T}^{\text{wie}}_{4D}\left(\mathbf{G}^{\hat{y}^{\text{ht}}}_{S^{\hat{y}^{\text{ht}}}_{x_R}}\right)\right|^2}{\left|\mathcal{T}^{\text{wie}}_{4D}\left(\mathbf{G}^{\hat{y}^{\text{ht}}}_{S^{\hat{y}^{\text{ht}}}_{x_R}}\right)\right|^2 + s^2}, \tag{13}$$

where $s = \hat{\sigma}_{x_R}$ or $s = c$ when the noise follows a Gaussian or Rician distribution, respectively. As usual, the Rician-distributed data is first stabilized by a VST that shall be eventually inverted after the filtering.

The final estimate $\hat{y}^{\text{wie}}$ is produced through (11) using aggregation weights defined as

$$w^{\text{wie}}_{x_R} = \hat{\sigma}^{-2}_{x_R} \left|\left|\mathbf{W}_{S^{\hat{y}^{\text{ht}}}_{x_R}}\right|\right|_2^{-2}, \tag{14}$$

which, similarly to (12), give an estimate of the total residual noise variance of the corresponding Wiener filtered group.[1]

5. EXPERIMENTS

We evaluate the denoising performances of the proposed algorithm, termed BM4D-AV, on magnetic resonance (MR) images. As quality measure, we compute the PSNR of the denoised data as

$$\text{PSNR}\left(y, \hat{y}\right) = 10 \log_{10}\left(\frac{D^2 |\tilde{X}|}{\sum_{x \in \tilde{X}} \left(\hat{y}(x) - y(x)\right)^2}\right), \tag{15}$$

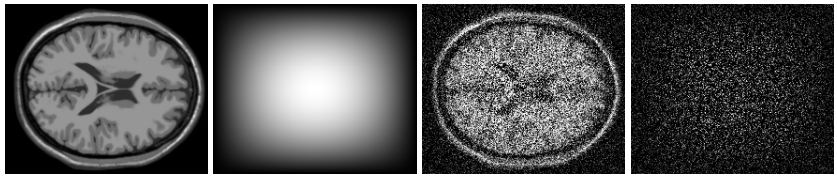


Figure 1. From left to right: original cross-section of the BrainWeb phantom; noise modulation map, with modulation factors ranging from 1 (black) to 3 (white); noisy BrainWeb phantom corrupted by Gaussian noise with standard deviation $\sigma \in [15\% \sim 45\%]$ varied with the modulation map; realization of the spatially varying Gaussian noise.

Table 1. Standard deviation values maximizing the PSNR denoising performance of the non-adaptive ODCT3D,[14] PRI-NLM3D,[14] and BM4D[4] filters applied to the BrainWeb phantom corrupted by spatially varying noise.

Noise	**Filter**	σ							
		$1\% \sim 3\%$	$3\% \sim 9\%$	$5\% \sim 15\%$	$7\% \sim 21\%$	$9\% \sim 27\%$	$11\% \sim 33\%$	$13\% \sim 39\%$	$15\% \sim 45\%$
Gauss.	ODCT3D$_{\mathcal{N}}$	2.33%	6.86%	11.34%	15.62%	19.91%	24.70%	29.01%	33.61%
	PRI-NLM3D$_{\mathcal{N}}$	2.35%	6.39%	10.90%	15.11%	19.90%	23.39%	28.47%	32.83%
	BM4D$_{\mathcal{N}}$	2.25%	6.98%	11.62%	16.80%	21.60%	27.22%	32.17%	38.25%
Rician	ODCT3D$_{\mathcal{R}}$	2.33%	5.96%	9.59%	12.64%	15.69%	18.88%	22.37%	25.27%
	PRI-NLM3D$_{\mathcal{R}}$	2.18%	5.96%	9.45%	12.64%	15.69%	19.22%	22.23%	25.49%
	BM4D$_{\mathcal{R}}$	2.25%	6.30%	9.37%	12.86%	15.52%	19.39%	22.42%	25.87%

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[14]), and $|\tilde{X}|$ is the cardinality of $\tilde{X}$. Additionally, we evaluate the results of denoising via the 3-D extension of the structure similarity index (SSIM),[14,15] which should better agree with subjective perceptual quality.

The observations used in our experiments are corrupted by either Gaussian or Rician noise, and the volumetric test data is the T1 brain phantom of size $181 \times 217 \times 181$ voxels from the BrainWeb database.[16] According to (1) and (2), we synthetically generate the noisy observations $z_{\mathcal{N}}$ and $z_{\mathcal{R}}$ by adding spatially varying Gaussian and Rician noise having different ranges of standard deviation σ, expressed as percentage of the maximum value of the signal y. Specifically, we first generate a realization of Gaussian or Rician noise with uniform standard deviation σ, then we multiply each sample of such realizations by a volumetric noise modulation map as in,[9] which smoothly increases the amount of noise from the extrema to the center of the volume up to a factor of 3. Figure 1 illustrates an example of noisy observation obtained by a modulated Gaussian noise with varying standard deviation $\sigma \in [15\% \sim 45\%]$.

As a comparison, we validate the denoising performances of the proposed BM4D algorithm against the optimized adaptive blockwise nonlocal means OB-AR-NLM3D-WM.[9] The BM4D is tuned as proposed in,[4] and the noise-variance estimation and the collaborative filtering steps use the same transform, i.e. $\mathcal{T}_{4D} \equiv \mathfrak{T}_{4D}$, both in the hard-thresholding and Wiener-filtering stages. In this way, the groups need to be transformed only once. We also present the performances of the current best-performing non-adaptive methods. In particular, we test the BM4D,[4] ODCT3D,[14] and PRI-NLM3D[14] filters, using constant standard deviation values found maximizing the restored quality in terms of PSNR. Table 1 reports the optimum values of standard deviation used by the three non-adaptive algorithms during the denoising of the BrainWeb phantom corrupted by spatially varying noise having eight different ranges of standard-deviation. Moreover, we present the results obtained by an *Oracle* filter, namely the state-of-the-art BM4D,[4] having exact knowledge on the varying standard deviation $\sigma(x)$ for each $x \in X$.

As one can clearly see, both the objective performances reported in Table 2 and the visual appearance of the denoised phantoms shown in Figure 2, substantiate the superior quality of the results produced by the proposed BM4D-AV. In particular, BM4D-AV outperforms the state-of-the-art adaptive filter OB-AR-NLM3D-WM and the non-adaptive state-of-the-art filter BM4D with PSNR improvements of up to 2.5dB and 0.5dB in case of Gaussian observations, and about 0.2dB in case of Rician observations. Let us remark that BM4D-AV performs

Table 2. PSNR (left value in each cell) and SSIM[15] (right value in each cell) denoising performances on the volumetric test data from the BrainWeb database[16] of the non-adaptive ODCT3D,[14] PRI-NLM3D,[14] BM4D[4] filters, the adaptive OB-AR-NLM3D-WM[9] filter, and the proposed adaptive BM4D-AV (tuned with the modified profile as in[4]). Two kinds of observations are tested, corrupted by spatially varying Gaussian and Rician noise, synthetically generated according to the observation models (1) and (2), respectively. Both cases are tested under different ranges of standard-deviations σ, expressed as percentage relative to the maximum intensity value of the original volumetric data. The ranges of variation for σ are shown in the header of the table. The PSNR and SSIM values of the noisy data, and of the denoised phantoms produced by an *Oracle*, namely BM4D with exact knowledge of the varying σ, are also shown for comparison. The subscripts $\mathcal{N}$ (Gaussian) and $\mathcal{R}$ (Rician) denote the addressed noise distribution.

Noise	**Filter**	σ							
		1% ∼ 3%	3% ∼ 9%	5% ∼ 15%	7% ∼ 21%	9% ∼ 27%	11% ∼ 33%	13% ∼ 39%	15% ∼ 45%
Gauss.	*Noisy data*	34.34|0.90	24.80|0.62	20.36|0.44	17.44|0.33	15.26|0.25	13.51|0.20	12.06|0.16	10.82|0.13
	ODCT3D$_{\mathcal{N}}$	40.04|**0.98**	34.09|0.94	31.43|0.90	29.69|0.86	28.42|0.83	27.40|0.80	26.52|0.77	25.74|0.74
	PRI-NLM3D$_{\mathcal{N}}$	**40.71**|**0.98**	34.50|0.94	31.75|0.91	29.95|0.87	28.60|0.83	27.49|0.80	26.55|0.77	25.79|0.74
	BM4D$_{\mathcal{N}}$	40.42|**0.98**	34.90|0.95	32.57|0.92	31.05|0.89	29.91|**0.87**	28.99|**0.85**	28.23|**0.83**	27.56|**0.81**
	OB-AR-NLM3D-WM$_{\mathcal{N}}$	40.38|**0.98**	34.50|0.94	31.57|0.89	29.61|0.83	28.11|0.78	26.89|0.73	25.86|0.68	24.95|0.64
	BM4D-AV$_{\mathcal{N}}$	40.45|**0.98**	**35.48**|**0.96**	**33.10**|**0.93**	**31.48**|**0.90**	**30.24**|**0.87**	**29.22**|**0.85**	**28.35**|0.82	**27.59**|0.79
	Oracle$_{\mathcal{N}}$	*40.96*|*0.98*	*35.56*|*0.96*	*33.14*|*0.93*	*31.56*|*0.91*	*30.36*|*0.88*	*29.40*|*0.86*	*28.58*|*0.84*	*27.87*|*0.82*
Rician	*Noisy data*	34.35|0.90	24.87|0.62	20.50|0.44	17.64|0.33	15.50|0.25	13.78|0.19	12.32|0.15	11.04|0.12
	ODCT3D$_{\mathcal{R}}$	39.70|**0.98**	33.13|0.92	29.58|0.86	26.92|0.79	24.72|**0.74**	22.85|**0.70**	21.12|**0.66**	19.47|**0.62**
	PRI-NLM3D$_{\mathcal{R}}$	**40.53**|**0.98**	33.21|0.93	29.29|0.87	26.21|0.80	23.71|**0.74**	21.68|0.69	19.88|0.65	18.21|0.61
	BM4D$_{\mathcal{R}}$	40.34|**0.98**	33.76|0.93	30.19|0.86	27.37|0.80	24.95|0.73	22.89|0.68	21.07|0.63	19.39|0.59
	OB-AR-NLM3D-WM$_{\mathcal{R}}$	40.28|**0.98**	34.29|**0.94**	31.16|0.87	28.73|0.81	26.43|**0.74**	24.17|0.67	22.00|0.60	20.00|0.54
	BM4D-AV$_{\mathcal{R}}$	40.43|**0.98**	**34.41**|**0.94**	**31.27**|**0.89**	**28.80**|**0.82**	**26.55**|**0.74**	**24.21**|0.67	**22.11**|0.61	**20.01**|0.56
	Oracle$_{\mathcal{R}}$	*40.90*|*0.98*	*34.85*|*0.95*	*31.59*|*0.91*	*28.99*|*0.85*	*26.82*|*0.76*	*24.55*|*0.70*	*22.43*|*0.65*	*20.37*|*0.61*

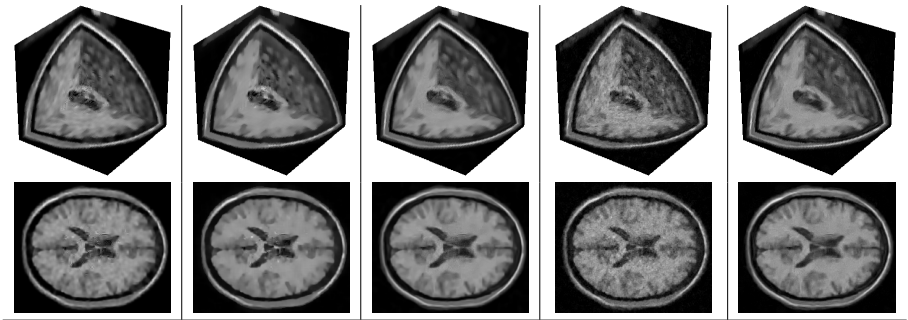


Figure 2. From left to right, denoising results of the ODCT3D, PRI-NLM3D, BM4D, OB-AR-NLM3D-WM, and the proposed BM4D-AV filter applied to the BrainWeb phantom corrupted by spatially varying Gaussian noise with standard deviation $\sigma \in [15\% \sim 45\%]$. The original and corrupted data can be seen in Figure 1. For each algorithm, both the 3-D and 2-D transversal cross-section of the phantom are presented in the top and bottom row, respectively.

only marginally worse than the *Oracle* filter, which, not surprisingly, always achieves the best performances. Figure 2 confirms the objective results. We observe that the denoised phantom produced by OB-AR-NLM3D-WM is considerably affected by residual noise, thus suggesting that the variance is underestimated during the filtering. The non-adaptive filters behave reasonably well, even though the effects of a fixed level of noise are clearly visible. In particular, the center of the phantom is under-smoothed as the applied amount of filtering is not sufficient to completely remove the noise. Conversely, the peripheral areas are over-smoothed.

6. CONCLUSIONS

Experiments show that the proposed adaptive BM4D-AV achieves state-of-the-art performances in volumetric data denoising under condition of spatially varying Gaussian- or Rician-distributed noise in terms of objective (Table 2) and subjective visual (Figure 2) quality. The groupwise noise estimation embedded in the proposed BM4D-AV allows for a correct filtering of the noisy data in any section of the phantom. As a matter of fact, our filter is able to simultaneously preserve the edges of fine details and the smoothness of flat areas. We also wish to remark that the proposed algorithm exhibits the most gentle performance decay as the level of noise increases. Thus, BM4D-AV can be a viable and effective tool in medical image processing when there is no precise knowledge about the statistics of the noise corrupting the observed data.

REFERENCES

[1] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing* **16**, 2080–2095 (August 2007).

[2] De Bonet, J. S., "Noise reduction through detection of signal redundancy," tech. rep., Rethinking Artificial Intelligence, MIT AI Lab (1997).

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

[4] Maggioni, M., Katkovnik, V., Egiazarian, K., and Foi, A., "A nonlocal transform-domain filter for volumetric data denoising and reconstruction," *submitted to IEEE Transactions on Image Processing* (2011).

[5] Pruessmann, K. P., Weiger, M., Scheidegger, M., and Boesiger, P., "SENSE: Sensitivity encoding for fast MRI," *Magnetic Resonance in Medicine* **42**, 952–962 (1999).

[6] Griswold, M., Jakob, P., Heidemann, R., Nittka, M., Jellus, V., Wang, J., Kiefer, B., and Haase, A., "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magnetic Resonance in Medicine* **47**(6), 1202–1210 (2002).

[7] Samsonov, A. and Johnson, C., "Noise-adaptive nonlinear diffusion filtering of MR images with spatially varying noise levels," *Magnetic Resonance in Medicine* **52**, 798–806 (October 2004).

[8] Delakis, I., Hammad, O., and Kitney, R. I., "Wavelet-based de-noising algorithm for images acquired with parallel magnetic resonance imaging (MRI)," *Physics in Medicine and Biology* **52**(13), 3741 (2007).

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

[10] Hampel, F., "The influence curve and its role in robust estimation," *Journal of the American Statistical Association* **69**, 383–393 (June 1974).

[11] Donoho, D., Johnstone, I., and Johnstone, I., "Ideal spatial adaptation by wavelet shrinkage," *Biometrika* **81**(3), 425–455 (1993).

[12] Danielyan, A. and Foi, A., "Noise variance estimation in nonlocal transform domain," in [*International Workshop on Local and Non-Local Approximation in Image Processing (LNLA)*], 41–45 (August 2009).

[13] Foi, A., "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*], (2011).

[14] Manjón, J. V., Coupé, P., Buades, A., Collins, D. L., and Robles, M., "New methods for MRI denoising based on sparseness and self-similarity," *Medical Image Analysis (in press)* (2011).

[15] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E., "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**, 600–612 (April 2004).

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

Publication III

M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms. *IEEE Transactions on Image Processing*, 21(9):3952–3966, Sep. 2012

Video Denoising, Deblocking and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms

Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, Karen Egiazarian

***Abstract*—We propose a powerful video filtering algorithm that exploits temporal and spatial redundancy characterizing natural video sequences. The algorithm implements the paradigm of nonlocal grouping and collaborative filtering, where a higher-dimensional transform-domain representation of the observations is leveraged to enforce sparsity and thus regularize the data: 3-D spatiotemporal volumes are constructed by tracking blocks along trajectories defined by the motion vectors. Mutually similar volumes are then grouped together by stacking them along an additional fourth dimension, thus producing a 4-D structure, termed group, where different types of data correlation exist along the different dimensions: local correlation along the two dimensions of the blocks, temporal correlation along the motion trajectories, and nonlocal spatial correlation (i.e. self-similarity) along the fourth dimension of the group. Collaborative filtering is then realized by transforming each group through a decorrelating 4-D separable transform and then by shrinkage and inverse transformation. In this way, the collaborative filtering provides estimates for each volume stacked in the group, which are then returned and adaptively aggregated to their original positions in the video. The proposed filtering procedure addresses several video processing applications, such as denoising, deblocking, and enhancement of both grayscale and color data. Experimental results prove the effectiveness of our method in terms of both subjective and objective visual quality, and shows that it outperforms the state of the art in video denoising.**

***Index Terms*—Video filtering, video denoising, video deblocking, video enhancement, nonlocal methods, adaptive transforms, motion estimation.**

I. Introduction

SEVERAL factors such as noise, blur, blocking, ringing, and other acquisition or compression artifacts, typically impair digital video sequences. The large number of practical applications involving digital videos has motivated a significant interest in restoration or enhancement solutions, and the literature contains a plethora of such algorithms (see [3], [4] for a comprehensive overview).

At the moment, the most effective approach in restoring images or video sequences exploits the redundancy given by the *nonlocal* similarity between patches at different locations within the data [5], [6]. Algorithms based on this approach have been proposed for various signal-processing problems, and mainly for image denoising [4], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15]. Specifically, in [7] has been introduced an adaptive pointwise image filtering strategy, called *nonlocal means*, where the estimate of each pixel $\mathbf{x}_i$ is obtained as a weighted average of, in principle, all the pixels $\mathbf{x}_j$ of the noisy image, using a family of weights proportional to the similarity between two neighborhoods centered at $\mathbf{x}_i$ and $\mathbf{x}_j$. So far, the most effective image-denoising algorithm is BM3D [10], [6], which relies on the so-called grouping and collaborative filtering paradigm: the observation is processed in a blockwise manner and mutually similar 2-D image blocks are stacked into a 3-D group (grouping), which is then filtered through a transform-domain shrinkage (collaborative filtering), simultaneously providing different estimates for each grouped block. These estimates are then returned to their respective locations and eventually aggregated resulting in the denoised image. In doing so, BM3D leverages the spatial correlation of natural images both at the nonlocal and local level, due to the abundance of mutually similar patches and to the high correlation of image data within each patch, respectively. The BM3D filtering scheme has been successfully applied to video denoising in our previous work, V-BM3D [11], as well as to several other applications including image and video super-resolution [14], [15], [16], image sharpening [13], and image deblurring [17].

In V-BM3D, groups are 3-D arrays of mutually similar blocks extracted from a set of consecutive frames of the video sequence. A group may include multiple blocks from the same frame, naturally exploiting in this way the nonlocal similarity characterizing images. However, it is typically along the temporal dimension that most mutually similar blocks can be found. It is well known that motion-compensated videos [18] are extremely smooth along the temporal axis and this fact is exploited by nearly all modern video-coding techniques. Furthermore, experimental analysis in [12] shows that, even when fast motion is present, the similarity along the motion trajectories is much stronger than the nonlocal similarity existing within an individual frame. In spite of this, in V-BM3D the blocks are grouped regardless of whether their similarity comes from the motion tracking over time or the nonlocal spatial content. Consequently, during the filtering, V-BM3D is not able to distinguish between temporal and spatial nonlocal similarity. We recognize this as a conceptual as well

Matteo Maggioni, Alessandro Foi and Karen Egiazarian are with the Department of Signal Processing, Tampere University of Technology, Finland. Giacomo Boracchi is with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

This paper is based on and extends the authors' preliminary conference publications [1], [2]

This work was supported by the Academy of Finland (project no. 213462, Finnish Programme for Centres of Excellence in Research 20062011, project no. 252547, Academy Research Fellow 20112016, and project no. 129118, Postdoctoral Researchers Project 20092011), and by Tampere Graduate School in Information Science and Engineering (TISE).

as practical weakness of the algorithm. As a matter of fact, the simple experiments reported in Section VIII demonstrate that the denoising quality do not necessarily increase with the number of spatially self-similar blocks in each group; in contrast, the performances are always improved by exploiting the temporal correlation of the video.

This work proposes V-BM4D, a novel video-filtering approach that, to overcome the above weaknesses, separately exploits the temporal and spatial redundancy of the video sequences. The core element of V-BM4D is the spatiotemporal volume, a 3-D structure formed by a sequence of blocks of the video following a specific trajectory (obtained, for example, by concatenating motion vectors along time) [19], [20]. Thus, contrary to V-BM3D, V-BM4D does not group blocks, but mutually similar spatiotemporal volumes according to a nonlocal search procedure. Hence, groups in V-BM4D are 4-D stacks of 3-D volumes, and the collaborative filtering is then performed via a separable 4-D spatiotemporal transform. The transform leverages the following three types of correlation that characterize natural video sequences: local spatial correlation between pixels in each block of a volume, local temporal correlation between blocks of each volume, and nonlocal spatial and temporal correlation between volumes of the same group. The 4-D group spectrum is thus highly sparse, which makes the shrinkage more effective than in V-BM3D, yielding superior performance of V-BM4D in terms of noise reduction.

In this work we extend the basic implementation of V-BM4D as a grayscale denoising filter introduced in the conference paper [1] presenting its modifications for the deblocking and deringing of compressed videos, as well as for the enhancement (sharpening) of low-contrast videos. Then, leveraging the approach presented in [10], [21], we generalize V-BM4D to perform collaborative filtering of color (multichannel) data. An additional, and fundamental, contribution of this paper is an experimental analysis of the different types of correlation characterizing video data, and how these affect the filtering quality.

The paper is organized as follows. Section II introduces the observation model, the formal definitions, and describes the fundamental steps of V-BM4D, while Section III discusses the implementation aspects, with particular emphasis on the computation of motion vectors. The application of V-BM4D to deblocking and deringing is given in Section IV, where it is shown how to compute the thresholds used in the filtering from the compression parameters of a video; video enhancement (sharpening) is presented in Section V. Before the conclusions, we provide a comprehensive collection of experiments and a discussion of the V-BM4D performance in Section VI, and a detailed analysis of its computational complexity in Section VII.

II. Basic Algorithm

The aim of the proposed algorithm is to provide an estimate of the original video from the observed data. For the algorithm design, we assume the common additive white Gaussian noise model.

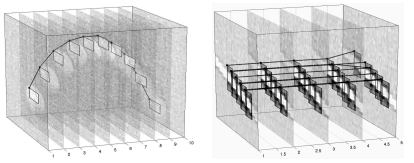


Fig. 1. Illustration of a trajectory and the associated volume (left), and a group of mutually similar volumes (right). These have been calculated from the sequence *Tennis* corrupted by white Gaussian noise with $\sigma = 20$.

A. Observation Model

We consider the observed video as a noisy image sequence $z : X \times T \to \mathbb{R}$ defined as

$$z(\mathbf{x}, t) = y(\mathbf{x}, t) + \eta(\mathbf{x}, t), \qquad \mathbf{x} \in X,\, t \in T, \tag{1}$$

where y is the original (unknown) video, $\eta(\cdot,\cdot) \sim \mathcal{N}(0, \sigma^2)$ is i.i.d. white Gaussian noise, and $(\mathbf{x}, t)$ are the 3-D spatiotemporal coordinates belonging to the spatial domain $X \subset \mathbb{Z}^2$ and time domain $T \subset \mathbb{Z}$, respectively. The frame of the video z at time t is denoted by $z(X, t)$.

The V-BM4D algorithm comprises three fundamental steps inherited from the BM3D paradigm, specifically grouping (Section II-C), collaborative filtering (Section II-D) and aggregation (Section II-E). These steps are performed for each spatiotemporal volume of the video (Section II-B).

B. Spatiotemporal Volumes

Let $B_z(\mathbf{x}_0, t_0)$ denote a square block of fixed size $N \times N$ extracted from the noisy video z; without loss of generality, the coordinates $(\mathbf{x}_0, t_0)$ identify the top-left pixel of the block in the frame $z(X, t_0)$. A spatiotemporal volume is a 3-D sequence of blocks built following a specific trajectory along time, which is supposed to follow the motion in the scene. Formally, the trajectory associated to $(\mathbf{x}_0, t_0)$ is defined as

$$\text{Traj}(\mathbf{x}_0, t_0) = \Big\{(\mathbf{x}_j, t_0 + j)\Big\}_{j=-h^-}^{h^+}, \tag{2}$$

where the elements $(\mathbf{x}_j, t_0 + j)$ are time-consecutive coordinates, each of these represents the position of the reference block $B_z(\mathbf{x}_0, t_0)$ within the neighboring frames $z(X, t_0 + j)$, $j = -h^-, \ldots, h^+$. For the sake of simplicity, in this section it is assumed $h^- = h^+ = h$ for all $(\mathbf{x}, t) \in X \times T$.

The trajectories can be either directly computed from the noisy video, or, when a coded video is given, they can be obtained by concatenating motion vectors. In what follows we assume that, for each $(\mathbf{x}_0, t_0) \in X \times T$, a trajectory $\text{Traj}(\mathbf{x}_0, t_0)$ is given and thus the 3-D spatiotemporal volume associated to $(\mathbf{x}_0, t_0)$ can be determined as

$$V_z(\mathbf{x}_0, t_0) = \big\{B_z(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \text{Traj}(\mathbf{x}_0, t_0)\big\}, \tag{3}$$

where the subscript z specifies that the volumes are extracted from the noisy video.

C. Grouping

Groups are stacks of mutually similar volumes and constitute the nonlocal element of V-BM4D. Mutually similar volumes are determined by a nonlocal search procedure as in [10].

Specifically, let $\text{Ind}(\mathbf{x}_0, t_0)$ be the set of indices identifying those volumes that, according to a distance operator δ^{v}, are similar to $V_z(\mathbf{x}_0, t_0)$:

$$\text{Ind}(\mathbf{x}_0, t_0) = \big\{(\mathbf{x}_i, t_i) : \delta^{\text{v}}(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)) < \tau_{\text{match}}\big\}.$$

The parameter $\tau_{\text{match}} > 0$ controls the minimum degree of similarity among volumes with respect to the distance δ^{v}, which is typically the ℓ^2-norm of the difference between two volumes.

The group associated to the reference volume $V_z(\mathbf{x}_0, t_0)$ is then

$$G_z(\mathbf{x}_0, t_0) = \big\{V_z(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \text{Ind}(\mathbf{x}_0, t_0)\big\}. \quad (4)$$

In (4) we implicitly assume that the 3-D volumes are stacked along a fourth dimension; hence the groups are 4-D data structures. The order of the spatiotemporal volumes in the 4-D stacks is based on their similarity with the reference volume. Note that since $\delta^{\text{v}}(V_z, V_z) = 0$, every group $G_z(\mathbf{x}_0, t_0)$ contains, at least, its reference volume $V_z(\mathbf{x}_0, t_0)$. Figure 1 shows an example of trajectories and volumes belonging to a group.

D. Collaborative Filtering

According to the general formulation of the grouping and collaborative-filtering approach for a d-dimensional signal [10], groups are $(d+1)$-dimensional structures of similar d-dimensional elements, which are then jointly filtered. In particular, each of the grouped elements influences the filtered output of all the other elements of the group: this is the basic idea of collaborative filtering. It is typically realized through the following steps: firstly a $(d+1)$-dimensional separable linear transform is applied to the group, then the transformed coefficients are shrunk, for example by hard thresholding or by Wiener filtering, and finally the $(d+1)$-dimensional transform is inverted to obtain an estimate for each grouped element.

The core elements of V-BM4D are the spatiotemporal volumes ($d = 3$), and thus the collaborative filtering performs a 4-D separable linear transform $\mathcal{T}_{4D}$ on each 4-D group $G_z(\mathbf{x}_0, t_0)$, and provides an estimate for each grouped volume V_z:

$$\hat{G}_y(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{-1}\big(\Upsilon\left(\mathcal{T}_{4D}\left(G_z(\mathbf{x}_0, t_0)\right)\right)\big),$$

where Υ denotes a generic shrinkage operator. The filtered 4-D group $\hat{G}_y(\mathbf{x}_0, t_0)$ is composed of volumes $\hat{V}_y(\mathbf{x}, t)$

$$\hat{G}_y(\mathbf{x}_0, t_0) = \big\{\hat{V}_y(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \text{Ind}(\mathbf{x}_0, t_0)\big\},$$

with each $\hat{V}_y$ being an estimate of the corresponding unknown volume V_y in the original video y.

E. Aggregation

The groups $\hat{G}_y$ constitute a very redundant representation of the video, because in general the volumes $\hat{V}_y$ overlap and, within the overlapping parts, the collaborative filtering provides multiple estimates at the same coordinates $(\mathbf{x}, t)$. For this reason, the estimates are aggregated through a convex combination with adaptive weights. In particular, the estimate $\hat{y}$ of the original video is computed as

$$\hat{y} = \frac{\sum_{(\mathbf{x}_0,t_0)\in X\times T}\big(\sum_{(\mathbf{x}_i,t_i)\in \text{Ind}(\mathbf{x}_0,t_0)} w_{(\mathbf{x}_0,t_0)}\hat{V}_y(\mathbf{x}_i,t_i)\big)}{\sum_{(\mathbf{x}_0,t_0)\in X\times T}\big(\sum_{(\mathbf{x}_i,t_i)\in \text{Ind}(\mathbf{x}_0,t_0)} w_{(\mathbf{x}_0,t_0)}\chi_{(\mathbf{x}_i,t_i)}\big)}, \quad (5)$$

where we assume $\hat{V}_y(\mathbf{x}_i, t_i)$ to be zero-padded outside its domain, $\chi_{(\mathbf{x}_i,t_i)} : X \times T \to \{0, 1\}$ is the characteristic function (indicator) of the support of the volume $\hat{V}_y(\mathbf{x}_i, t_i)$, and the aggregation weights $w_{(\mathbf{x}_0,t_0)}$ are different for different groups. Aggregation weights may depend on the result of the shrinkage in the collaborative filtering, and these are typically defined to be inversely proportional to the total sample variance of the estimate of the corresponding groups [10]. Intuitively, the sparser is the shrunk 4-D spectrum $\hat{G}_y(\mathbf{x}_0, t_0)$, the larger is the corresponding weight $w_{(\mathbf{x}_0,t_0)}$. Such aggregation is a well-established procedure to obtain a global estimate from different overlapping local estimates [22], [23].

III. Implementation Aspects

A. Computation of the Trajectories

In our implementation of V-BM4D, we construct trajectories by concatenating motion vectors which are defined as follows.

1) Location prediction: As far as two consecutive spatiotemporal locations $(\mathbf{x}_{i-1}, t_i - 1)$ and $(\mathbf{x}_i, t_i)$ of a block are known, we can define the corresponding motion vector (velocity) as $\mathbf{v}(\mathbf{x}_i, t_i) = \mathbf{x}_{i-1} - \mathbf{x}_i$. Hence, under the assumption of smooth motion, we can predict the position $\hat{\mathbf{x}}_i(t_i + 1)$ of the block in the frame $z\,(X, t_i + 1)$ as

$$\hat{\mathbf{x}}_i(t_i + 1) = \mathbf{x}_i + \gamma_p \cdot \mathbf{v}(\mathbf{x}_i, t_i), \quad (6)$$

where $\gamma_p \in [0, 1]$ is a weighting factor of the prediction. In the case $(\mathbf{x}_{i-1}, t_i - 1)$ is not available, we consider the lack of motion as the most likely situation and we set $\hat{\mathbf{x}}_i(t_i + 1) = \mathbf{x}_i$. Analogous predictions can be made when looking for precedent blocks in the sequence.

2) Similarity criterion: The motion of a block is generally tracked by identifying the most similar block in the subsequent or precedent frame. However, since we deal with noisy signals, it is advisable to enforce motion-smoothness priors to improve the tracking. In particular, given the predicted future $\hat{\mathbf{x}}_i(t_i+1)$ or past $\hat{\mathbf{x}}_i(t_i - 1)$ positions of the block $B_z(\mathbf{x}_i, t_i)$, we define the similarity between $B_z(\mathbf{x}_i, t_i)$ and $B_z(\mathbf{x}_j, t_i \pm 1)$, through a penalized quadratic difference

$$\delta^{\text{b}}\big(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_j, t_i \pm 1)\big) = \frac{||B_z(\mathbf{x}_i, t_i) - B_z(\mathbf{x}_j, t_i \pm 1)||_2^2}{N^2} + \gamma_d \left|\left|\hat{\mathbf{x}}_i(t_i \pm 1) - \mathbf{x}_j\right|\right|_2, \quad (7)$$

where $\hat{\mathbf{x}}_i(t_i \pm 1)$ is defined as in (6), and $\gamma_d \in \mathbb{R}^+$ is the penalization parameter. Observe that the tracking is performed separately in time $t_i + 1$ and $t_i - 1$.

V-BM4D constructs the trajectory (2) by repeatedly minimizing (7). Formally, the motion of $B_z(\mathbf{x}_i, t_i)$ from time t_i to $t_i \pm 1$ is determined by the position $\mathbf{x}_{i\pm1}$ that minimizes (7) as

$$\mathbf{x}_{i\pm1} = \underset{\mathbf{x}_k \in \mathcal{N}_i}{\arg\min} \Big\{\delta^{\text{b}}\big(B_z(\mathbf{x}_i, t_i), B_z(\mathbf{x}_k, t_i \pm 1)\big)\Big\},$$

where $\mathcal{N}_i$ is an adaptive spatial search neighborhood in the frame $z(X, t_i \pm 1)$ (further details are given in Section III-A3). Even though such $\mathbf{x}_{i\pm1}$ can be always found, we stop the trajectory construction whenever the corresponding minimum distance δ^{b} exceeds a fixed parameter $\tau_{\text{traj}} \in \mathbb{R}^+$, which imposes a minimum amount of similarity along the spatiotemporal volumes. This allows V-BM4D to effectively

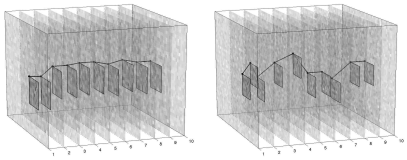


Fig. 2. Effect of different penalties $\gamma_d = 0.025$ (left) and $\gamma_d = 0$ (right) on the background textures of the sequence *Tennis* corrupted by Gaussian noise with $\sigma = 20$. The block positions at time $t = 1$ are the same in both experiments.

deal with those situations, such as occlusions and changes of scene, where consistent blocks (in terms of both similarity and motion smoothness) cannot be found.

Figure 2 illustrates two trajectories estimated using different penalization parameters γ_d. Observe that the penalization term becomes essential when blocks are tracked within flat areas or homogeneous textures in the scene. In fact, the right image of Figure 2 shows that without a position-dependent distance metric the trajectories would be mainly determined by the noise. As a consequence, the collaborative filtering would be less effective because of the badly conditioned temporal correlation of the data within the volumes.

3) Search neighborhood: Because of the penalty term $\gamma_d \left|\left|\hat{\mathbf{x}}_i(t_i \pm 1) - \mathbf{x}_j\right|\right|_2$, the minimizer of (7) is likely close to $\hat{\mathbf{x}}_i(t_i \pm 1)$. Thus, we can rightly restrict the minimization of (7) to a spatial search neighborhood $\mathcal{N}_i$ centered at $\hat{\mathbf{x}}_i(t_i \pm 1)$. We experienced that it is convenient to make the search-neighbor size, $N_{PR} \times N_{PR}$, adaptive on the velocity of the tracked block (magnitude of motion vector) by setting

$$N_{PR} = N_S \cdot \left(1 - \gamma_w \cdot e^{-\frac{||\mathbf{v}(\mathbf{x}_i, t_i)||_2^2}{2 \cdot \sigma_w^2}}\right),$$

where N_S is the maximum size of $\mathcal{N}_i$, $\gamma_w \in [0, 1]$ is a scaling factor and $\sigma_w > 0$ is a tuning parameter. As the velocity $\mathbf{v}$ increases, N_{PR} approaches N_S accordingly to σ_w; conversely, when the velocity is zero $N_{PR} = N_S(1 - \gamma_w)$. By setting a proper value of σ_w we can control the decay rate of the exponential term as a function of $\mathbf{v}$ or, in other words, how permissive is the window contraction with respect to the velocity of the tracked block.

B. Sub-volume Extraction

So far, the number of frames spanned by all the trajectories has been assumed fixed and equal to h. However, because of occlusions, scene changes or heavy noise, any trajectory $\text{Traj}(\mathbf{x}_i, t_i)$ can be interrupted at any time, i.e. whenever the distance between consecutive blocks falls below the threshold τ_{traj}. Thus, given a temporal extent $\left[t_i - h_i^-, t_i + h_i^+\right]$ for the trajectory $\text{Traj}(\mathbf{x}_i, t_i)$, we have that in general $0 \leq h_i^- \leq h$ and $0 \leq h_i^+ \leq h$, where h denotes the maximum forward and backward extent of the trajectories (thus of volumes) allowed in the algorithm.

As a result, in principle, V-BM4D may stack together volumes having different lengths. However, in practice, because of the separability of the transform $\mathcal{T}_{4D}$, every group $G_z(\mathbf{x}_i, t_i)$ has to be composed of volumes having the same length. Thus, for each reference volume $V_z(\mathbf{x}_0, t_0)$, we only consider the volumes $V_z(\mathbf{x}_i, t_i)$ such that $t_i = t_0$, $h_i^- \geq h_0^-$ and $h_i^+ \geq h_0^+$. Then, we extract from each $V_z(\mathbf{x}_i, t_i)$ the sub-volume having temporal extent $[t_0 - h_0^-, t_0 + h_0^+]$, denoted as $\mathcal{E}_{L_0}\big(V_z(\mathbf{x}_i, t_i)\big)$. Among all the possible criteria for extracting a sub-volume of length $L_0 = h_0^- + h_0^+ + 1$ from a longer volume, our choice aims at limiting the complexity while maintaining a high correlation within the grouped volumes, because we can reasonably assume that similar objects at different positions are represented by similar volumes along time.

In the grouping, we set as distance operator δ^{v} the ℓ^2-norm of the difference between time-synchronous volumes normalized with respect to their lengths:

$$\delta^{\text{v}}\big(V_z(\mathbf{x}_0, t_0), V_z(\mathbf{x}_i, t_i)\big) = \frac{\left|\left|V_z(\mathbf{x}_0, t_0) - \mathcal{E}_{L_0}\big(V_z(\mathbf{x}_i, t_i)\big)\right|\right|_2^2}{L_0}. \tag{8}$$

C. Two-Stage Implementation with Collaborative Wiener Filtering

The general procedure described in Section II is implemented in two cascading stages, each composed of the grouping, collaborative filtering and aggregation steps.

1) Hard-thresholding stage: In the first stage, volumes are extracted from the noisy video z, and groups are then formed using the δ^{v}-operator (8) and the predefined threshold $\tau_{\text{match}}^{\text{ht}}$. Collaborative filtering is realized by hard thresholding each group $G_z(\mathbf{x}, t)$ in 4-D transform domain:

$$\hat{G}_y^{\text{ht}}(\mathbf{x}, t) = \mathcal{T}_{4D}^{\text{ht}^{-1}}\left(\Upsilon^{\text{ht}}\left(\mathcal{T}_{4D}^{\text{ht}}\left(G_z(\mathbf{x}_0, t_0)\right)\right)\right), \ (\mathbf{x}, t) \in X \times T,$$

where $\mathcal{T}_{4D}^{\text{ht}}$ is the 4-D transform and Υ^{ht} is the hard-threshold operator with threshold $\sigma\lambda_{4D}$.

The outcome of the hard-thresholding stage, $\hat{y}^{\text{ht}}$, is obtained by aggregating with a convex combination all the estimated groups $\hat{G}_y^{\text{ht}}(\mathbf{x}, t)$, as defined in (5). The adaptive weights used in this combination are inversely proportional to the number $N_{(\mathbf{x}_0, t_0)}^{\text{ht}}$ of non-zero coefficients of the corresponding hard-thresholded group $\hat{G}_y^{\text{ht}}(\mathbf{x}_0, t_0)$: that is $w_{(\mathbf{x}_0, t_0)}^{\text{ht}} = 1/N_{(\mathbf{x}_0, t_0)}^{\text{ht}}$, which provides an estimate of the total variance of $\hat{G}_y^{\text{ht}}(\mathbf{x}, t)$. In such a way, we assign larger weights to the volumes belonging to groups having sparser representation in $\mathcal{T}_{4D}$ domain.

2) Wiener-filtering stage: In the second stage, the motion estimation is improved by extracting new trajectories $\text{Traj}_{\hat{y}^{\text{ht}}}$ from the basic estimate $\hat{y}^{\text{ht}}$, and the grouping is performed on the new volumes $V_{\hat{y}^{\text{ht}}}$. Volume matching is still performed through the δ^{v}-distance, but using a different threshold $\tau_{\text{match}}^{\text{wie}}$. The indices identifying similar volumes $\text{Ind}_{\hat{y}^{\text{ht}}}(\mathbf{x}, t)$ are used to construct both groups G_z and $G_{\hat{y}^{\text{ht}}}$, composed by volumes extracted from the noisy video z and from the estimate y^{ht}, respectively.

Collaborative filtering is hence performed using an empirical Wiener filter in $\mathcal{T}_{4D}^{\text{wie}}$ transform domain. Shrinkage is realized by scaling the 4-D transform coefficients of each group $G_z(\mathbf{x}_0, t_0)$, extracted from the noisy video z, with the Wiener attenuation coefficients $\mathbf{W}(\mathbf{x}_0, t_0)$,

$$\mathbf{W}(\mathbf{x}_0, t_0) = \frac{\left|\mathcal{T}_{4D}^{\text{wie}}\left(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)\right)\right|^2}{\left|\mathcal{T}_{4D}^{\text{wie}}\left(G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)\right)\right|^2 + \sigma^2},$$

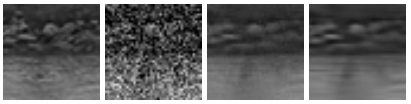


Fig. 3. V-BM4D two stage denoising of the sequence *Coastguard*. From left to right: original video y, noisy video z ($\sigma = 40$), result of the first stage y^{ht} (frame PSNR 28.58 dB) and final estimate y^{wie} (frame PSNR 29.38 dB).

that are computed from the energy of the 4-D spectrum of the group $G_{\hat{y}^{\text{ht}}}(\mathbf{x}_0, t_0)$. Eventually, the group estimate is obtained by inverting the 4-D transform as

$$\hat{G}_y^{\text{wie}}(\mathbf{x}_0, t_0) = \mathcal{T}_{4D}^{\text{wie}^{-1}}\left(\mathbf{W}(\mathbf{x}_0, t_0) \cdot \mathcal{T}_{4D}^{\text{wie}}\left(G_z(\mathbf{x}_0, t_0)\right)\right),$$

where $\cdot$ denotes the element-wise product. The final global estimate $\hat{y}^{\text{wie}}$ is computed by the aggregation (5), using the weights $w^{\text{wie}}_{(\mathbf{x}_0,t_0)} = ||\mathbf{W}(\mathbf{x}_0, t_0)||_2^{-2}$, which follow from considerations similar to those underlying the adaptive weights used in the first stage.

D. Settings

The parameters involved in the motion estimation and in the grouping, that is γ_d, τ_{traj} and τ_{match}, depend on the noise standard deviation σ. Intuitively, in order to compensate the effects of the noise, the larger is σ, the larger become the thresholds controlling blocks and volumes matching. For the sake of simplicity we model such dependencies as second-order polynomials in σ: $\gamma_d(\sigma)$, $\tau_{\text{traj}}(\sigma)$ and $\tau_{\text{match}}(\sigma)$. The nine coefficients required to describe the three polynomials are jointly optimized using the Nelder-Mead simplex direct search algorithm [24], [25]. As optimization criterion, we maximize the sum of the restoration performance (PSNR) of V-BM4D applied over a collection of test videos corrupted by synthetic noise having different values of σ. Namely, we considered *Salesman*, *Tennis*, *Flower Garden*, *Miss America*, *Coastguard*, *Foreman*, *Bus*, and *Bicycle* corrupted by white Gaussian noise having σ levels ranging from 5 and 70. The resulting polynomials are

$$\gamma_d(\sigma) = 0.0005 \cdot \sigma^2 - 0.0059 \cdot \sigma + 0.0400, \quad (9)$$

$$\tau_{\text{traj}}(\sigma) = 0.0047 \cdot \sigma^2 + 0.0676 \cdot \sigma + 0.4564, \quad (10)$$

$$\tau_{\text{match}}(\sigma) = 0.0171 \cdot \sigma^2 + 0.4520 \cdot \sigma + 47.9294. \quad (11)$$

The solid lines in Figure 4 show the above functions. We also plot, using different markers, the best values of the three parameters obtained by unconstrained and independent optimizations of V-BM4D for each test video and value of σ. Empirically, the polynomials demonstrate a good approximation of the optimum (γ_d, τ_{traj}, τ_{match}). Within the considered σ range, the curve (9) is "practically" monotone increasing despite its negative first-degree coefficient. We refrain from introducing additional constraints to the polynomials as well as from considering additional σ values smaller than 5, because the resulting sequences would be mostly affected by the noise and quantization artifacts intrinsic in the original test-data.

During the second stage (namely, the Wiener filtering) the γ_d, τ_{traj} and τ_{match} parameters can be considered as constants and independent, because in the processed sequence $\hat{y}^{\text{ht}}$ the noise has been considerably reduced with respect to the observation z; this is evident when looking at the second and third image of Figure 3. Moreover, since in this stage both the trajectories and groups are determined from the basic estimate $\hat{y}^{\text{ht}}$, there is no a straightforward relation with σ, the noise standard deviation in z.

IV. Deblocking

Most video compression techniques, such as MPEG-4 [26] or H.264 [27], make use of block-transform coding and thus may suffer, especially at low bitrates, from several compression artifacts such as blocking, ringing, mosquito noise, and flickering. These artifacts are mainly due to the coarse quantization of the block-transform coefficients and to the motion compensation. Moreover, since each block is processed separately, the correlation between pixels at the borders of neighboring blocks is typically lost during the compression, resulting in false discontinuities in the decoded video (such as those shown in the blocky frames in Figure 8).

A large number of deblocking filters have been proposed in the last decade; among them we mention frame-based enhancement using a linear low-pass filter in spatial or transform domain [28], projection onto convex sets (POCS) methods [29], spatial block boundary filter [30], statistical modeling methods [31] or shifted thresholding [32]. Additionally, most of modern video coding block-based techniques, such as H.264 or MPEG-4, embed an in-loop deblocking filter as an additional processing step in the decoder [26].

Inspired by [33], we treat the blocking artifacts as additive noise. This choice allows us to model the compressed video z as in (1), where y now corresponds to the original uncompressed video, and η represents the compression artifacts. In what follows, we focus our attention on MPEG-4 compressed videos. In this way, the proposed filter can be applied reliably over different types of data degradations with little need of adjustment or user intervention.

In order to use V-BM4D as a deblocking filter, we need to determine a suitable value of σ to handle the artifacts in a compressed video. To this purpose, we proceed as in the previous section and we identify the optimum value of σ for a set of test sequences compressed at various rates. Figure 5 shows these optimum values plotted against the average bit-per-pixel (bpp) rate of the compressed video and the parameter q that controls the quantization of the block-transform coefficients [26] (Figure 5(a)). Let us observe that both the bpp and q parameters are easily accessible from any given MPEG-4 coded video. These plots suggest that a power law may conveniently explain the relation between the optimum value of σ and both the bpp rate and q. Hence, we fit such bivariate function to the optimum values via least-squares regression, obtaining the adaptive value of σ for the V-BM4D deblocking filter as

$$\sigma(\text{bpp}, q) = 0.09 \cdot q^{1.11} \cdot \text{bpp}^{-0.46} + 3.37 \quad (12)$$

The function $\sigma(\text{bpp}, q)$ is shown in Figure 5 (right). Note that in MPEG-4 the parameter q ranges from 2 to 31, where higher values correspond to a coarser quantization and consequently lower bitrates. As a matter of fact, when q increases and/or bpp decreases, the optimum σ increases, in order to effectively cope with stronger blocking artifacts. Clearly, a much larger

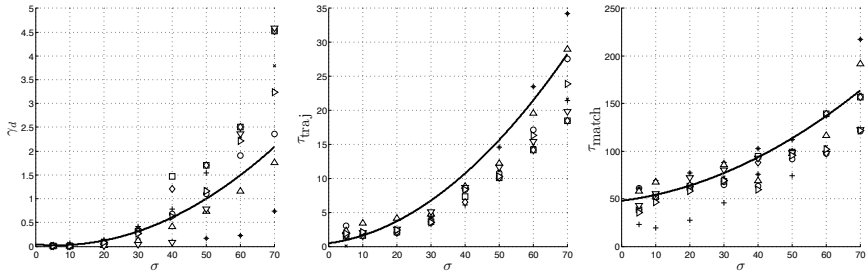
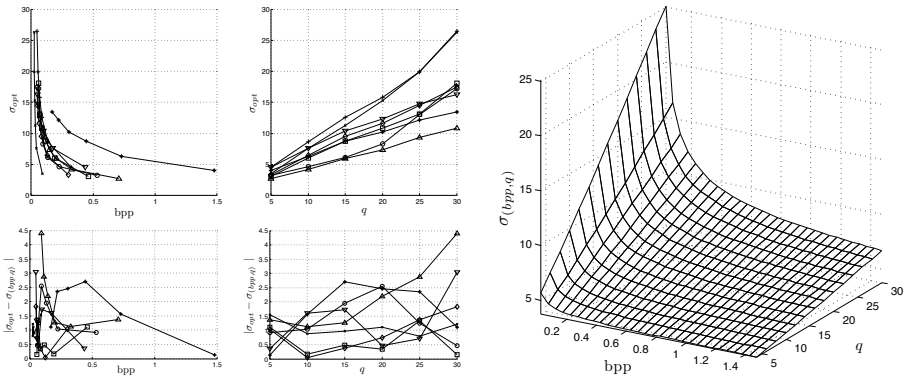


Fig. 4. From left to right, the second-order polynomials (9), (10), and (11) describing the relation between the parameters γ_d, τ_{traj} and τ_{match} and the noise standard deviation σ. The nine coefficients of the three polynomials have been determined by maximizing the sum of the PSNR of the test sequences *Salesman* ($+$), *Tennis* ($\circ$), *Flower Garden* ($*$), *Miss America* ($\times$), *Coastguard* ($\square$), *Foreman* ($\diamond$), *Bus* ($\triangle$), and *Bicycle* ($\triangledown$), corrupted by white Gaussian noise having σ ranging between 5 and 70. As comparison, we superimpose the optimum parameter for each test sequence and σ.



(a) Optimum values of σ plotted against the bit-per-pixel (bpp) rate (top-left) and quantization parameter q (top-right). The bottom plots show the corresponding residual absolute errors.

(b) Fitted curve of the optimum values of σ as a bivariate function $\sigma\,(\text{bpp}, q)$ (12) used by the V-BM4D deblocking filter.

Fig. 5. The sequences used in the fitting are *Salesman* ($+$), *Tennis* ($\circ$), *Flower Garden* ($*$), *Miss America* ($\times$), *Coastguard* ($\square$), *Foreman* ($\diamond$), *Bus* ($\triangle$), and *Bicycle* ($\triangledown$).

value of σ could result in oversmoothing, while much smaller values may not suffice for effectively reducing the compression artifacts. While in this paper we mostly deal with short test sequences, and we compute the bpp as the average rate over the whole sequence, we argue that in practice this rate should be computed as the average over a limited set of frames, namely the so-called group of pictures (GOP) built around each intra-coded frame. In principle, one could learn a model for σ together with all the remaining V-BM4D parameters at once (possibly achieving better results); but this would have increased the risk of overfitting the many parameters to the peculiarities of this compression method, and would have complicated the optimization task.

Let us remark that V-BM4D deblocking can be straightforwardly applied also to videos compressed by other encoders than MPEG-4, because the q parameter can be both estimated as a subjective quality metric for compressed videos, or as an objective measurement [34] on the impairing artifacts to be filtered out.

V. Enhancement

Enhancement is used to improve the video quality, so that the filtered video becomes more pleasing to human subjective judgment and/or better suited for subsequent automatic interpretation tasks, as segmentation or pattern recognition. In particular, by enhancement we refer to the sharpening of degraded details in images (frames) characterized by low contrast.

Among the existing enhancement techniques we mention methods based on histogram manipulation [35], linear and non-linear unsharp masking [36], [37], [38], fuzzy logic [39], and weighted median filter [40], [41]. Transform-domain methods generally apply a nonlinear operator to the transform coefficients of the processed image/video in order to accentuate specific portions of the spectrum, which eventually results

in sharpening of details [42], [43], [35], [13]. One of the most popular technique is alpha-rooting [42], which raises the magnitude of each transform coefficient ϕ_i of the processed spectrum Φ to a power $\frac{1}{\alpha}$, with $\alpha > 1$ as

$$\bar{\phi}_i = \begin{cases} sign\big[\phi_i\big]\big|\phi_0\big|\left|\frac{\phi_i}{\phi_0}\right|^{\frac{1}{\alpha}}, & \text{if } \phi_0 \neq 0, \\ \phi_i, & \text{otherwise,} \end{cases} \tag{13}$$

where ϕ_0 is the DC term and $\bar{\phi}_i$ is the resulting sharpened coefficients. Observe that $\alpha > 1$ induces sharpening, as it scales the large coefficients relatively to the small ones, i.e. those carrying high-frequency information [42]. Although (13) assumes real-valued transform coefficients, it can be generalized to complex-valued ones, observing that alpha-rooting preserves the sign in the former case, and the phase in the latter.

A critical issue in enhancement is the amplification of the noise together with the sharpening of image details [44], [42], an effect that becomes more severe as the amount of applied sharpening increases. In order to cope with this problem, a joint application of a denoising and sharpening filter is often recommendable, and in particular this practice has been investigated in [13], [39].

Enhancement of digital videos, following the approach proposed in [13], can be easily performed by combining the V-BM4D filter with the alpha-rooting operator (13), in order to simultaneously reduce the noise and sharpen the original signal. The V-BM4D sharpening algorithm still comprises the grouping, collaborative filtering and aggregation steps, and it is carried out through the hard-thresholding stage only. The alpha-rooting operator is applied on the thresholded coefficients within the collaborative filtering step, before inverting the 4-D transform. Note that, since the alpha-rooting amplifies the group coefficients, the total variance of the filtered group changes, thus the aggregation weights cannot be estimated from the number of retained non-zero coefficients $N^{\text{har}}_{(\mathbf{x}_0,t_0)}$. A simple estimator is devised in [13], and can be used to define the weights of (5) as

$$w^{\text{har}}_{(\mathbf{x}_0,t_0)} = \frac{1}{\sum_{\Phi(i)\neq 0} w_i \sigma^2},$$

having

$$w_i = \left(1 - \frac{1}{\alpha}\right)^2 |\phi_0|^{-\frac{2}{\alpha}} |\phi_i|^{\frac{2}{\alpha}} + \frac{1}{\alpha^2} |\phi_i|^{\frac{2}{\alpha}-2} |\phi_0|^{2-\frac{2}{\alpha}},$$

where Φ is the transformed spectrum of the group $G^{\text{ht}}_z(\mathbf{x}_0, t_0)$ resulting from hard thresholding, and ϕ_0 is its corresponding DC coefficient. The DC-term is not alpha-rooted, thus its contribution to the total variance of the sharpened group should be σ^2. However, in order to avoid completely flat blocks being awarded with excessively large weights, the weight for the DC-term is set equal to the weight of the smallest retained coefficients, i.e. those having magnitude $\sigma\lambda_{4D}$ as

$$w_0 = \left(1 - \frac{1}{\alpha}\right)^2 |\phi_0|^{-\frac{2}{\alpha}} |\sigma\lambda_{4D}|^{\frac{2}{\alpha}} + \frac{1}{\alpha^2} |\sigma\lambda_{4D}|^{\frac{2}{\alpha}-2} |\phi_0|^{2-\frac{2}{\alpha}}.$$

The separability of the 4-D transform can be exploited to extend this approach, by treating in a different way different portions of the thresholded 4-D spectrum. Let us remind that the 4-D spectrum is structured according to the four dimensions of the corresponding group, i.e. two local spatial, one local temporal, and one for the non-local similarity. In particular, it includes a 2-D surface (face) corresponding to the DC terms of the two 1-D transforms used for decorrelating the temporal and non-local dimensions of the group, and 3-D volume corresponding to the DC term of the 1-D temporal transform. Hence, the value of α can be decreased for the coefficients that do not belong to this 3-D volume, in order to attenuate the temporal flickering artifacts. Likewise, the portion of spectrum in the 2-D surface can be used to characterize the group content as proposed in [45], for example by using lower values of α on flat regions to avoid noise accentuation.

We introduce the sharpening operator in the first stage (hard thresholding) only, as this guarantees excellent subjective results, and we address to future work the application of alpha-rooting during Wiener filtering.

VI. Experiments

In this section we present the experimental results obtained with a C/MATLAB implementation of the V-BM4D algorithm. The filtering performance is measured using the PSNR, computed on the whole processed video as

$$\text{PSNR}(\hat{y}, y) = 10\log_{10}\left(\frac{255^2|X||T|}{\sum_{(\mathbf{x},t)\in X\times T}\big(y((\mathbf{x},t)) - \hat{y}(\mathbf{x},t)\big)^2}\right), \tag{14}$$

where $|X|$ and $|T|$ stand for the cardinality of X and T, respectively. Additionally, we measure the performance of V-BM4D by means of the MOVIE index [46], a recently introduced video quality assessment (VQA) metric that is expected to be closer to the human visual judgement than the PSNR, because it concurrently evaluates space, time and jointly space-time video quality.

The transforms employed in the collaborative filtering are similar to those in [10], [11]: $\mathcal{T}^{\text{ht}}_{4D}$ (used in the hard-thresholding stage) is a 4-D separable composition of 1-D biorthogonal wavelet in both spatial dimensions, 1-D DCT in the temporal dimension, and 1-D Haar wavelet in the fourth (grouping) dimension while, $\mathcal{T}^{\text{wie}}_{4D}$ (used in the Wiener-filtering stage) differs from $\mathcal{T}^{\text{ht}}_{4D}$ as in the spatial dimension it performs a 2-D DCT. Note that, because of the Haar transform, the cardinality M of each group is set to a power of 2. To reduce the complexity of the grouping phase, we restrict the search of similar volumes within a $N_G \times N_G$ neighborhood centered around the coordinates of the reference volume, and we introduce a step of $N_{\text{step}} \in \mathbb{N}$ pixels in both horizontal and vertical directions between each reference volume. Although we set $N_{step} > 1$, we have to compute beforehand the trajectory of every possible volume in the video, since each volume is a potential candidate element of every group. Table I provides a complete overview of the parameters setting in V-BM4D.

The remaining part of this section presents the results of experiments concerning grayscale Denoising (Section VI-A), Deblocking (Section VI-B), Enhancement (Section VI-C), and Color Filtering (Section VI-D).

TABLE I
PARAMETER SETTINGS OF V-BM4D FOR THE FIRST (HARD-THRESHOLDING) AND THE SECOND (WIENER-FILTERING) STAGE. IN THE HARD-THRESHOLDING STAGE, THE THREE PARAMETERS γ_d, τ_{TRAJ}, AND τ_{MATCH} VARY ACCORDING TO THE NOISE STANDARD DEVIATION.

Stage	N	N_S	N_G	h	M	λ_{4D}	γ_p	γ_w	σ_w	N_{step}	γ_d	τ_{traj}	τ_{match}
Hard thr.	8	11	19	4	32	2.7	0.3	0.5	1	6	$\gamma_d(\sigma)$	$\tau_{\text{traj}}(\sigma)$	$\tau_{\text{match}}(\sigma)$
Wiener filt.	7		27		8	*Unused*				4	0.005	1	13.5

TABLE II
DENOISING PERFORMANCE OF V-BM3D AND V-BM4D. THE PSNR (DB) AND MOVIE INDEX [46] (THE LOWER THE BETTER) VALUES ARE REPORTED IN THE LEFT AND RIGHT PART OF EACH CELL, RESPECTIVELY. IN ORDER TO ENHANCE THE READABILITY OF THE RESULTS, EVERY MOVIE INDEX HAS BEEN MULTIPLIED BY 10^3. THE TEST SEQUENCES ARE CORRUPTED BY WHITE GAUSSIAN NOISE WITH DIFFERENT VALUES OF STANDARD DEVIATION σ.

σ	Video:	*Salesm.*	*Tennis*	*Fl. Gard.*	*Miss Am.*	*Coastg.*	*Foreman*	*Bus*	*Bicycle*
	Res.:	288×352	240×352	240×352	288×360	144×176	288×352	288×352	576×720
	Frames:	50	150	150	150	300	300	150	30
5	V-BM4D	**41.00**|**0.02**	**39.02**|**0.03**	**37.24**|**0.02**	**42.16**|**0.03**	**39.27**|**0.02**	**40.34**|**0.03**	**38.35**|**0.04**	**41.04**|**0.02**
	V-BM3D	40.44|**0.02**	38.47|**0.03**	36.46|**0.02**	41.58|**0.03**	38.25|0.03	39.77|0.04	37.55|0.05	40.89|**0.02**
10	V-BM4D	**37.30**|**0.09**	**35.22**|**0.12**	**32.81**|**0.07**	**40.09**|**0.08**	**35.54**|**0.09**	**36.94**|**0.11**	**34.26**|**0.14**	**37.66**|**0.09**
	V-BM3D	37.21|**0.09**	34.68|0.15	32.11|0.09	39.61|0.11	34.78|0.13	36.46|0.13	33.32|0.20	37.62|**0.09**
15	V-BM4D	35.25|0.24	**33.04**|**0.34**	**30.34**|**0.14**	**38.85**|**0.17**	**33.41**|**0.19**	**35.03**|**0.21**	**31.87**|**0.32**	35.61|0.19
	V-BM3D	**35.44**|**0.21**	32.63|0.37	29.81|0.18	38.64|0.20	33.00|0.25	34.64|0.24	31.05|0.45	**35.67**|**0.17**
20	V-BM4D	33.79|**0.46**	**31.59**|**0.60**	**28.63**|**0.23**	**37.98**|**0.27**	**31.94**|**0.32**	**33.67**|**0.33**	**30.26**|**0.53**	34.10|0.30
	V-BM3D	**34.04**|**0.46**	31.20|0.73	28.24|0.28	37.85|0.31	31.71|0.41	33.30|0.38	29.57|0.72	**34.18**|**0.27**
25	V-BM4D	32.66|**0.75**	**30.56**|**0.85**	**27.35**|**0.33**	**37.24**|**0.37**	**30.81**|**0.48**	**32.61**|**0.46**	**29.10**|**0.73**	32.89|0.42
	V-BM3D	**32.79**|0.93	30.11|1.10	27.00|0.39	37.10|0.44	30.62|0.65	32.19|0.55	28.48|1.00	**32.90**|**0.39**
30	V-BM4D	**31.75**|**1.07**	**29.72**|**1.10**	**26.29**|**0.45**	**36.58**|**0.48**	**29.90**|**0.66**	**31.80**|**0.60**	**28.17**|**0.94**	**31.83**|0.56
	V-BM3D	31.68|1.56	29.22|1.46	25.89|0.55	36.41|0.58	29.68|0.96	31.27|0.75	27.59|1.30	31.77|**0.54**
35	V-BM4D	**30.99**|**1.41**	**29.04**|**1.33**	**25.40**|**0.59**	**35.98**|**0.59**	**29.17**|**0.88**	**31.11**|**0.74**	**27.39**|**1.15**	**30.92**|**0.72**
	V-BM3D	30.72|2.36	28.56|1.85	25.16|0.70	35.87|0.74	28.92|1.36	30.56|0.98	26.91|1.61	30.85|0.73
40	V-BM4D	**30.35**|**1.76**	**28.49**|**1.56**	**24.60**|**0.75**	**35.47**|**0.70**	**28.54**|**1.13**	**30.52**|**0.89**	**26.72**|**1.37**	**30.10**|**0.89**
	V-BM3D	29.93|3.09	27.99|2.17	24.33|0.92	35.45|0.89	28.27|1.86	29.97|1.21	26.28|1.93	30.02|0.94

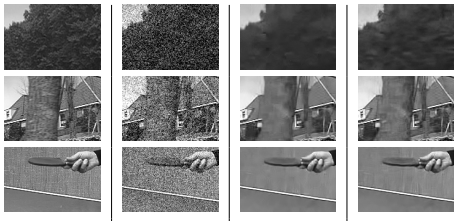


Fig. 6. From top to bottom, visual comparison of the denoising performance of V-BM4D and V-BM3D on the sequences *Bus*, *Flower Garden* and *Tennis* corrupted by white Gaussian noise with standard deviation $\sigma = 40$.

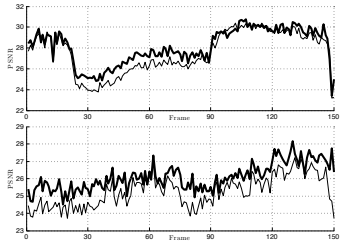


Fig. 7. Frame-by-frame PSNR (dB) output of the sequences *Tennis* (left) and *Bus* (right) corrupted by white Gaussian noise with standard deviation $\sigma = 40$ denoised by V-BM4D (thick line) and V-BM3D (thin line).

A. Grayscale Denoising

We compare the proposed filtering algorithm against V-BM3D [11], as this represents the state of the art in video denoising and we refer the reader to [11] for comparisons with other methods that are less effective than V-BM3D. Table II reports the denoising performance of V-BM3D and V-BM4D in terms of PSNR and MOVIE index. In our experiments the two algorithms are applied to a set of test sequences corrupted by white Gaussian noise with increasing standard deviation σ, which is assumed known. Observations z are obtained by synthetically adding Gaussian noise to grayscale video sequences, according to (1). Further details concerning the original sequences, such as the resolution and number of frames, are reported in the header of the tables.

As one can see, V-BM4D outperforms V-BM3D in nearly all the experiments, with PSNR improvement of almost 1 dB. It is particularly interesting to observe that V-BM4D effectively handles the sequences characterized by rapid motions and frequent scene changes, especially under heavy noise, such as *Tennis*, *Flower Garden*, *Coastguard* and *Bus*. Figure 7 shows that, as soon as the sequence presents a significant change in the scene, the denoising performance decreases

Fig. 8. Deblocking: visual comparison of V-BM4D and MPlayer on few frames. The test sequences (from top to bottom, *Foreman*, *Tennis* and *Coastguard*) have been compressed with the MPEG-4 encoder with quantization parameter $q = 25$.

significantly for both the algorithms, but, in these situations, V-BM4D requires much less frames to recover the previous PSNR values, as shown by the lower peaks at frame 90 of *Tennis*.

Finally, Figure 6 offers a visual comparison of the performance of the two algorithms. As a subjective quality assessment, V-BM4D better preserves textures, without introducing disturbing artifacts in the restored video: this is clearly visible in the tree leaves of the *Bus* sequence or in the background texture of *Tennis*. Such improvement well substantiates the considerable reduction in the MOVIE index values reported in Table II.

B. Deblocking

Table III compares, in terms of objective measurements, the V-BM4D deblocking filter against the *MPlayer accurate deblocking filter*[1], as, to the best of our knowledge, it represents one of the best deblocking algorithm. Eight sequences compressed by the MPEG-4 encoder with different values of the quantization parameter q have been considered: additional details and the bit-per-pixel rates concerning these sequences are reported in the table. Numerical results show that V-BM4D outperforms *Mplayer* in all the experiments, with improvement peaks of almost 2dB in terms of PSNR. For the sake of completeness, we also report the MOVIE index. Observe that, MOVIE often prefers the compressed observation rather than the filtered sequences, thus showing a general preference towards piecewise smooth images. However, let us observe that such results do not conform to the visual quality of the deblocked videos.

Figure 8 shows the results of V-BM4D deblocking on the *Foreman*, *Tennis* and *Coastguard* sequences, encoded at aggressive compression level ($q = 25$). The visual quality of the filtered videos has been significantly improved, since the compression artifacts, such as blocking or ghosting, have been successfully filtered without losing fine image details. In particular, we can note how the face in *Foreman*, the player and the white poster in *Tennis*, and the stone-wall in *Coastguard*, sharply emerge from their blocky counterparts, while almost-uniform areas, such as the white striped building in *Foreman*, or the table and the wall in *Tennis*, have been pleasingly smoothed without introducing blur.

[1]Source code and documentation can be found at `http://sourceforge.net/projects/ffdshow-tryout/` and `http://www.mplayerhq.hu/`

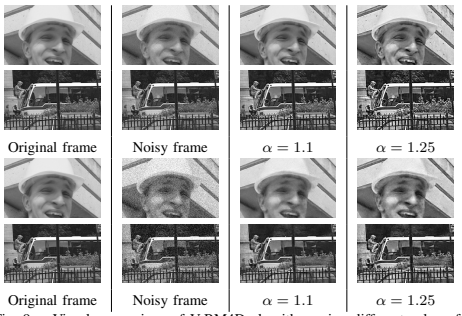


Fig. 9. Visual comparison of V-BM4D algorithm using different value of α. The test sequences, (*Foreman* and *Bus*), have been corrupted by white Gaussian noise with standard deviation $\sigma = 5$ (top) and $\sigma = 25$ (bottom), and have been jointly denoised and sharpened by V-BM4D.

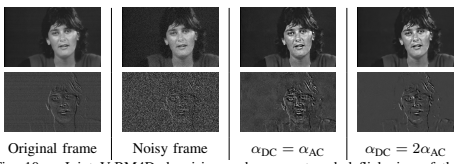


Fig. 10. Joint V-BM4D denoising, enhancement and deflickering of the sequence *Miss America* corrupted by white Gaussian noise with standard deviation $\sigma = 10$. From left to right, the bottom row shows the temporal differences between the frames presented in the top row and the preceding frames in the original, noisy, and enhanced sequences. The right-most column shows the sharpening result using different a α in the temporal DC and AC coefficients of the groups spectra, thus obtaining an effective deflickering yet maintaining spatial sharpness. The images in the bottom row are all drawn with respect to the same gray colormap, which is stretched 4 times in order to improve the visualization.

C. Enhancement

In the enhancement experiments we use the same settings reported in Table I, testing two values of α, i.e. the parameter that controls the amount of sharpening in the alpha-rooting. Figure 9 presents the results of the V-BM4D enhancement filter applied on the *Foreman* and *Bus* sequences, corrupted by white Gaussian noise having standard deviation $\sigma \in \{5, 25\}$, and sharpened using $\alpha = 1.1$, and $\alpha = 1.25$. As the images demonstrate, the combination of V-BM4D and alpha-rooting produces satisfying results, as the fine details are effectively preserved together with a fairly good noise suppression. Such properties allowed the application of the V-BM4D enhancement filter in biomedical imaging, to facilitate the tracking of microtubules in RFP-EB3 time-lapse videomicroscopy sequences corrupted by heavy noise [2].

In particular, V-BM4D sharpens fine details, such as the tree leaves in *Bus*, and reveals barely visible information hidden in the noisy videos, as the background building of *Foreman*. The proposed enhancement filter is minimally susceptible to noise even when strong sharpening is performed (i.e., $\alpha = 1.25$), as shown by the smooth reconstruction of flat areas like the hat of *Foreman* and the bus roof of *Bus*.

TABLE III
DEBLOCKING PERFORMANCE OF V-BM4D AND MPLAYER ACCURATE DEBLOCKING FILTER. THE PSNR (DB) AND MOVIE INDEX [46] (THE LOWER THE BETTER) VALUES ARE REPORTED IN THE LEFT AND RIGHT PART OF EACH CELL, RESPECTIVELY. IN ORDER TO ENHANCE THE READABILITY OF THE RESULTS, EVERY MOVIE INDEX HAS BEEN MULTIPLIED BY 10^3. THE PARAMETER q CONTROLS THE QUANTIZATION MATRIX OF THE MPEG-4 ENCODER AND BPP DENOTES THE AVERAGE BIT-PER-PIXEL RATE OF THE COMPRESSED VIDEO. AS A REFERENCE, WE ALSO SHOW THE PSNR AND MOVIE INDEX OF THE UNFILTERED COMPRESSED (COMPR.) VIDEOS.

q	Video:	*Salesm.*	*Tennis*	*Fl. Gard.*	*Miss Am.*	*Coastg.*	*Foreman*	*Bus*	*Bicycle*
	Res.:	288×352	240×352	240×352	288×360	144×176	288×352	288×352	576×720
	Frames:	50	150	150	150	300	300	150	30
5	bpp	0.3232	0.5323	1.4824	0.0884	0.4609	0.3005	0.7089	0.4315
	V-BM4D	**35.95**|**0.16**	**34.41**|0.18	**33.54**|**0.05**	**39.51**|0.15	**34.75**|**0.13**	**36.49**|0.16	**35.05**|0.13	**38.01**|**0.08**
	Mplayer	35.14|0.17	33.79|**0.17**	32.73|0.07	38.58|0.14	34.00|**0.13**	35.60|**0.14**	34.36|**0.10**	36.53|0.11
	Compr.	35.28|0.17	33.87|**0.17**	32.81|0.07	39.03|**0.13**	34.12|**0.13**	35.70|**0.14**	34.45|**0.10**	36.71|0.11
10	bpp	0.1319	0.2249	0.7288	0.0399	0.1926	0.1276	0.3285	0.2076
	V-BM4D	**32.12**|0.87	**30.39**|0.83	**27.93**|**0.26**	**37.30**|0.48	**30.75**|**0.50**	**32.91**|0.49	**30.69**|0.43	**33.54**|**0.36**
	Mplayer	31.66|1.08	29.87|0.89	27.40|0.31	36.61|0.53	30.23|0.53	32.16|0.52	30.11|0.41	32.45|0.46
	Compr.	31.54|**0.86**	29.84|**0.78**	27.41|0.29	36.66|**0.46**	30.19|0.51	32.09|**0.48**	30.07|**0.36**	32.37|0.46
15	bpp	0.0865	0.1326	0.4470	0.0318	0.1184	0.0812	0.2039	0.1333
	V-BM4D	**30.06**|1.89	**28.48**|1.49	**25.15**|**0.58**	**36.13**|0.82	**28.73**|**1.01**	**31.10**|**0.90**	**28.48**|0.85	**31.16**|**0.79**
	Mplayer	29.65|2.39	28.03|1.52	24.68|0.68	35.59|0.90	28.30|1.10	30.36|0.98	27.89|0.83	30.12|0.95
	Compr.	29.48|**1.78**	27.97|**1.39**	24.67|0.63	35.41|**0.81**	28.18|1.03	30.27|**0.90**	27.83|**0.71**	30.00|0.98
20	bpp	0.0661	0.0943	0.3058	0.0280	0.0852	0.0625	0.1453	0.0985
	V-BM4D	**28.66**|3.03	**27.24**|2.07	**23.34**|**0.95**	**35.02**|**1.21**	**27.42**|**1.73**	**29.85**|1.38	**26.96**|1.38	**29.52**|**1.26**
	Mplayer	28.31|3.76	26.82|2.12	22.90|1.12	32.93|1.58	27.04|1.96	29.12|1.55	26.42|1.42	28.60|1.56
	Compr.	28.11|**2.71**	26.76|**1.93**	22.88|1.02	34.21|**1.21**	26.90|**1.73**	29.03|**1.37**	26.35|**1.16**	28.43|1.58
25	bpp	0.0546	0.0710	0.2225	0.0257	0.0679	0.0523	0.1121	0.0846
	V-BM4D	**27.63**|4.19	**26.34**|2.55	**22.07**|**1.38**	**34.31**|**1.54**	**26.47**|2.53	**29.01**|1.87	**25.93**|1.96	**28.32**|**1.78**
	Mplayer	27.30|5.09	25.96|2.57	21.63|1.64	33.66|1.70	26.11|2.95	28.25|2.13	25.38|2.04	27.35|2.18
	Compr.	27.07|**3.63**	25.85|**2.38**	21.62|1.49	33.45|1.57	25.98|**2.45**	28.10|**1.86**	25.27|**1.66**	27.22|2.20
30	bpp	0.0477	0.0604	0.1697	0.0244	0.0584	0.0480	0.0921	0.0676
	V-BM4D	**26.84**|5.38	**25.59**|2.99	**21.08**|**1.86**	**33.25**|**1.90**	**25.72**|3.53	**28.30**|2.33	**25.06**|2.57	**27.40**|**2.34**
	Mplayer	26.51|6.31	25.26|3.02	20.65|2.24	32.80|2.08	25.38|4.20	27.57|2.68	24.55|2.70	26.54|2.88
	Compr.	26.28|**4.59**	25.11|**2.77**	20.64|1.99	32.39|1.97	25.25|**3.31**	27.37|**2.31**	24.41|**2.19**	26.35|2.88

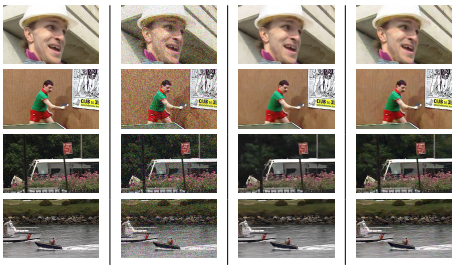
As explained in Section V, the spectrum coefficients of the group can be treated differently along the temporal dimension to attenuate video temporal artifacts such as flickering. In Figure 10 we show the enhancement results of V-BM4D applied to the video *Miss America* corrupted by white Gaussian noise with $\sigma = 10$ using two settings for the sharpening parameter α. In the former experiment a fixed $\alpha_{\{\mathrm{DC,AC}\}} = 1.25$ is used to sharpen the whole spectrum of the groups, while in the latter different values of α are used in the temporal DC and AC planes. In particular, the temporal DC coefficients are sharpened using $\alpha_{\mathrm{DC}} = 1.25$, and the temporal AC are sharpened using the halved value $\alpha_{\mathrm{AC}} = 0.625$. By using different values of α, V-BM4D significantly attenuates the flickering artifacts without compromising the effectiveness of neither the sharpening nor the denoising. In Figure 10, the flickering artifacts of the non-uniform intensities within the temporal difference in the background are clearly visible when the sequence is processed using $\alpha_{\mathrm{DC}} = \alpha_{\mathrm{AC}}$. In contrast, the sequence processed using a modified values of α exhibits a better temporal consistency as demonstrated by the smooth background in the temporal difference, yet maintaining excellent enhancement and noise reduction properties.

TABLE IV
COLOR DENOISING PERFORMANCE OF V-BM3D AND V-BM4D IN TERMS OF PSNR (DB) AND MOVIE INDEX [46] (THE LOWER THE BETTER) VALUES ARE REPORTED IN THE LEFT AND RIGHT PART OF EACH CELL, RESPECTIVELY. IN ORDER TO ENHANCE THE READABILITY OF THE RESULTS, EVERY MOVIE INDEX HAS BEEN MULTIPLIED BY 10^3. THE TEST SEQUENCES ARE CORRUPTED BY WHITE GAUSSIAN NOISE WITH DIFFERENT VALUES OF STANDARD DEVIATION σ.

σ	Video:	*Tennis*	*Coastg.*	*Foreman*	*Bus*
	Res.:	240×352	144×176	288×352	288×352
	Frames:	150	300	300	150
5	V-BM4D	**39.98**|**0.01**	**41.13**|**0.01**	**41.38**|**0.01**	**40.21**|**0.01**
	V-BM3D	39.45|**0.01**	40.18|**0.01**	40.56|**0.01**	39.07|**0.01**
10	V-BM4D	**36.42**|**0.04**	**37.28**|**0.03**	**37.92**|**0.05**	**36.23**|**0.05**
	V-BM3D	36.04|**0.04**	36.82|**0.03**	37.52|0.04	34.96|0.07
20	V-BM4D	**32.88**|**0.17**	**33.61**|**0.13**	**34.62**|**0.15**	**32.27**|**0.20**
	V-BM3D	32.54|0.18	33.39|0.14	34.49|0.16	31.03|0.32
40	V-BM4D	**29.52**|**0.70**	**30.00**|**0.42**	**31.30**|**0.44**	**28.32**|**0.70**
	V-BM3D	29.20|0.82	29.99|0.63	31.17|0.56	27.34|1.32

D. Color Filtering

The proposed V-BM4D algorithm can be extended to color filtering using the same approach of the Color-BM3D image denoising algorithm [10], [21]. We consider the denoising of noisy color videos, such as a RGB videos, having each



Original frame | Noisy frame | V-BM3D | V-BM4D

Fig. 11. Comparison of V-BM3D and V-BM4D color denoising performances. The test sequences (from top to bottom, *Foreman*, *Tennis*, *Bus* and *Coastguard*) have been corrupted by white Gaussian noise with standard deviation $\sigma = 40$.

channel independently corrupted by white Gaussian noise with variance σ^2.

The algorithm proceeds as follows. At first, the RGB noisy video is transformed to a luminance-chrominance color space, then, both the motion estimation and the grouping are computed from the luminance channel only, as this usually has the highest SNR and carries most of the significant information. In fact, image structures do not typically vary among different channels, and the results of the motion estimation and the grouping on the luminance can be directly reused in the two chrominance channels as well. Once the groups are formed, each channel undergoes the collaborative filtering and aggregation independently, and three individual estimates are produced. Eventually, the final RGB estimate is produced by inverting the color space transformation. Such approach is a reasonable tradeoff between the achieved denoising quality and the required computational complexity. Figure 11 compares the denoising performances of V-BM4D against the state-of-the-art V-BM3D filter, on the color sequences *Foreman*, *Tennis*, *Bus*, and *Coastguard*, corrupted by white Gaussian noise having standard deviation $\sigma = 40$. As a subjective assessment, V-BM4D better preserves fine details, such as the face and the background building in *Foreman*, the background texture in *Tennis*, the leaves in *Bus* and the grass in *Coastguard*. From an objective point of view, as reported in Table IV, V-BM4D performs better than V-BM3D in every experiment, with PSNR gains of up to 1.5dB. The MOVIE index confirms the superior performances of V-BM4D, especially when the observations are corrupted with high level of noise.

VII. Complexity

In our analysis the complexity of the algorithm is measured through the number of basic arithmetic operations performed; other factors that may also influence the execution time, such as the number of memory accesses or memory consumption, have not been considered.

Each run of V-BM4D involves the execution of the hard-thresholding stage (whose complexity is $\mathcal{C}^{\text{ht}}_{\text{V-BM4D}}$), of the Wiener-filtering stage (whose complexity is $\mathcal{C}^{\text{wie}}_{\text{V-BM4D}}$), and two runs of the motion estimation algorithm (whose complexity is $\mathcal{C}_{\text{CT}}$). Hence, the V-BM4D overall complexity is:

$$\mathcal{C}_{\text{V-BM4D}} = 2\mathcal{C}_{\text{CT}} + \mathcal{C}^{\text{ht}}_{\text{V-BM4D}} + \mathcal{C}^{\text{wie}}_{\text{V-BM4D}}. \tag{15}$$

Differently, V-BM3D does not require any motion estimation, and thus its complexity ($\mathcal{C}_{\text{V-BM3D}}$) is given by the sum of the complexity of its hard-thresholding ($\mathcal{C}^{\text{ht}}_{\text{V-BM3D}}$) and Wiener-filtering ($\mathcal{C}^{\text{wie}}_{\text{V-BM3D}}$) stages:

$$\mathcal{C}_{\text{V-BM3D}} = \mathcal{C}^{\text{ht}}_{\text{V-BM3D}} + \mathcal{C}^{\text{wie}}_{\text{V-BM3D}}. \tag{16}$$

Table V shows a comprehensive summary of the parameters involved in the complexity analysis, as well as a brief description of their role in the algorithm. To provide a fair comparison, we assume that in V-BM4D the number of blocks in any spatiotemporal volume (referred as $\bar{h}$) coincides with the size of temporal search window N_{FR} in V-BM3D; similarly, we assume that the number of grouped volumes in V-BM4D (referred to as M) corresponds to the number of grouped blocks in V-BM3D.

TABLE V
Summary of the parameters involved in the complexity analysis.

Parameter	*Notes*
T	Total number of frames in the video.
n	Number of pixels per frame (i.e. $\#X$).
N	Size of the 2-D square blocks.
h	Temporal extent of the volumes in V-BM4D, size of the temporal search window in V-BM3D, corresponding to $(2N_{FR}+1)$ in [11].
N_S	Size of the motion estimation window.
M	Size of the groups, that is the number of grouped volumes in V-BM4D or the number of grouped blocks in V-BM3D.
N_G	Size of the window used in the grouping.
N_{step}	Processing step (refer to Section VI for further details).
$\mathcal{C}_{(m,p,n)}$	Numeric operations required by a multiplication between matrices of size $m \times p$ and $p \times n$ (i.e. the cost of a linear transformation).

A. Computation of the Trajectory

The computation of the trajectory requires searching for the most similar block within an adaptive search window of size $N_S \times N_S$ once for each of the preceding and following frames, i.e. $\bar{h}-1$ times. Computing the ℓ^2 distance between a pair of blocks consists in $3N^2$ operations, as it requires two additions and one multiplication for each of the corresponding pixel. Since a trajectory is constructed for each pixel in every frame of the video, the total cost is

$$\mathcal{C}_{\text{CT}} = nT(\bar{h}-1)N_S^2\left(3N^2\right). \tag{17}$$

B. Hard-Thresholding Stage

In the hard-thresholding stage, for each processed block according to N_{step}, at most M similar volumes are first extracted within a search window of size $N_G \times N_G$, then stacked together in a group, and finally transformed by a separable 4-D transform. Observe that the hard-thresholding, which is performed via element-wise comparison, requires one arithmetical operation per pixel. Eventually, the basic estimate is obtained by aggregating the inverse 4-D transform of the

filtered groups. Thus, we obtain:

$$\mathcal{C}^{\text{ht}}_{\text{V-BM4D}} = \frac{n}{N^2_{\text{step}}} T\Big(\underbrace{N_G^2 3\bar{h} N^2}_{\text{Grouping}} + \underbrace{2\left(2M\bar{h}\mathcal{C}_{(N,N,N)} + M\mathcal{C}_{(\bar{h},\bar{h},N^2)} + \mathcal{C}_{(M,M,\bar{h}N^2)}\right)}_{\text{Forward and Inverse Transformations}} + \underbrace{M\bar{h}N^2}_{\text{Thresholding}} + \underbrace{M\bar{h}N^2}_{\text{Aggregation}} \Big), \tag{18}$$

where the symbol $\mathcal{C}_{(\cdot,\cdot,\cdot)}$ stands for the cost of a matrix multiplication operation, as explained in Table V, and the factor 3 in the grouping complexity is due to the computation of the ℓ^2 distance between two 3-D volumes of size $N \times N \times \bar{h}$. The cost of the is the sum of four matrix multiplications, one for each dimension of the group, as this is linear and separable

In V-BM3D, the grouping is accomplished by predictive-search block-matching [11]: briefly it performs a full-search within a $N_G \times N_G$ window in the first frame to extract the N_B best-matching blocks, then, in the following $\bar{h}$ frames, it inductively searches for other N_B best-matching blocks within windows of size $N_{PR} \times N_{PR}$ (with $N_{PR} \ll N_G$) centered at the position of the previous N_B blocks. Furthermore, since the fourth dimension is missing, the algorithm performs a 3-D transform of the M blocks of each group. The complexity of this stage is:

$$\mathcal{C}^{\text{ht}}_{\text{V-BM3D}} = \frac{n}{N^2_{\text{step}}} T\Big(\underbrace{\left(N_G^2 + N_B\bar{h}N_{PR}^2\right) 3N^2}_{\text{Grouping}} + \underbrace{2\left(2M\mathcal{C}_{(N,N,N)} + \mathcal{C}_{(M,M,N^2)}\right)}_{\text{Forward and inverse transformations}} + \underbrace{MN^2}_{\text{Thresholding}} + \underbrace{MN^2}_{\text{Aggregation}} \Big). \tag{19}$$

C. Wiener-filtering Stage

The complexity of the Wiener-filtering stage can be expressed as that of hard-thresholding stage in (18), with the exception that the transformation involves two groups having equal size, and that the coefficients shrinkage (performed via element-wise multiplication) involves the computation of a set of weights, which requires 6 arithmetic operations per pixel:

$$\mathcal{C}^{\text{wie}}_{\text{V-BM4D}} = \frac{n}{N^2_{\text{step}}} T\Big(\underbrace{N_G^2 3\bar{h} N^2}_{\text{Grouping}} + \underbrace{4\left(2M\bar{h}\mathcal{C}_{(N,N,N)} + M\mathcal{C}_{(\bar{h},\bar{h},N^2)} + \mathcal{C}_{(M,M,\bar{h}N^2)}\right)}_{\text{Forward and Inverse Transformations}} + \underbrace{6M\bar{h}N^2}_{\text{Shrinkage}} + \underbrace{M\bar{h}N^2}_{\text{Aggregation}} \Big). \tag{20}$$

Analogously, in V-BM3D the complexity of Wiener-filtering stage is

$$\mathcal{C}^{\text{wie}}_{\text{V-BM3D}} = \frac{n}{N^2_{\text{step}}} T\Big(\underbrace{\left(N_G^2 + N_B\bar{h}N_{PR}^2\right) 3N^2}_{\text{Grouping}} + \underbrace{4\left(2M\mathcal{C}_{(N,N,N)} + \mathcal{C}_{(M,M,N^2)}\right)}_{\text{Forward and Inverse Transformations}} + \underbrace{6MN^2}_{\text{Shrinkage}} + \underbrace{MN^2}_{\text{Aggregation}} \Big). \tag{21}$$

TABLE VI
SCALABILITY OF THE V-BM4D DENOISING ALGORITHM. THE TEST SEQUENCE IS *Tennis*, CORRUPTED BY WHITE GAUSSIAN NOISE HAVING $\sigma = 25$. THE PARAMETERS M, N_G, AND $N_{\text{STEP}} = 6$ HAVE BEEN USED IN BOTH THE HARD-THRESHOLDING AND WIENER-FILTERING STAGE. TWO DIFFERENT MOTION ESTIMATION STRATEGIES HAVE BEEN EMPLOYED: A FAST DIAMOND SEARCH [47] MODIFIED IN ORDER TO INCORPORATE THE PENALTY TERM DESCRIBED IN SECTION III-A2 INTO THE BLOCK MATCHING, AND THE ONE PROPOSED IN SECTION III-A. THE TIME REQUIRED TO FILTER A SINGLE FRAME, AND (IN PARENTHESIS) THE TIME SOLELY SPENT DURING THE MOTION ESTIMATION ARE REPORTED IN THE LAST COLUMN.

Mot. est.	M	N_G	*PSNR*	*1 / fps*	
Mod. [47]	1	1	29.88	3.07	(2.8)
	1	19	29.88	7.36	(2.8)
	32	19	30.17	14.57	(2.8)
Sec. III-A	1	1	30.07	22.42	(22.1)
	1	19	30.07	26.76	(22.1)
	32	19	30.32	33.99	(22.1)

D. Comparative Analysis

The complexities of V-BM3D and V-BM4D scale linearly with the number of processed pixels, thus both algorithms are $\mathcal{O}(n)$. However, it is worth carrying out a deeper analysis since different multiplying factors may have a remarkable impact on the final cost of the two algorithms. In this comparison we assume that V-BM3D and V-BM4D share the same parameters. In this manner, we can analyze the complexities by comparing the corresponding terms of the cost expansions (18) and (19). At first, we observe that costs of the grouping can be neglected since they are similar in both algorithms. Differently, in V-BM4D the coefficients shrinkage (in the Wiener stage) and the aggregation (in both the Wiener and hard-thresholding stages) require exactly $\bar{h}$ times more operations than in V-BM3D. We can easily compare the complexity of the transformation, as in V-BM4D it involves the additional dimension corresponding to the spatiotemporal volumes. Therefore we can conclude that the overall cost due to the transformation is more than $\bar{h}$ times the corresponding cost in V-BM3D. An analogous inference can be made also for the costs of the Wiener-filtering stage given in (20) and (21). In conclusion, we can state that in these conditions, V-BM4D is at least $\bar{h}$ times computationally more demanding than V-BM3D. However, V-BM4D is also burdened by the motion-estimation step, whose cost is expanded in (17). Let us observe that this cost can be entirely eliminated when the input video is encoded with a motion-compensated algorithm, such as MPEG-4 or H.264, since the motion vectors required to build the spatiotemporal volumes can be directly extracted from the encoded video.

Table VI reports the PSNR values and the corresponding seconds per frame required by V-BM4D to process the video

Tennis (CIF resolution) on a single 3GHz core. We use different settings to quantify the computational load of the grouping and the filtering, by modifying in both stages the size of the search window N_G and the number of grouped volumes M, respectively. Then, we analyze two different motion estimation strategies, specifically the predictive search described in Section III-A and the fast diamond search algorithm presented in [47] modified to incorporate the penalty term described in Section III-A2 into the block matching. Finally, we fix $N_{\text{step}} = 6$ in both stages to keep the average frame-per-second (*fps*) count unbiased. All the remaining V-BM4D parameters are set as in Table I. The speed-ups induced by the fast motion estimation algorithm (~8x), the smaller search window (~15x), or the smaller group size (~2.5x), correspond to marginal PSNR losses, thus demonstrating the good scalability properties of the proposed V-BM4D. Note that, when the nonlocality features are disabled (i.e. $M = 1$ and $N_G = 1$) the motion estimation does not need to be performed for every block in the video, because only one block every N_{step} in both spatial directions is actually processed during the filtering. Thus, by skipping the motion estimation of the useless blocks, it is possible to achieve an additional speed-up of ~12x that allows V-BM4D to process nearly 4 *fps* without affecting the final reconstruction quality.

VIII. Discussion

As anticipated in the introduction, a severe limitation of V-BM3D lies in the grouping step, because it does not distinguish between the nonlocal and temporal correlation within the data. The improved effectiveness of V-BM4D indicates the importance of separately treating different types of correlation, and of explicitly accounting the motion information. In what follows we analyze how the PSNR provided by the two algorithms change when a temporal-based or nonlocal-based grouping is encouraged by varying the parameters that control the grouping strategy (both int the hard-thresholding and Wiener-filtering stage), i.e. (M,h) in V-BM4D and (N_B,N_{FR}) in V-BM3D. In these experiments we consider two videos: *Salesman* and *Tennis*, being representative of a static and a dynamic sequence, respectively.

We recall that for a given pair (M,h) V-BM4D builds volumes having temporal extent up to $2h+1$ and stacks up to M of such volumes in the grouping step. In this analysis, we consider the pairs (M,h) = (1,7), which yields groups composed of a single volume having temporal extent 15, and (M,h) = (16,0), which yields groups composed of 16 volumes of extent having temporal extent 1. These settings correspond to a temporal-based grouping strategy in the former case, and to a nonlocal-based grouping strategy in the latter. The results reported in Table VII show that, although the temporal-based groups have a smaller number of blocks than the nonlocal-based groups, they yield a PSNR improvement of about 17% in *Salesman* and 13% in *Tennis* with respect to the basic configuration (M,h) = (1,0). In contrast, the PSNR improvement inducted by nonlocal-based groups is only about 4% in *Salesman* and 3% in *Tennis*. Note that the size of the groups in V-BM4D can be reduced down to one, somehow resembling V-BM3D, without suffering from a substantial loss in terms of restoration quality. As a matter of fact, the PSNR values shown in Table VII when $M = 1$ are only less than 0.2dB worse than the corresponding results reported in Table II, obtained using bigger values of M. Interestingly, the sequence *Salesman* shows a regular loss in performance for every $h \geq 3$ as the dimension of the groups M increases, thus manifesting that in stationary videos the nonlocality actually worsen the correlation properties of the groups.

To reproduce the nonlocal-based grouping strategy in V-BM3D, we increase the parameter N_B, controlling the number of self-similar blocks to be followed in the adjacent frames, and further we set $d_s = 0$ to give no preference towards blocks belonging to different frames (i.e. blocks having the same coordinates of the reference one [11]). Additionally we fix the maximum size of the groups to $N_2 = 16$, so that bigger groups can be formed as N_{FR} and/or N_B increase. We stress that the group composition in V-BM3D is not known when $N_B \times N_{FR} > N_2$, since the number of potential block candidates is greater than the maximum size of the group, and such candidates are unpredictably extracted from both the nonlocal and temporal dimension. Figure 12 illustrates the V-BM3D denoising performance. Similarly to V-BM4D, the graph shows a consistent PSNR improvement along the temporal dimension (i.e. as N_{FR} increases), and an almost regular loss along the nonlocal dimension (i.e. as N_B becomes larger).

This analysis empirically demonstrates that, 1) in our framework, the nonlocal spatial correlation within the data does not dramatically affect the global PSNR of the restored video, although it becomes crucial in sequences in which the temporal correlation can not be exploited (e.g., having frequent occlusions and scene changes), and 2) a grouping based only on temporal-correlated data always guarantees, both in V-BM4D and V-BM3D, higher performance than a grouping that only exploits nonlocal spatial similarity. Additionally, if the volumes are composed by blocks having the same spatial coordinate (i.e. zero motion assumption, or equivalently $\gamma_d = \infty$), the denoising quality significantly decreases: in the case of *Flower Garden* and $\sigma = 25$, the PSNR loss is ~2.5dB.

IX. Conclusions

Experiments show that V-BM4D outperforms V-BM3D both in terms of objective (denoising) performance (PSNR, MOVIE index), and of visual appearance (as shown in Figure 6 and 11), thus achieving state-of-the-art results in video denoising. In particular, V-BM4D can restore fine image details much better than V-BM3D, even in sequences corrupted by heavy noise ($\sigma = 40$): this difference is clearly visible in the processed frames shown in Figure 6. However, the computational complexity of V-BM4D is obviously higher than V-BM3D, because of the motion-estimation step and the need to process higher-dimensional data. Our analysis of the V-BM4D and V-BM3D frameworks highlights that the temporal correlation is a key element in video denoising, and that it represents an effective prior that has to be exploited when designing nonlocal video restoration algorithms. Thus, V-BM4D can

TABLE VII
PSNR (dB) OUTPUTS OF V-BM4D TUNED WITH DIFFERENT SPACE (M) AND TIME (h) PARAMETERS COMBINATIONS. RECALL THAT THE TEMPORAL EXTENT IS DEFINED AS $2h + 1$. THE TEST SEQUENCES *Salesman* AND *Tennis* HAVE BEEN CORRUPTED BY WHITE GAUSSIAN NOISE WITH STANDARD DEVIATION $\sigma = 20$.

M	*Video*	h								
		0	1	2	3	4	5	6	7	8
1	*Salesm.*	29.22	32.22	33.12	33.54	33.78	33.93	34.03	34.10	34.16
	Tennis	28.04	30.38	31.04	31.33	31.48	31.56	31.61	31.64	31.65
2	*Salesm.*	29.70	32.19	32.90	33.20	33.37	33.45	33.50	33.52	33.52
	Tennis	28.42	30.54	31.15	31.42	31.55	31.62	31.65	31.67	31.67
4	*Salesm.*	30.08	32.32	32.92	33.14	33.22	33.24	33.22	33.18	33.13
	Tennis	28.63	30.62	31.18	31.42	31.52	31.56	31.57	31.56	31.53
8	*Salesm.*	30.35	32.51	33.11	33.36	33.46	33.49	33.48	33.45	33.40
	Tennis	28.74	30.65	31.21	31.44	31.55	31.60	31.61	31.60	31.57
16	*Salesm.*	30.47	32.65	33.29	33.57	33.72	33.79	33.82	33.81	33.80
	Tennis	28.78	30.66	31.21	31.45	31.56	31.63	31.65	31.66	31.65

TABLE VIII
PSNR (dB) OUTPUTS OF V-BM3D TUNED WITH DIFFERENT SPACE (N_B) AND TIME (N_{FR}) PARAMETERS COMBINATIONS. THE SIZE OF THE 3-D GROUPS HAS BEEN SET TO $N_2 = 16$ IN BOTH WIENER AND HARD-THRESHOLDING STAGES; ADDITIONALLY WE SET THE DISTANCE PENALTY TO $d_s = 0$. THE TEST SEQUENCES *Salesman* AND *Tennis* HAVE BEEN CORRUPTED BY WHITE GAUSSIAN NOISE WITH STANDARD DEVIATION $\sigma = 20$.

N_B	*Video*	N_{FR}								
		1	3	5	7	9	11	13	15	17
1	*Salesm.*	29.21	30.83	32.43	32.39	33.43	33.46	33.48	33.46	33.96
	Tennis	27.89	29.29	30.42	30.40	30.93	30.94	30.94	30.93	31.04
3	*Salesm.*	29.50	32.06	32.53	32.99	33.24	33.37	33.51	33.64	33.75
	Tennis	28.13	29.78	30.29	30.39	30.61	30.70	30.79	30.87	30.96
7	*Salesm.*	29.84	31.90	32.43	32.78	33.04	33.20	33.36	33.50	33.61
	Tennis	28.31	29.64	30.07	30.27	30.51	30.62	30.72	30.82	30.91
11	*Salesm.*	30.15	31.83	32.39	32.75	33.02	33.18	33.34	33.49	33.60
	Tennis	28.45	29.58	30.03	30.25	30.50	30.61	30.71	30.81	30.90
15	*Salesm.*	30.15	31.81	32.38	32.75	33.02	33.18	33.34	33.48	33.59
	Tennis	28.45	29.56	30.02	30.25	30.50	30.60	30.71	30.81	30.90

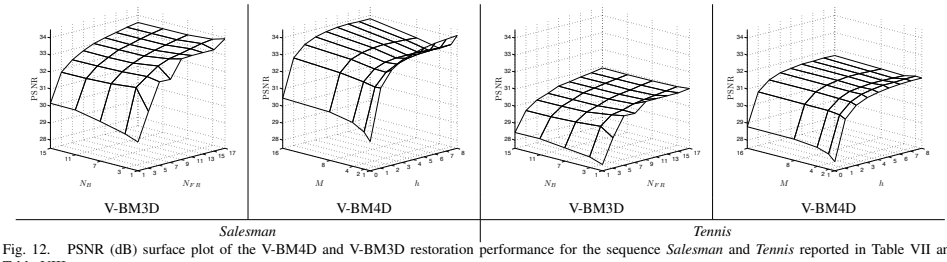


Fig. 12. PSNR (dB) surface plot of the V-BM4D and V-BM3D restoration performance for the sequence *Salesman* and *Tennis* reported in Table VII and Table VIII.

be a viable alternative to V-BM3D especially in applications where the highest restoration quality is paramount or when the separation of the four dimensions is essential.

V-BM4D can be also used as a joint denoising and sharpening filter, as well as a deblocking filter providing excellent performance on both objective and subjective visual quality. Additionally, by exploiting the separability of the 4-D transform, spatiotemporal artifacts (such as flickering) can be alleviated by acting differently on different transform coefficients. Furthermore, we remark that V-BM4D can be extended to color data filtering in each of its applications, namely denoising, deblocking and sharpening.

REFERENCES

[1] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4-D nonlocal spatiotemporal transforms," in *SPIE Electronic Imaging*, Jan. 2011.

[2] M. Maggioni, R. Mysore, E. Coffey, and A. Foi, "Four-dimensional collaborative denoising and enhancement of timelapse imaging of mCherry-EB3 in hippocampal neuron growth cones," in *BioPhotonics and Imaging Conference (BioPIC)*, Oct. 2010.

[3] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Trans. on Image Proc.*, vol. 18, no. 1, pp. 27–35, 2009.

[4] M. Ghoniem, Y. Chahir, and A. Elmoataz, "Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs," *Signal Processing*, vol. 90, no. 8, pp. 2445–2455, 2010, special Section on Processing and Analysis of High-Dimensional Masses of Image and Signal Data.

[5] J. S. De Bonet, "Noise reduction through detection of signal redundancy," Rethinking Artificial Intelligence, MIT AI Lab, Tech. Rep., 1997.

[6] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision*, vol. 86, no. 1, pp. 1–32, 2010.

[7] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[8] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," *Int. Journal of Computer Vision*, vol. 76, no. 2, pp. 123–139, 2008.

[9] X. Li and Y. Zheng, "Patch-based video processing: a variational bayesian approach," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 29, pp. 27–40, Jan. 2009.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, August 2007.

[11] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *European Signal Processing Conference (EUSIPCO)*, Poznan, Poland, Sep. 2007.

[12] G. Boracchi and A. Foi, "Multiframe raw-data denoising based on block-matching and 3-D filtering for low-light imaging and stabilization," in *Int. Workshop on Local and Non-Local Approx. in Image Proc. (LNLA)*, 2008.

[13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Joint image sharpening and denoising by 3D transform-domain collaborative filtering," in *Int. TICSP Workshop Spectral Meth. Multirate Signal Process. (SMMSP)*, 2007.

[14] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Image and video super-resolution via spatially adaptive block-matching filtering," in *Int. Workshop on Local and Non-Local Approx. in Image Process. (LNLA)*, August 2008.

[15] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, "Image upsampling via spatially adaptive block-matching filtering," in *European Signal Processing Conference, (EUSIPCO)*, 2008.

[16] 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, Sep. 2010.

[17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," in *SPIE Electronic Imaging*, vol. 6812, no. 6812-1D, Jan. 2008.

[18] H.-M. Hang, Y.-M. Chou, and S.-C. Cheng, "Motion estimation for video coding standards," *Journal of VLSI Signal Processing Systems*, vol. 17, no. 2/3, pp. 113–136, 1997.

[19] R. Megret and D. Dementhon, "A survey of spatio-temporal grouping techniques," Tech. Rep., 2002.

[20] A. Basharat, Y. Zhai, and M. Shah, "Content based video matching using spatiotemporal volumes," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 360–377, 2008.

[21] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Int. Conf. Image Process. (ICIP)*, Sep. 2007.

[22] H. Oktem, V. Katkovnik, K. Egiazarin, and J. Astola, "Local adaptive transform based image denoising with varying window size," in *Int. Conf. on Image Proc.*, vol. 1, 2001, pp. 273–276.

[23] O. Guleryuz, "Weighted overcomplete denoising," in *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*, vol. 2, Nov. 2003, pp. 1992–1996.

[24] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer J.*, vol. 7, pp. 308–313, 1965.

[25] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM J. of Opt.*, vol. 9, pp. 112–147, 1998.

[26] "The MPEG-4 video standard verification model," pp. 142–154, 2001.

[27] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[28] T. Chen, H. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 5, pp. 594–602, May 2001.

[29] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Multiframe blocking-artifact reduction for transform-coded video," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 12, no. 4, pp. 276–282, April 2002.

[30] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," pp. 33–35, Feb. 1998.

[31] S.-W. Hong, Y.-H. Chan, and W.-C. Siu, "Subband adaptive regularization method for removing blocking effect," vol. 2, p. 2523, Oct. 1995.

[32] A. Wong and W. Bishop, "Deblocking of block-transform compressed images using phase-adaptive shifted thresholding," in *IEEE Int. Symp. on Multimedia (ISM)*, Dec. 2008, pp. 97–103.

[33] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, May 2007.

[34] Z. Wang, A. Bovik, and B. Evan, "Blind measurement of blocking artifacts in images," in *Int. Conf. on Image Proc. (ICIP)*, vol. 3, 2000, pp. 981–984.

[35] S. Agaian, B. Silver, and K. Panetta, "Transform coefficient histogram-based image enhancement algorithms using contrast entropy," *IEEE Trans. on Image Proc.*, vol. 16, no. 3, pp. 741–758, Mar. 2007.

[36] G. Ramponi, "Polynomial and rational operators for image processing and analysis," pp. 203–223, 2001.

[37] G. Ramponi, N. Strobel, S. K. Mitra, and T. Yu, "Nonlinear unsharp masking methods for image contrast enhancement," *Journal of Electronic Imaging*, vol. 5, pp. 353–366, July 1996.

[38] A. Polesel, G. Ramponi, and V. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Trans. on Image Proc.*, vol. 9, no. 3, pp. 505–510, Mar. 2000.

[39] F. Russo, "An image enhancement technique combining sharpening and noise reduction," in *IEEE Instrumentation and Measurement Technology Conference (IMTC)*, vol. 3, 2001, pp. 1921–1924.

[40] T. Aysal and K. Barner, "Quadratic weighted median filters for edge enhancement of noisy images," *IEEE Trans. on Image Proc.*, vol. 15, no. 11, pp. 3294–3310, Nov. 2006.

[41] J. Fischer, M. and. Paredes and G. Arce, "Image sharpening using permutation weighted medians," in *in Proc. X Eur. Signal Processing Conf. Tampere, Finland*, Sep. 2000.

[42] S. Aghagolzadeh and O. K. Ersoy, "Transform image enhancement," *Optical Engineering*, vol. 31, no. 3, pp. 614–626, 1992. [Online]. Available: http://link.aip.org/link/?JOE/31/614/1

[43] S. Hatami, R. Hosseini, M. Kamarei, and H. Ahmadi, "Wavelet based fingerprint image enhancement," in *IEEE Int. Symposium on Circuits and Systems (ISCAS)*, vol. 5, May 2005, pp. 4610–4613.

[44] J. McClellan, "Artifacts in alpha-rooting of images," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, Apr. 1980, pp. 449–452.

[45] H. Tong and A. Venetsanopoulos, "A perceptual model for JPEG applications based on block classification, texture masking, and luminance masking," in *Int. Conf. on Image Proc. (ICIP)*, vol. 3, Oct. 1998, pp. 428–432.

[46] K. Seshadrinathan and A. C. Bovik, "Motion tuned spatio-temporal quality assessment of natural videos," *IEEE Trans. on Image Proc.*, vol. 19, no. 2, pp. 335–350, 2010.

[47] Y. Ismail, J. McNeelly, M. Shaaban, and M. Bayoumi, "Enhanced efficient diamond search algorithm for fast block motion estimation," in *IEEE Int. Symposium on Circuits and Systems (ISCAS).*, May 2009, pp. 3198–3201.

Publication IV

M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Transactions on Image Processing*, 22(1):119–133, Jan. 2013

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)

This work was supported by the Academy of Finland (project no. 213462, Finnish Programme for Centres of Excellence in Research 2006-2011, project no. 129118, Postdoctoral Researcher's Project 2009-2011, and project no. 252547, Academy Research Fellow 2011-2016), and by Tampere Graduate School in Information Science and Engineering (TISE).

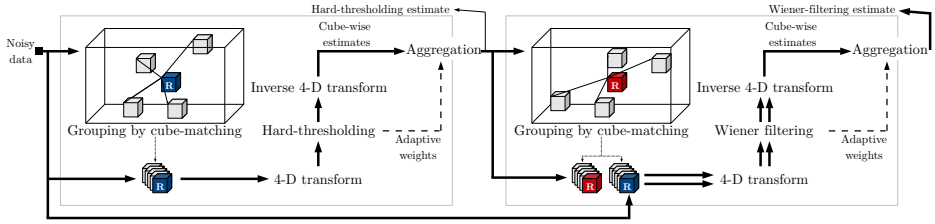


Fig. 1. Flow-diagram of the proposed BM4D algorithm. In both Hard-thresholding (left box) and Wiener-filtering (right box) stage, the grouping, collaborative filtering and aggregation steps are performed for each reference cube of the observed volumetric data.

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 $C^z_{x_R}$ 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 $C^z_{x_R}$. Specifically, the similarity between two cubes is measured via the photometric distance

$$d\big(C^z_{x_i}, C^z_{x_j}\big) = \frac{\big|\big|C^z_{x_i} - C^z_{x_j}\big|\big|_2^2}{L^3}, \tag{2}$$

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 $C^z_{x_R}$. Two cubes are considered similar if their distance (2) is smaller than or equal to a predefined threshold $\tau^{\text{ht}}_{\text{match}}$ which thus controls the minimum accepted cube-similarity. Formally, we first define a set containing the indices of the cubes similar to $C^z_{x_R}$ as

$$S^z_{x_R} = \left\{ x_i \in X : d\left(C^z_{x_R}, C^z_{x_i}\right) \leq \tau^{\text{ht}}_{\text{match}} \right\}. \tag{3}$$

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

$$\mathbf{G}^z_{S^z_{x_R}} = \coprod_{x_i \in S^z_{x_R}} C^z_{x_i}, \tag{4}$$

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 $C^z_{x_R}$.

During the collaborative filtering step, four 1-D decorrelating linear transform, which we denote as a joint four-dimensional transform $\mathcal{T}^{\text{ht}}_{4D}$, 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

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

The transform $\mathcal{T}^{\text{ht}}_{4D}$ 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 $\hat{\mathbf{G}}^y_{S^y_{x_R}}$, is produced by inverting the four-dimensional transform as

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

being each $\hat{C}^y_{x_i}$ an estimate of the original $C^y_{x_i}$ 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

$$\hat{y}^{\text{ht}} = \frac{\sum_{x_R \in X}\left(\sum_{x_i \in S^z_{x_R}} w^{\text{ht}}_{x_R} \hat{C}^y_{x_i}\right)}{\sum_{x_R \in X}\left(\sum_{x_i \in S^z_{x_R}} w^{\text{ht}}_{x_R} \chi_{x_i}\right)}, \tag{7}$$

where $w^{\text{ht}}_{x_R}$ are group-dependent weights, $\chi_{x_i} : X \to \{0, 1\}$ is the characteristic (indicator) function of the domain of $\hat{C}^y_{x_i}$ (i.e. $\chi_{x_i} = 1$ over the coordinates of the voxels of $\hat{C}^y_{x_i}$ and $\chi_{x_i} = 0$ elsewhere), and every $\hat{C}^y_{x_i}$ 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

$$w^{\text{ht}}_{x_R} = \frac{1}{\sigma^2 N^{\text{ht}}_{x_R}}, \tag{8}$$

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

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

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

The collaborative filtering is implemented as an empirical Winer 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

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

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

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

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

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

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

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

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

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

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, $z \sim \mathcal{R}\left(y,\sigma\right)$ 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

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

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

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

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*	

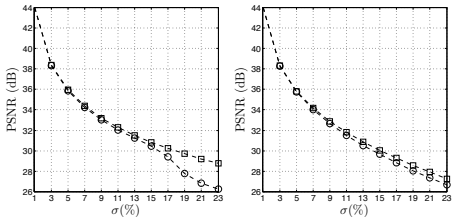


Fig. 2. PSNR denoising performance of BM4D under the normal (∘) and modified (□) profile applied to the BrainWeb phantom [7] corrupted by i.i.d. Gaussian noise (left) and Rician noise (right) with varying level of σ.

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.

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)	$1.0 \times 1.0 \times 1.25$

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	32.66|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

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

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

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

$k = 1$ 3

while $k \leq k_{\text{final}}$ 4

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

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

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

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

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

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

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

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

$k \leftarrow k+1$ 13

end 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 $k \geq 1$, 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

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

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

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

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

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

and (excited) phase

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

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

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

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 $\hat{\phi}^{(k)}$ 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

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

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

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

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

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

and for (21)

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

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

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

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 $\tilde{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ and $\hat{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ become equivalent; conversely, when observed spectrum is noisy, $\tilde{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ 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 $\sigma^{(k)}_{\text{excite}} = \sigma$ for all k, Algorithm 1 coincides with a one-time application of the filter Φ on $\hat{y}^{(0)}_{\text{excite}} = \left|\mathcal{T}^{-1}(\theta)\right|$ with assumed noise standard deviation σ, because the inputs $y^{(k)}_{\text{excite}}$ 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 $\sigma^{(k)}_{\text{excite}} \to \sigma = 0$, as observed above we have that $\hat{y}^{(k)}e^{\imath\hat{\phi}^{(k)}}$ approaches $\hat{y}^{(k)}_{\text{excite}}e^{\imath\hat{\phi}^{(k)}}$. 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. $\phi = \mathbf{0}$. Since in such case the magnitude $\left|ye^{\imath\phi}\right|$ is equal to the real component $\mathrm{Re}(ye^{\imath\phi}) = y$, 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

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

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

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

where $\eta^{(k)}_{\text{excite}}$ is again i.i.d. complex Gaussian noise with zero mean and standard deviation $\sigma^{(k)}_{\text{excite}}$. 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

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

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

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

being $\sigma^{(k)}_{\text{excite}}$ 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 $\mathcal{T}^{-1}\left(\hat{\theta}^{(k)}_{\text{excite}}\right)$, and thus its excited observation model agrees with (1).

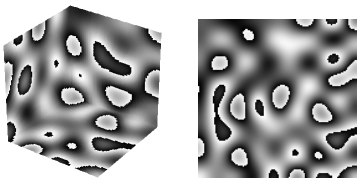


Fig. 3. Original phase ϕ used for the reconstruction experiments (black and white correspond to $-\pi$ and π, respectively).

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

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

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 $\eta^{(k)}_{\text{excite}}$ is chosen to be i.i.d. complex Gaussian noise with zero mean and variance

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

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

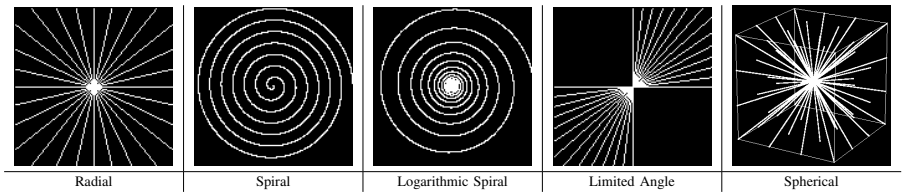


Fig. 4. Examples of different sampling trajectories. These trajectories define which k-space coefficients will be retained during the MR acquisition process.

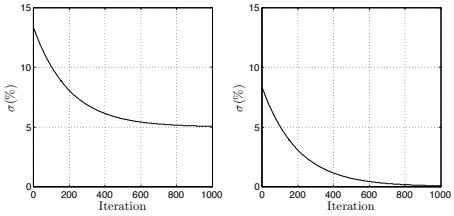


Fig. 5. Standard deviation σ_{excite} of the excitation noise (26) for noisy (left) and noise-free (right) data of parameters $\alpha = 1.01$, $\beta = 500$, and $\sigma = 5\%$.

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

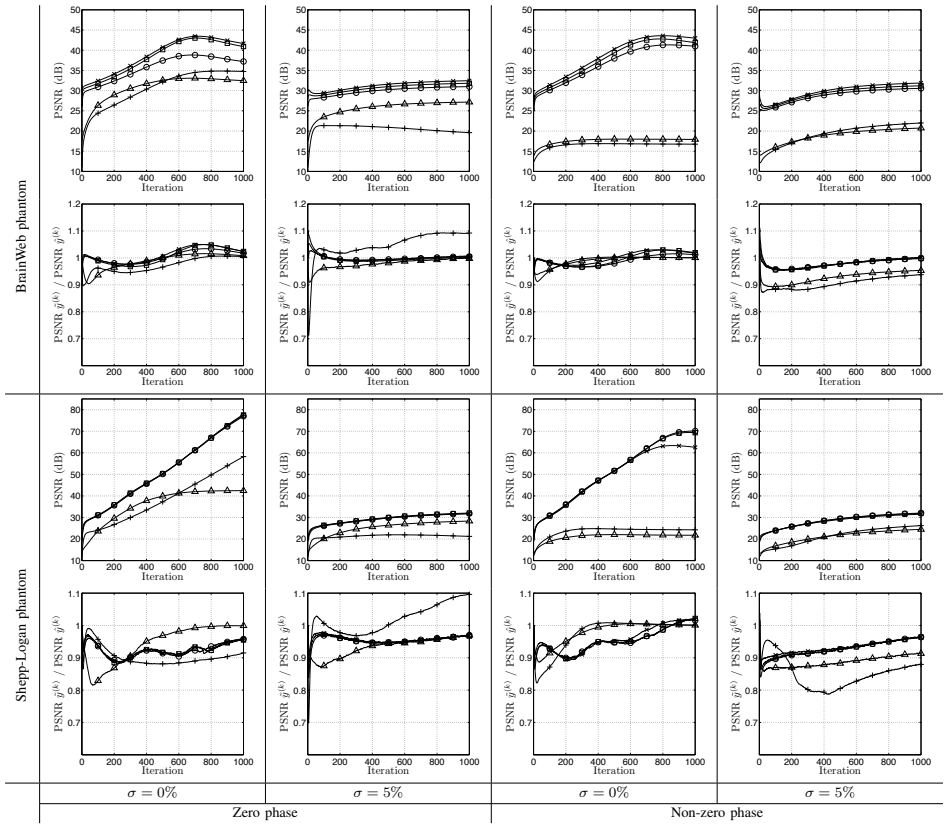


Fig. 6. PSNR progression for the iterative reconstruction of the noisy and noise-free BrainWeb having zero or non-zero phase. The plots in the top row illustrate the PSNR progressions of $\tilde{y}^{(k)}$, whereas the plots in the bottom row illustrate the progression of the ratio between the PSNR of $\tilde{y}^{(k)}$ and $\hat{y}^{(k)}$. The sampling trajectories are *Radial* ($\circ$), *Spiral* ($+$), *Logarithmic Spiral* ($\square$), *Limited Angle* ($\triangle$), and *Spherical* ($\times$). 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.

ACKNOWLEDGMENT

The authors would like to thank the Reviewers for their constructive and helpful comments. Additionally the authors wish to thank Jose V. Manjón and Pierrick Coupé for clearly documenting and distributing the source codes of their denoising algorithms [8], [9], [10], [11].

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.

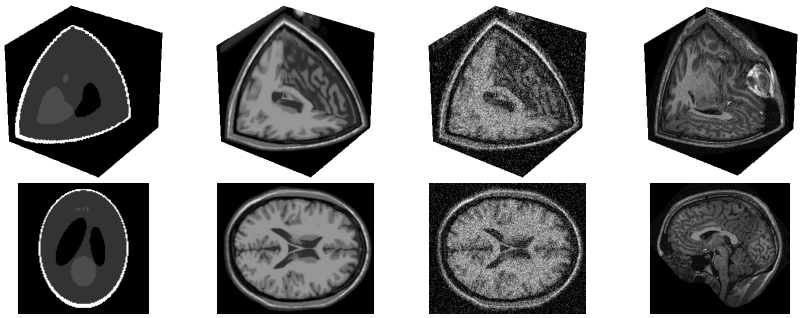
[32] Z. P. Liang and P. C. Lauterbur, *Principles of Magnetic Resonance Imaging: A Signal Processing Perspective*. Wiley-IEEE Press, October 1999.

[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.

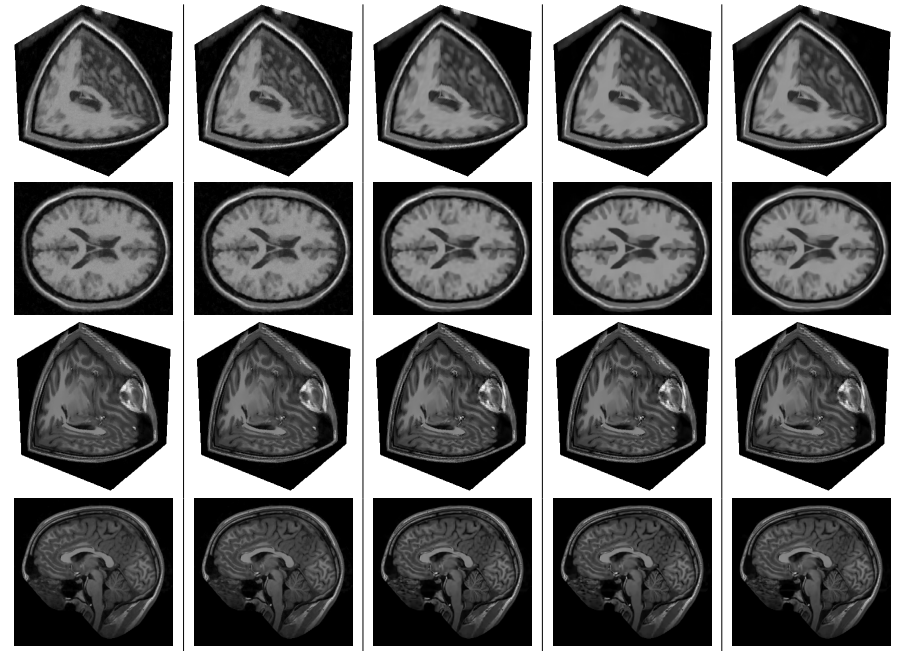


Fig. 8. From left to right, denoising results of the OB-NLM3D, OB-NLM3D-WM, ODCT3D, PRI-NLM3D, and the proposed BM4D filter applied to the BrainWeb phantom corrupted by i.i.d. Gaussian noise with standard deviation $\sigma = 15\%$ (top) and the OASIS phantom (bottom) corrupted by Rician noise with standard 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.

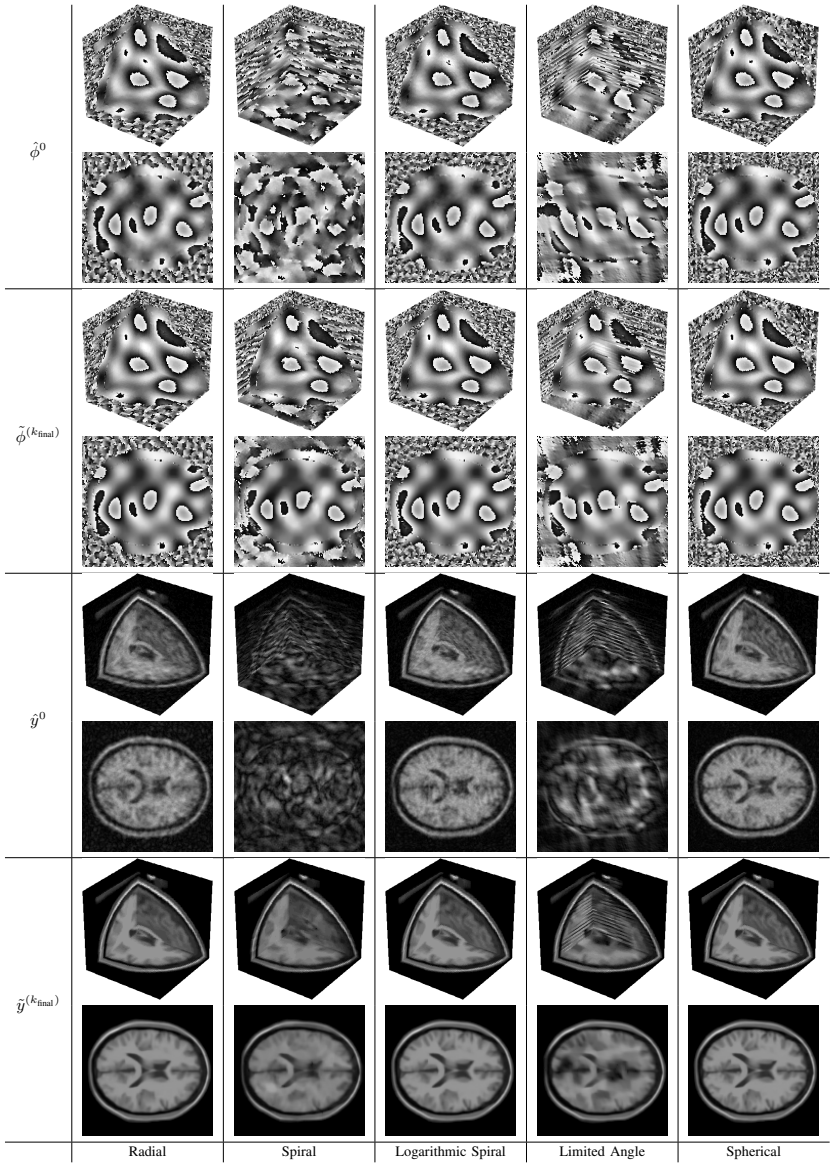


Fig. 9. Initial back-projections and final estimates of the magnitude and phase after $k_{\text{final}} = 1000$ iterations of the noisy reconstruction of the BrainWeb phantom ($\sigma = 5\%$) subsampled with ratio 30%. The original magnitude and phase volumes are shown in Fig. 7(b) and Fig. 3, respectively.

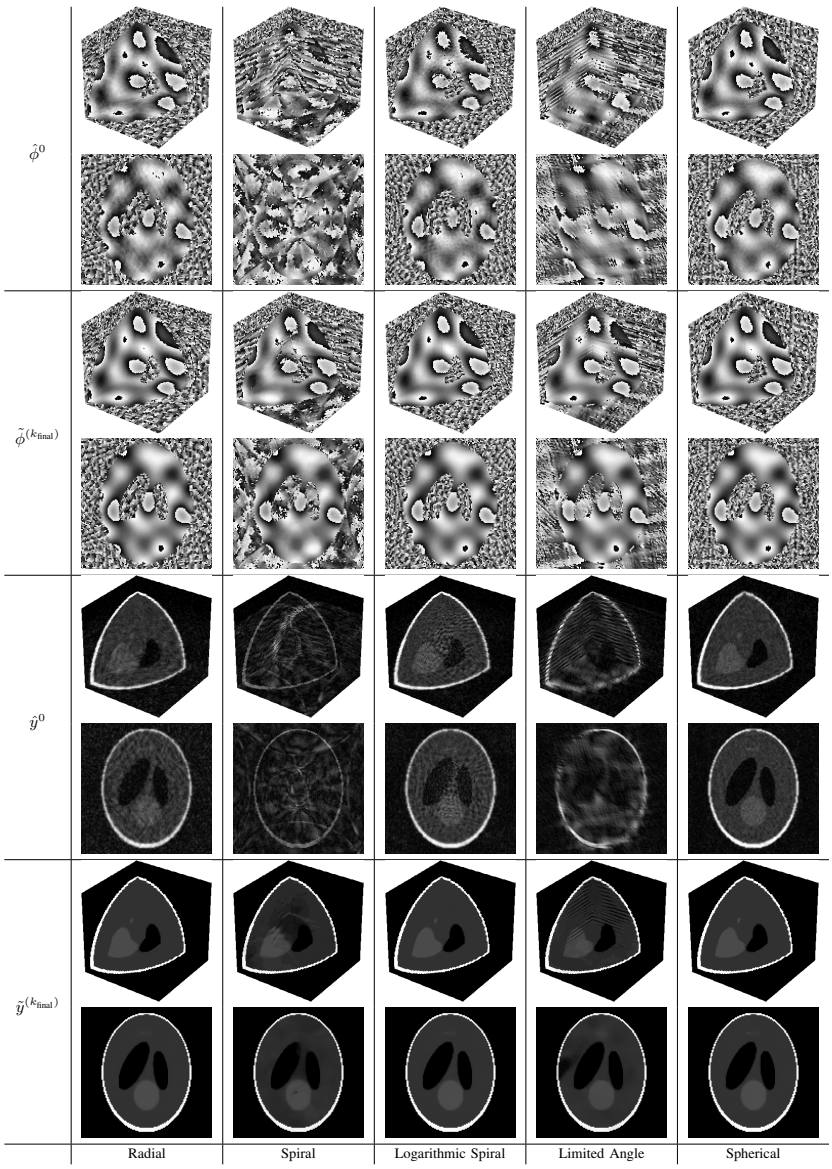


Fig. 10. Initial back-projections and final estimates of the magnitude and phase after $k_{\text{final}} = 1000$ iterations of the noisy reconstruction of the Shepp-Logan phantom ($\sigma = 5\%$) subsampled with ratio 30%. The original magnitude and phase volumes are shown in Fig. 7(b) and Fig. 3, respectively.

Publication V

M. Maggioni, E. Sánchez-Monge, and A Foi. Joint removal of random and fixed-pattern noise through spatiotemporal video filtering. *IEEE Transactions on Image Processing*, 23(10):4282–4296, Oct. 2014

IEEE TRANSACTIONS ON IMAGE PROCESSING, 2014 (preprint) 1

Joint Removal of Random and Fixed-Pattern Noise through Spatiotemporal Video Filtering

Matteo Maggioni, Enrique Sánchez-Monge, Alessandro Foi

***Abstract*—We propose a framework for the denoising of videos jointly corrupted by spatially correlated (i.e. non-white) random noise and spatially correlated fixed-pattern noise. Our approach is based on motion-compensated 3-D spatiotemporal volumes, i.e. a sequence of 2-D square patches extracted along the motion trajectories of the noisy video. First, the spatial and temporal correlations within each volume are leveraged to sparsify the data in 3-D spatiotemporal transform domain, and then the coefficients of the 3-D volume spectrum are shrunk using an adaptive 3-D threshold array. Such array depends on the particular motion trajectory of the volume, the individual power spectral densities of the random and fixed-pattern noise, and also the noise variances which are adaptively estimated in transform domain. Experimental results on both synthetically corrupted data and real infrared videos demonstrate a superior suppression of the random and fixed-pattern noise from both an objective and a subjective point of view.**

***Index Terms*—Video denoising, spatiotemporal filtering, fixed-pattern noise, power spectral density, adaptive transforms, thermal imaging.**

I. Introduction

DIGITAL videos may be degraded by several spatial and temporal corrupting factors which include but are not limited to noise, blurring, ringing, blocking, flickering, and other acquisition, compression or transmission artifacts. In this work we focus on the joint presence of random and fixed-pattern noise (FPN). The FPN typically arises in raw images acquired by focal plane arrays (FPA), such as CMOS sensors or thermal microbolometers, where spatial and temporal nonuniformities in the response of each photodetector generate a pattern superimposed on the image approximately constant in time. The spatial correlation characterizing the noise corrupting the data acquired by such sensors [1], [2], [3] invalidates the classic AWGN assumptions of independent and identically distributed (i.i.d.) –and hence white– noise.

The FPN removal task is prominent in the context of long wave infrared (LWIR) thermography and hyperspectral imaging. Existing denoising methods can be classified into reference-based (also known as calibration-based) or scene-based approaches. Reference-based approaches first calibrate the FPA using (at least) two homogeneous infrared targets, having different and known temperatures, and then linearly estimate the nonuniformities of the data [4], [5]. However, since the FPN slowly drifts in time, the normal operations of the camera need to be periodically interrupted to update the estimate which has become obsolete. Differently, scene-based approaches are able compensate the noise directly from the acquired data, by modeling the statistical nature of the FPN; this is typically achieved by leveraging nonlocal self-similarity and/or the temporal redundancy present along the direction of motion [6], [7], [8], [9], [10], [11].

We propose a scene-based denoising framework for the joint removal of random and fixed-pattern noise based on a novel observation model featuring two spatially correlated (non-white) noise components. Our framework, which we denote as RF3D, is based on motion-compensated 3-D spatiotemporal volumes characterized by local spatial and temporal correlation, and on a filter designed to sparsify such volumes in 3-D spatiotemporal transform domain leveraging the redundancy of the data in a fashion similar to [12], [13], [14], [15]. Particularly, the 3-D spectrum of the volume is filtered through a shrinkage operator based on a threshold array calculated from the motion trajectory of the volume and both from the individual power spectral densities (PSD) and the noise variances of the two noise components. The PSDs are assumed to be known, whereas the noise standard deviations are adaptively estimated from the noisy data. We also propose an enhancement of RF3D, denoted E-RF3D, in which the realization of the FPN is first progressively estimated using the data already filtered, and then subtracted from the subsequent noisy frames.

To demonstrate the effectiveness of our approach, we evaluate the denoising performance of the proposed method and the current state of the art in video and volumetric data denoising [13], [15] using videos corrupted by synthetically generated noise and also real LWIR therm sequences acquired with a FLIR Tau 320 microbolometer camera. We implement RF3D (and E-RF3D) as a two-stage filter: in each stage use the same multi-scale motion estimator to build the 3-D volumes but a different shrinkage operator for the filtering. Specifically, we use a hard-thresholding operator in the first stage and an empirical Wiener filter in the second. Let us remark that the proposed framework can be also generalized to other filtering strategies based on a separable spatiotemporal patch-based model.

The remainder of the paper is organized as follows. In

Matteo Maggioni and Alessandro Foi are with the Department of Signal Processing, Tampere University of Technology, Finland. Enrique Sánchez-Monge is with Noiseless Imaging Ltd, Tampere, Finland.

This work was supported in part by the Academy of Finland through the Academy Research Fellow 2011-2016 under Project 252547 and in part by the Tampere Graduate School in Information Science and Engineering, Tampere, Finland, and Tekes, the Finnish Funding Agency for Technology and Innovation (Decision 40081/14, Dnro 338/31/2014, Parallel Acceleration Y2).

Section II we formalize the observation model, and in Section III we analyze the class of spatiotemporal transform-domain filters. Section IV gives a description of the proposed denoising framework, whereas Section V discusses the modification required to implement the enhanced fixed-pattern suppression scheme. The experimental evaluation and the conclusions are eventually given in Section VI and Section VII, respectively.

II. Observation Model

We consider an observation model characterized by two spatially correlated noise components having distinctive PSDs defined with respect to the corresponding spatial frequencies. Formally, we denote a noisy video $z : X \times T \to \mathbb{R}$ as

$$z(\mathbf{x}, t) = y(\mathbf{x}, t) + \eta_{\text{RND}}(\mathbf{x}, t) + \eta_{\text{FPN}}(\mathbf{x}, t), \quad (1)$$

where $(\mathbf{x}, t) \in X \times T$ is a voxel of spatial coordinate $\mathbf{x} \in X \subseteq \mathbb{Z}^2$ and temporal coordinate $t \in T \subseteq \mathbb{Z}$, $y : X \times T \to \mathbb{R}$ is the unknown noise-free video, and $\eta_{\text{FPN}} : X \times T \to \mathbb{R}$ and $\eta_{\text{RND}} : X \times T \to \mathbb{R}$ denote a realization of the FPN and zero-mean random noise, respectively.

In particular, we model η_{RND} and η_{FPN} as colored Gaussian noise whose variance can be defined as

$$\begin{aligned} \text{var}\left\{ \mathcal{T}_{2\text{D}}\left(\eta_{\text{RND}}(\cdot, t)\right)(\boldsymbol{\xi}) \right\} &= \sigma^2_{\text{RND}}(\boldsymbol{\xi}, t) \\ &= \varsigma^2_{\text{RND}}(t)\, \Psi_{\text{RND}}(\boldsymbol{\xi}), \quad (2) \\ \text{var}\left\{ \mathcal{T}_{2\text{D}}\left(\eta_{\text{FPN}}(\cdot, t)\right)(\boldsymbol{\xi}) \right\} &= \sigma^2_{\text{FPN}}(\boldsymbol{\xi}, t) \\ &= \varsigma^2_{\text{FPN}}(t)\, \Psi_{\text{FPN}}(\boldsymbol{\xi}), \quad (3) \end{aligned}$$

where $\mathcal{T}_{2\text{D}}$ is a 2-D transform, such as the DCT, operating on $N \times N$ blocks, $\boldsymbol{\xi}$ belongs to the $\mathcal{T}_{2\text{D}}$ domain Ξ, σ^2_{RND} and σ^2_{FPN} are the time-variant PSDs of the random and fixed-pattern noise defined with respect to $\mathcal{T}_{2\text{D}}$; the time-variant PSDs can be separated into their normalized time-invariant counterparts $\Psi_{\text{RND}}, \Psi_{\text{FPN}} : \Xi \to \mathbb{R}$ and the corresponding time-variant scaling factors $\varsigma^2_{\text{RND}}, \varsigma^2_{\text{FPN}} : T \to \mathbb{R}$. We observe that the PSDs Ψ_{RND} and Ψ_{FPN} are known and fixed; moreover the random noise component η_{RND} is independent with respect to t, whereas the fixed-pattern noise component η_{FPN} is roughly constant in time, that is

$$\frac{\partial}{\partial t}\eta_{\text{FPN}}(\mathbf{x}, t) \approx 0. \quad (4)$$

The model (1) is much more flexible than the standard i.i.d. AWGN model commonly used in image and video denoising. In this paper we successfully use (1) to describe the raw output of a LWIR microbolometer array thermal camera; specifically, Fig. 1 and Fig. 2 show the PSDs of the random and fixed-pattern noise of video acquired by a FLIR Tau 320 camera. The power spectral densities in the figures are defined with respect to the global 2-D Fourier transform and the 8×8 2-D block DCT, respectively. As can be clearly seen from the figures, the two noise components are not white and instead are characterized by individual and nonuniform PSDs.

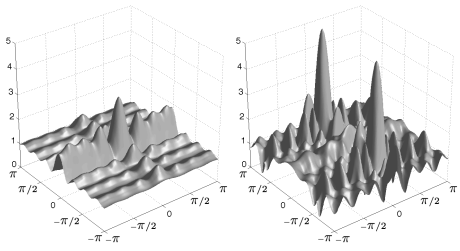


Fig. 1. Normalized root power spectral densities of the random (left) and fixed-pattern (right) noise components computed with respect to the global 2-D Fourier transform. The DC coefficient is located in the center (0,0) of the grid.

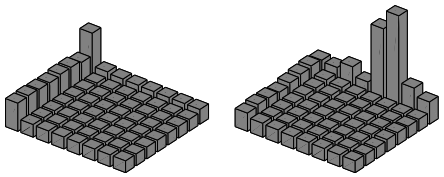


Fig. 2. Power spectral densities of the random (left) and fixed-pattern (right) noise components calculated with respect to the 2-D block DCT of size 8×8. The DC coefficient is located in the top corner.

III. Spatiotemporal Filtering

In this section we generally analyze the class of spatiotemporal video filters, and, in particular, those characteristics of spatiotemporal filtering that are essential to the proposed noise removal framework.

A. Related Work

Natural signals tend to exhibit high auto correlation and repeated patterns at different location within the data [16], thus significant interest has been given to image denoising and compression methods which leverage redundancy and self-similarity [17], [18], [19], [20], [21]. For example, in [18] each pixel estimate is obtained by averaging all pixels in the image within an adaptive convex combination, whereas in [12] self-similar patches are first stacked together in a higher dimensional structure called "group", and then jointly filtered in transform domain. Highly correlated data can be sparsely represented with respect to a suitable basis in transform domain [22], [23], where the energy of the noise-free signal can be effectively separated from that of the noise through coefficient shrinkage. Thus, self-similarity and sparsity are the foundations of modern image [18], [12], video [13], [20], [14], and volumetric data [24], [15] denoising filters.

For the case of video processing, self-similarity can be naturally found along the temporal dimension. In [25], [26], [14] it has been shown that natural videos exhibit a strong temporal smoothness, whereas the nonlocal spatial redundancy only provides a marginal contribution to the filtering quality

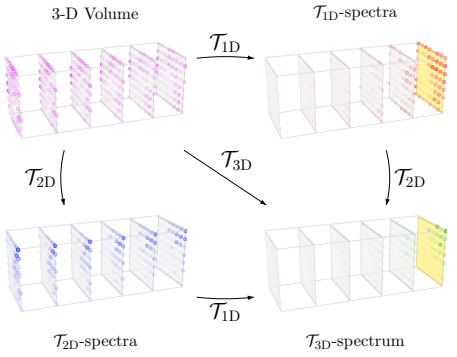


Fig. 3. Separable $\mathcal{T}_{3D}$ DCT transform applied to the 3-D volume illustrated in the top-left position. The magnitude of each 3-D element is proportional to its opacity. Whenever the temporal $\mathcal{T}_{1D}$ transform is applied, we highlight the 2-D temporal DC plane with a yellow background. Note that $\mathcal{T}_{1D}$-spectra is sparse outside the temporal DC plane, and $\mathcal{T}_{2D}$-spectra becomes sparser as we move away from the spatial DC coefficients (top-right corner of each block). Consequently, the energy of $\mathcal{T}_{3D}$-spectrum is concentrated around the spatial DC of the temporal DC plane.

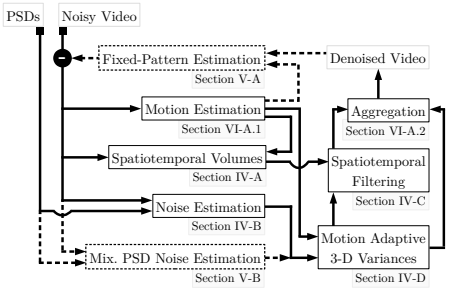


Fig. 4. Flowchart of random and fixed-pattern joint noise removal framework RF3D and the enhanced E-RF3D. A complete overview of RF3D is given in Section IV, while the modifications required to implement E-RF3D, illustrated as dashed lines, are described in Section V.

[14]. Methods that do not explicitly account motion information have also been investigated [27], [28], [29], [30], but motion artifacts might occur around the moving features of the sequence if the temporal nonstationarities are not correctly compensated. Typical approaches employ a motion estimation technique to first compensate the data and then apply the filtering along the estimated motion direction [31], [32], [14]. A proper motion estimation technique is required to overcome the imperfections of the motion model, computational constraints, temporal discontinuities (e.g., occlusions in the scene), and the presence of the noise [33].

In this work, we focus on spatiotemporal video filters, so that the peculiar correlations present in the spatial and temporal dimension can be leveraged to minimize filtering artifacts in the estimate [34].

B. Filtering in Transform Domain

The spatiotemporal volume is a sequence of 2-D blocks following a motion trajectory of the video, and thus, in a fashion comparable to the "group" in [12], is characterized by local spatial correlation within each block and temporal correlation along its third dimension. As in [12], [13], [14], [15], the filtering is formalized as a coefficient shrinkage in spatiotemporal transform domain after a separable linear transform is applied on the data to separate the meaningful part of the signal from the noise. We use an orthonormal 3-D transform $\mathcal{T}_{3D}$ composed by a 2-D spatial transform $\mathcal{T}_{2D}$ applied to each patch in the volume followed by a 1-D temporal transform $\mathcal{T}_{1D}$ applied along the third dimension.

The $\mathcal{T}_{1D}$ transform should be comprised of a DC (direct current) coefficient representing the mean of the data, and a number of AC (alternating current) coefficients representing the local changes within the data. The 2-D temporal DC plane obtained after the application of the 1-D temporal transforms along the third dimension of the volume is of particular interest, as it encodes the features shared among the blocks, and thus can be used to capture the FPN present in the spatiotemporal volume. Fig. 3 provides a schematic representation of the 3-D spectrum obtained after applying a $\mathcal{T}_{1D}$, $\mathcal{T}_{2D}$, and $\mathcal{T}_{3D}$ DCT transforms on a typical spatiotemporal 3-D volume. The magnitude of each spectrum coefficient is directly proportional to its opacity, thus coefficients close to zero are almost transparent. The 2-D temporal DC plane in $\mathcal{T}_{1D}$-spectra and $\mathcal{T}_{3D}$-spectrum is highlighted with a yellow background, whereas the spatial DC coefficients in $\mathcal{T}_{2D}$-spectra are located at the top-right corner of each 2-D spectrum. Thus, the 3-D DC coefficient of the $\mathcal{T}_{3D}$-spectrum is located at the top-right corner of the temporal DC plane. Note how the data is differently sparsified in the different spectrum: in $\mathcal{T}_{1D}$-spectra the energy is concentrated in the temporal DC plane, in $\mathcal{T}_{2D}$-spectra the energy is concentrated around each spatial DC coefficients, and consequently in $\mathcal{T}_{3D}$-spectrum the energy is concentrated around the spatial DC of the temporal DC plane.

The PSDs of the noise in (1) are defined with respect to the 2-D spatial transform $\mathcal{T}_{2D}$. For example, in Fig. 2 we show the root PSDs of the random and fixed-pattern noise, obtained from a 2-D DCT of size 8×8. These PSDs provide the variances of the two noise components within each 2-D block coefficients before the application of the 1-D temporal transform to the spatiotemporal volume. The analogies with the corresponding PSDs defined with respect to the 2-D Fourier transform can be appreciated by referring to Fig. 1.

IV. Joint Noise Removal Framework

In this section, we describe the proposed RF3D framework for the joint removal of random and fixed-pattern noise. The RF3D works as follows: first a 3-D spatiotemporal volume is built for a specific position in the video (Section IV-A), and then the noise standard deviations are estimated from a set of frames (Section IV-B). Finally, the 3-D volume is filtered in

spatiotemporal transform domain (Section IV-C) using adaptive shrinkage coefficients (Section IV-D). A flowchart of the framework is illustrated in Fig. 4. This generic algorithm and its various applications are the object of a patent application [35].

The model (1) is simplified by (2) and (3), where we assume that the PSDs of η_{RND} and η_{FPN} are fixed modulo normalization with the corresponding scaling factors ς^2_{RND} and ς^2_{FPN}. As a result, the PSDs do not need to be periodically estimated, but can be treated as known parameters. During the filtering, such parameters are scaled with the scaling factors to obtain the actual PSDs of the noise components corrupting the video. Further, we assume that the time-variant scaling factors of (2) and (3) vary slowly with time, so that they can be treated as constant within the local temporal extent of each spatiotemporal volume. Formally, we define the following conditions on the partial derivatives of ς_{RND} and ς_{FPN} with respect to time:

$$\frac{\partial}{\partial t}\varsigma_{\text{RND}}\left(t\right) \approx 0, \qquad \frac{\partial}{\partial t}\varsigma_{\text{FPN}}\left(t\right) \approx 0. \tag{5}$$

A. Spatiotemporal Volumes

The proposed framework is based on motion-compensated 3-D spatiotemporal volumes composed by a sequence of 2-D blocks following a motion trajectory of the video [12], [13], [14]. Let $B(\mathbf{x}, t)$ be a 2-D $N \times N$ block extracted from the noisy video z, whose top-right corner is located at the 3-D coordinate $(\mathbf{x}, t)$. Formally, a motion trajectory corresponding to a (reference) block $B(\mathbf{x}_R, t_R)$ is a sequence of coordinates defined as

$$\Gamma(\mathbf{x}_R, t_R) = \Big\{(\mathbf{x}_i, t_i)\Big\}_{i=h^-}^{h^+}, \tag{6}$$

where $\mathbf{x}_i$ is the spatial location of the block within the frame at time t_i with $i = h^-, \ldots, h^+$, and each voxel is consecutive in time with respect to the precedent, i.e. $t_{i+1} - t_i = 1 \ \forall i$. Note that in (6) we do not restrict the reference coordinate $(\mathbf{x}_R, t_R)$ to occupy a predefined position in the sequence, thus the trajectory can be grown backward and/or forward in time, i.e. $t_{h^-} \leq t_R \leq t_{h^+}$. Finally, we call $H = t_{h^+} - t_{h^-}$ the temporal extent of the volume.

Assuming that the trajectory for any given reference block $B(\mathbf{x}_R, t_R)$ is known, we can easily define the corresponding motion-compensated 3-D spatiotemporal volume as

$$\mathcal{V}(\mathbf{x}_R, t_R) = \Big\{B(\mathbf{x}_i, t_i) : (\mathbf{x}_i, t_i) \in \Gamma(\mathbf{x}_R, t_R)\Big\}. \tag{7}$$

The trajectories can be either known *a-priori*, or built in-loop, e.g., by concatenating motion vectors along time. However, let us stress that the motion estimation technique needs to be tolerant to noise [33], [29], [32], [14].

In Fig. 5, we show a schematic illustration of a spatiotemporal volume (7). In the figure, the reference block $B(\mathbf{x}_R, t_R)$ is shown in blue and occupies the middle position, the other blocks of the volume are shown in grey.

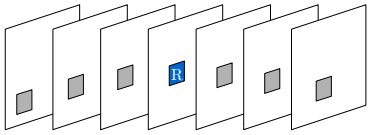


Fig. 5. Schematic illustration of a spatiotemporal volume. The blocks of the volume are grey with the exception of the reference block "R", which is blue.

B. Noise Estimation

The noise can be estimated leveraging the fact that the FPN is roughly constant in time (4): thus a spatial high-pass filtering of the video captures both random and fixed-pattern noise components, whereas a temporal high-pass filter captures only the random one.

The overall PSD σ^2 of the random and fixed-pattern noise is simply defined as the sum of (2) and (3)

$$\sigma^2(\boldsymbol{\xi}, t) = \varsigma^2_{\text{RND}}\left(t\right)\Psi_{\text{RND}}\left(\boldsymbol{\xi}\right) + \varsigma^2_{\text{FPN}}\left(t\right)\Psi_{\text{FPN}}\left(\boldsymbol{\xi}\right), \tag{8}$$

being Ψ_{FPN} and Ψ_{RND} the only known terms of the equation.

Firstly we estimate σ as the median absolute deviation (MAD) [36], [37] of the $\mathcal{T}_{2\text{D}}$-coefficients of all the blocks having temporal coordinates within $[t_{h^-}, t_{h^+}] \ni t$ as

$$\hat{\sigma}(\boldsymbol{\xi}, t) = \frac{1}{0.6745} \cdot \underset{\substack{\mathbf{x}\in X \\ t_{h^-}\leq\tau\leq t_{h^+}}}{\text{MAD}} \Big(\mathcal{T}_{2\text{D}}\Big(B\left(\mathbf{x}, \tau\right)\Big)(\boldsymbol{\xi})\Big), \tag{9}$$

because $\mathcal{T}_{2\text{D}}$ also embeds some high-pass filters and both ς_{RND} and ς_{FPN} are slowly varying in time (5). Then, we estimate σ_{RND} through a similar MAD on a temporal high-pass version of the video, obtained by differentiating consecutive blocks:

$$\hat{\sigma}_{\text{RND}}(\boldsymbol{\xi}, t) = \frac{1}{0.6745} \cdot \underset{\substack{\mathbf{x}\in X \\ t_{h^-}\leq\tau<t_{h^+}}}{\text{MAD}} \Big(\mathcal{T}_{2\text{D}}\Big(B\left(\mathbf{x}, \tau+1\right)\Big)(\boldsymbol{\xi}) - \mathcal{T}_{2\text{D}}\Big(B\left(\mathbf{x}, \tau\right)\Big)(\boldsymbol{\xi})\Big). \tag{10}$$

We recognize that the MAD scaled by the usual factor 0.6745 (from the inverse cumulative Gaussian distribution at 3/4) is designed for Gaussian data. Even though in the practice the distribution of the noise in (1) may deviate from a Gaussian, the MAD/0.6745 is nevertheless a viable estimator for (9) and (10) because it is not applied directly on the observed data but on the $\mathcal{T}_{2\text{D}}$ transform coefficients. Each transform coefficient is obtained as a linear combination involving many data samples (e.g., 64 samples when using a linear 8×8 $\mathcal{T}_{2\text{D}}$), a "Gaussianization" kicks in, analogous to the central limit theorem. This makes the MAD/0.6745 an unbiased estimator of the standard deviation of each individual subband of transformed coefficients. In other words, we can safely use the MAD to estimate the root-PSD.

According to (2) and (3), σ^2_{RND} and σ^2_{FPN} must be respectively equal to Ψ_{RND} and Ψ_{FPN} modulo the non-negative scaling factors ς^2_{RND} and ς^2_{FPN}, and as can be seen from (8) an analogous condition applies to σ^2. However, up to this point neither $\hat{\sigma}^2$ nor $\hat{\sigma}^2_{\text{RND}}$ are guaranteed to satisfy such

scaling property. To find such scaling factors, we resort to the following non-negative least-squares optimization, whose solutions $\hat{\varsigma}^2_{\text{RND}}(t)$ and $\hat{\varsigma}^2_{\text{FPN}}(t)$ are defined as

$$\begin{aligned}\underset{\substack{\varsigma^2_{\text{RND}}(t)\geq 0\\ \varsigma^2_{\text{FPN}}(t)\geq 0}}{\arg\min}\Bigg\{ \sum_{\boldsymbol{\xi}\in\Xi} \Big(&\Psi_{\text{RND}}(\boldsymbol{\xi})\,\varsigma^2_{\text{RND}}(t) \\ &+ \Psi_{\text{FPN}}(\boldsymbol{\xi})\,\varsigma^2_{\text{FPN}}(t) - \hat{\sigma}^2(\boldsymbol{\xi},t)\Big)^2 w_1^2(\boldsymbol{\xi}) \\ &+ \sum_{\boldsymbol{\xi}\in\Xi} \Big(\Psi_{\text{RND}}(\boldsymbol{\xi})\,\varsigma^2_{\text{RND}}(t) - \hat{\sigma}^2_{\text{RND}}(\boldsymbol{\xi},t)\Big)^2 w_2^2(\boldsymbol{\xi})\Bigg\},\end{aligned} \tag{11}$$

where $w_1, w_2 : \Xi \to \mathbb{R}$ give different weights to each coefficients fed to (9) and (10), and in practice can be used as logical operators to select linearly independent high-frequency coefficients in the $\mathcal{T}_{\text{2D}}$ domain.

C. Spatiotemporal Filtering

During the spatiotemporal filtering, the volume (7) is first transformed from its voxel representation to a new domain via a separable linear transform $\mathcal{T}_{\text{3D}}$, then a shrinkage operator Υ such as the hard thresholding modifies the magnitude of the spectrum coefficients to attenuate the noise. This strategy leverages the sparsification of the 3-D volume induced by $\mathcal{T}_{\text{3D}}$ as illustrated in Fig. 3. An estimate of the noise-free volume is eventually obtained after inverting the transform $\mathcal{T}_{\text{3D}}$ on the thresholded spectrum. The complete process can be formally defined as

$$\hat{\mathcal{V}}(\mathbf{x}_R, t_R) = \mathcal{T}_{\text{3D}}^{-1}\Big(\Upsilon\Big(\mathcal{T}_{\text{3D}}\big(\mathcal{V}(\mathbf{x}_R, t_R)\big)\Big)\Big), \tag{12}$$

which in turn generates individual estimates of each noise-free patch in the volume. This strategy is referred to as collaborative filtering, and a deeper analysis of its rationale can be found in [12], [13], [19], [14].

D. Motion-Adaptive 3-D Spectrum Variances

The shrinkage operator Υ in (12) modulates the applied filtering strength relying on the variances $s^2_{\mathbf{x}_R,t_R}(\boldsymbol{\xi},\vartheta)$ of the $\mathcal{T}_{\text{3D}}$-spectrum coefficients, where $\vartheta \in \{1,\dots,H\} \subset \mathbb{N}$ indicates the coefficient position with respect to the $\mathcal{T}_{\text{1D}}$ spectrum, $\vartheta = 1$ corresponding to the temporal DC. Observe that $s^2_{\mathbf{x}_R,t_R}$ constitutes a 3-D array of variances. Each $s^2_{\mathbf{x}_R,t_R}(\boldsymbol{\xi},\vartheta)$ depends on the $\mathcal{T}_{\text{2D}}$-PSDs (8) of each block $B(\mathbf{x}_i,t_i)$ in the volume (7) through the $\mathcal{T}_{\text{1D}}$ transform. Since both ς_{RND} and ς_{FPN} are slowly varying in time, we can use their respective estimates at the time t_R for the whole volume $\mathcal{V}(\mathbf{x}_R,t_R)$. However, due to the FPN, the relative spatial alignment of the blocks has an impact on the variance of the $\mathcal{T}_{\text{3D}}$ spectrum coefficients and thus needs to be taken into account for the design of the threshold coefficients.

To understand this phenomenon, let us consider the following two extreme cases. In one case all blocks are perfectly overlapping, i.e. they share the same spatial position $\mathbf{x}_i$ for all t_i in (7), such as when no motion is detected. Thus the FPN component, being the same across all blocks, accumulates through averaging in the 2-D temporal DC plane of the 3-D volume spectrum, shown in yellow in Fig. 3. For this reason the variances of temporal DC plane and AC coefficients are different:

$$\begin{aligned} s^2_{\mathbf{x}_R,t_R}(\boldsymbol{\xi},1) &= \varsigma^2_{\text{RND}}(t_R)\Psi_{\text{RND}}(\boldsymbol{\xi}) + H\varsigma^2_{\text{FPN}}(t_R)\Psi_{\text{FPN}}(\boldsymbol{\xi}),\\ s^2_{\mathbf{x}_R,t_R}(\boldsymbol{\xi},\vartheta) &= \varsigma^2_{\text{RND}}(t_R)\Psi_{\text{RND}}(\boldsymbol{\xi}),\end{aligned} \tag{13}$$

with $\vartheta \in \{2,\dots,H\}$. In the other extreme case all blocks have different spatial positions and their relative displacement is such that the FPN exhibits uncorrelated patterns over different blocks. Thus, restricted to the volume, the FPN behaves just like another random component and the variances of the coefficients can be simply obtained as

$$s^2_{\mathbf{x}_R,t_R}(\boldsymbol{\xi},\vartheta) = \varsigma^2_{\text{RND}}(t_R)\Psi_{\text{RND}}(\boldsymbol{\xi}) + \varsigma^2_{\text{FPN}}\Psi_{\text{FPN}}(\boldsymbol{\xi}), \tag{14}$$

for all $\vartheta \in \{1,\dots,H\}$.

We stress that the variances of the 3-D spectrum coefficients depend not only on the two PSDs and on the temporal extent H of the spatiotemporal volume, but also on the relative spatial alignment of the blocks within the volume, on the temporal position of the coefficients within the 3-D spectrum, and on the unknown covariance matrices of the overlapping blocks which however are impracticable to compute. Nevertheless we resort to a formulation that interpolates (13) and (14), approximating all the intermediate cases for which any number of blocks in the volume is aligned or partially aligned with any of the others.

For a spatiotemporal volume of temporal extent H, let $L_h \leq H$, with $1 \leq h \leq H$, be the number of blocks sharing the same spatial coordinates as the h-th block in the volume, and let $L = \max_{1\leq h\leq H}\{L_h\}$, with $1 \leq L \leq H$, denote the maximum number of perfectly overlapping blocks. With this, we approximate the variances of the 3-D spatiotemporal coefficients by interpolating (13) and (14) with respect to L as

$$\begin{aligned}\hat{s}^2_{\mathbf{x}_R,t_R}(\boldsymbol{\xi},1) &= \hat{\varsigma}^2_{\text{RND}}(t_R)\Psi_{\text{RND}}(\boldsymbol{\xi}) \\ &\quad + \frac{L^2+H-L}{H}\hat{\varsigma}^2_{\text{FPN}}(t_R)\Psi_{\text{FPN}}(\boldsymbol{\xi}),\end{aligned} \tag{15}$$

$$\begin{aligned}\hat{s}^2_{\mathbf{x}_R,t_R}(\boldsymbol{\xi},\vartheta) &= \hat{\varsigma}^2_{\text{RND}}(t_R)\Psi_{\text{RND}}(\boldsymbol{\xi}) \\ &\quad + \left[1 - \frac{L(L-1)}{H(H-1)}\right]\hat{\varsigma}^2_{\text{FPN}}(t_R)\Psi_{\text{FPN}}(\boldsymbol{\xi}),\end{aligned} \tag{16}$$

with $\vartheta \in \{2,\dots,H\}$. By construction, the variances (15) and (16) reduce to the exact formulae (13) for $L = H$ and to (14) for $L = 1$, but observe that (15) is also exact in the configuration where L blocks are perfectly overlapping and the other $H - L$ are completely displaced. In order to attain exact results in every configuration, (15) and (16) should have taken into account the basis coefficients of the $\mathcal{T}_{\text{1D}}$ temporal transform as well as the spatiotemporal position of the volume coefficients. The chosen formula (16) is such that the total $\mathcal{T}_{\text{2D}}$ noise spectrum, given by the sum of (15) with $H-1$ times (16), is the same for all values of L and is equal to H times (14). Other approximate formulae are possible.

V. Enhanced Fixed-Pattern Suppression

In this section, we discuss the enhanced noise removal framework E-RF3D. Leveraging the fact that the fixed-pattern noise component varies slowly with time (4), it is possible to exploit its actual realization, i.e. the *fixed pattern* (FP), in a progressive fashion. In particular, the FP is first estimated from the noise that has been removed during previous filtering, and then subtracted from the following noisy frames to ease the denoising task (Section V-A). Consequently, the PSDs and the noise standard deviation of the data after the subtraction of the FP are updated (Section V-B). The modifications required to implement E-RF3D are illustrated as dashed lines in Fig. 4.

A. Fixed-Pattern Estimation

According to (1) and assuming that $\hat{y}$ is a good estimate of y, the noise realization at any position $(\mathbf{x}, t) \in X \times T$ can be estimated as

$$\hat{\eta}_{\text{FPN}}(\mathbf{x}, t) + \hat{\eta}_{\text{RND}}(\mathbf{x}, t) = z(\mathbf{x}, t) - \hat{y}(\mathbf{x}, t). \tag{17}$$

Since the FPN component η_{FPN} is assumed to be time-invariant within any short temporal extent (4), an estimate $\hat{\eta}_{\text{FPN}}(\mathbf{x}, t)$ of the FP can be simply obtained by averaging the noise residuals (17) of the previous $M(t) \in \mathbb{N}$ frames as

$$\hat{\eta}_{\text{FPN}}(\mathbf{x}, t) = \frac{1}{M(t)} \sum_{\tau=t-M(t)-1}^{t-1} \Big(z(\mathbf{x}, \tau) - \hat{y}(\mathbf{x}, \tau) \Big), \tag{18}$$

for every position $\mathbf{x} \in X$ and time $t \in T$. Furthermore, if we assume that our estimate of the video is perfect, i.e. $\hat{y} = y$, then

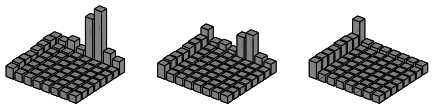
$$\hat{\eta}_{\text{FPN}}(\mathbf{x}, t) = \eta_{\text{FPN}}(\mathbf{x}, t) + \bar{\eta}_{\text{RND}}(\mathbf{x}, t), \tag{19}$$

where $\bar{\eta}_{\text{RND}}$ is an average random component which has the same distribution and spatial correlation of $\eta_{\text{RND}}/\sqrt{M(t)}$. In this case, (18) is unbiased:

$$\mathbb{E}\Big\{ \hat{\eta}_{\text{FPN}}(\mathbf{x}, t) \Big\} = \eta_{\text{FPN}}(\mathbf{x}, t).$$

The number of frames $M(t)$ in (18) can be adjusted in different manners. In this work, we empirically set $M(t)$ to be approximately proportional to $\hat{\varsigma}^2_{\text{RND}}(t)/\hat{\varsigma}^2_{\text{FPN}}(t)$. Thus, $M(t)$ adapts conveniently to the current noise characteristics by balancing the accuracy of (18) with respect to its variance, which is proportional to $\hat{\varsigma}^2_{\text{RND}}(t)/M(t)$. Note that the estimation of the FP is performed continuously during denoising in order to adapt to possible changes (drift) in the FP component.

Since $\hat{y}$ is never perfectly identical to y, (17) may contain structures belonging to the noise-free signal. This is particularly problematic whenever the video is stationary, because the static image content may accumulate into the FP (18). To counteract the consequent risks of fading and/or ghosting in the denoised signal, we select only those frames where motion is present. In particular, we use the displacement of the blocks between consecutive frames, since this information is readily available from (6): if the absolute mean displacement exceeds a certain threshold, we reckon that there is enough motion between the frames which can thus be used for the FP estimation.



$\gamma = 0.0$ $\gamma = 0.5$ $\gamma = 1.0$

Fig. 6. Mixture of power spectral densities Ψ_{FPNnew} (20) describing the updated FPN component after the fixed-pattern subtraction. The power spectral densities are computed with respect to the 2-D DCT transform $\mathcal{T}_{\text{2D}}$ of size 8×8 and three different values for γ.

B. Noise Estimation with Mixed Power Spectral Density

We observe from (19) that the FP estimate (18) is still corrupted by an average random component distributed as $\eta_{\text{RND}}/\sqrt{M(t)}$. Thus, after subtraction of $\hat{\eta}_{\text{FPN}}(\cdot, t)$ from $z(\cdot, t)$, a new estimation of the standard deviation and the PSD of the updated FPN component becomes necessary.

Firstly, we model the PSD of the updated FPN Ψ_{FPNnew} as a convex combination of the original PSDs Ψ_{RND} and Ψ_{FPN}

$$\Psi_{\text{FPNnew}}(\boldsymbol{\xi}, t) = \gamma(t) \Psi_{\text{RND}}(\boldsymbol{\xi}) + \Big(1 - \gamma(t)\Big) \Psi_{\text{FPN}}(\boldsymbol{\xi}), \tag{20}$$

where the parameter $\gamma \in [0, 1]$ determines the contributions of the original PSDs. In Fig. 6 we present few PSDs combinations obtained with different values of γ: obviously, at the extreme values $\gamma = 1$ and $\gamma = 0$ (20) reduces to the original Ψ_{RND} and Ψ_{FPN}, respectively.

Secondly, similar to (11), we estimate the scaling factors of the mixed PSDs as the solutions $\hat{\varsigma}^2_{\text{RND}}(t)$, $\hat{\varsigma}^2_{\text{FPNmix}}(t)$, and $\hat{\varsigma}^2_{\text{RNDmix}}(t)$ of the non-negative least-squares problem

$$\begin{aligned} \underset{\substack{\hat{\varsigma}^2_{\text{RND}}(t) \geq 0 \\ \hat{\varsigma}^2_{\text{FPNmix}}(t) \geq 0 \\ \hat{\varsigma}^2_{\text{RNDmix}}(t) \geq 0}}{\arg\min} \Bigg\{ & \sum_{\boldsymbol{\xi} \in \Xi} \Big(\Psi_{\text{RND}}(\boldsymbol{\xi}) \, \hat{\varsigma}^2_{\text{RND}}(t) + \Psi_{\text{FPN}}(\boldsymbol{\xi}) \, \hat{\varsigma}^2_{\text{FPNmix}}(t) \\ & + \Psi_{\text{RND}}(\boldsymbol{\xi}) \, \hat{\varsigma}^2_{\text{RNDmix}}(t) - \hat{\sigma}^2(\boldsymbol{\xi}, t) \Big)^2 w_1^2(\boldsymbol{\xi}) \\ & + \sum_{\boldsymbol{\xi} \in \Xi} \Big(\Psi_{\text{RND}}(\boldsymbol{\xi}) \, \hat{\varsigma}^2_{\text{RND}}(t) - \hat{\sigma}^2_{\text{RND}}(\boldsymbol{\xi}, t) \Big)^2 w_2^2(\boldsymbol{\xi}) \Bigg\}, \end{aligned} \tag{21}$$

where $\hat{\sigma}$ and $\hat{\sigma}_{\text{RND}}$ are obtained from the MAD of the high-frequency coefficients scaled by the weights $w_1, w_2 : \Xi \to \mathbb{R}$ as in (9) and (11). The optimization (21) aims to find the best non-negative solutions in the least-squares sense for the updated scaling factors using their definition (2) and (3). Note that the updated $\hat{\varsigma}_{\text{FPNnew}}(t)$ can be simply obtained from (21) as

$$\hat{\varsigma}^2_{\text{FPNnew}}(t) = \hat{\varsigma}^2_{\text{FPNmix}}(t) + \hat{\varsigma}^2_{\text{RNDmix}}(t).$$

Lastly, we compute the updated PSD (20) using a parameter γ defined as

$$\gamma(t) = \frac{\hat{\varsigma}^2_{\text{RNDmix}}(t)}{\hat{\varsigma}^2_{\text{FPNmix}}(t) + \hat{\varsigma}^2_{\text{RNDmix}}(t)}.$$

Note also that the updated Ψ_{FPNnew} and $\hat{\varsigma}_{\text{FPNnew}}$ are used for computing the adaptive threshold array (15)–(16) in place of Ψ_{FPN} and $\hat{\varsigma}_{\text{FPN}}$, respectively.

Fig. 7. Frames from the noise-free sequences *Foreman* (left) and *Miss America* (right).

VI. Experiments

We compare the filtering results of RF3D and E-RF3D against those obtained using the same spatiotemporal filter but with different *a-priori* assumptions on the observation model:

- *WR*: data corrupted by one additive white random noise component and no FPN component. In this case, $\hat{\varsigma}_{\text{RND}}$ reduces to a weighted average of (9) and (10) over Ξ, because in the non-negative least-squares minimization (11) we assume $\varsigma_{\text{FPN}} = 0$ and $\Psi_{\text{RND}}(\boldsymbol{\xi}) = 1$ for all $\boldsymbol{\xi} \in \Xi$.
- *CR*: data corrupted by one additive colored random noise component and no FPN component. The PSD of such noise is assumed equal to
$$\frac{\varsigma^2_{\text{RND}}\Psi_{\text{RND}} + \varsigma^2_{\text{FPN}}\Psi_{\text{FPN}}}{\varsigma^2_{\text{RND}} + \varsigma^2_{\text{FPN}}},$$
thus treating the FPN as another random component. Again, $\hat{\varsigma}_{\text{RND}}$ reduces to a weighted average of (9) and (10) over Ξ, because we assume $\varsigma_{\text{FPN}} = 0$.
- *WRWF*: data corrupted by two additive white noise components, namely random and fixed-pattern noise, with PSDs assumed as $\Psi_{\text{RND}}(\boldsymbol{\xi}) = \Psi_{\text{FPN}}(\boldsymbol{\xi}) = 1$ for all $\boldsymbol{\xi} \in \Xi$. Under this assumption, $\hat{\varsigma}_{\text{RND}}$ and $\hat{\varsigma}_{\text{FPN}}$ are given by (11).

Each of these simplified –and rough– assumptions reduce RF3D to an elementary algorithm that is unable to deal with the specific features of the actual noise model at hand. In particular, under the *WR* and *CR* assumptions, the FPN component is ignored and thus the filter is not able to account for the possible accumulation of FPN in the DC plane of the 3-D spectrum, which may hence remain unfiltered. Conversely, *WRWF* does model both the RND and FPN but ignores the spatial correlations that exist in the two noise components; thus filtering faces a particularly serious compromise between preservation of details and attenuation of noise. Additionally, we test the denoising performances of the state of the art in video and volumetric data denoising, namely V-BM3D [13] and BM4D [15], which are however designed for AWGN or, equivalently, for the WR assumption with i.i.d. Gaussian noise having standard deviation σ_{AWGN}.

In our experiments both synthetically corrupted sequences and real LWIR thermography data are considered. The objective denoising quality is measured by the peak signal-to-noise ratio (PSNR) of the estimate $\hat{y}$

$$10\log_{10}\left(\frac{I^2_{\max}|X||T|}{\sum_{\mathbf{x}\in X, t\in T}\left(\hat{y}(\mathbf{x},t) - y(\mathbf{x},t)\right)^2}\right),$$

where $I_{\max}$ is the maximum intensity value (peak) of the signal, y is the noise-free data, and $|X|$, $|T|$ are the cardinality of X and T, respectively. The data is hereafter considered to be in the range $[0, 255]$, i.e. $I_{\max} = 255$. We consider the standard sequences *Foreman*, *Coastguard*, *Miss America*, and *Flower Garden* corrupted as in (1) with different combinations of ς_{RND} and ς_{FPN}. In Fig. 7, we show two noise-free frames of *Foreman* and *Miss America*.

The remainder of this section is organized as follows. In Section VI-A we discuss the implementation details, parameter settings, and computational complexity of the proposed denoiser; in Section VI-B we present the denoising results for synthetic data; then, in Section VI-C, we show the denoising results of real thermography sequences to demonstrate that the proposed model (1) can appropriately describe the output of LWIR imagers.

A. Implementation Details

1) Motion Estimation: The proposed framework is relatively independent from the particular strategies used for the motion estimation. In our implementation, we use a coarse-to-fine two-scale motion estimator. First the sequence is downsampled by a factor of two; then the motion trajectories are computed using a fast diamond search [38] where the distance function is defined as the ℓ_2-norm difference of blocks of size $N \times N$, which thus cover an image area two times larger than that at the original resolution. Note that the downsampling increases the signal-to-noise ratio, and thus makes the motion estimation less impaired by noise. Finally, the found motion trajectories are refined on the full-resolution video using the same search process. For the refinement we employ a penalization term in the distance functional [14] to promote the matching of the blocks at the position predicted within the coarser scale.

2) Two-Stage Filtering: Similar to other algorithms [12], [13], [14], [15], we employ two cascading stages which differ for the particular shrinkage operator Υ (12): specifically we use a hard-thresholding operator in the first stage and an empirical Wiener filter in the second. The hard-thresholding stage is intended to provide a basic estimate which will serve as a pilot for the Wiener-filtering stage and uses an adaptive threshold array equal to the square root of the 3-D variances $s^2_{\mathbf{x}_R,t_R}$ scaled by a constant factor $\lambda_{3\text{D}}$ [22], [12]. In both stages, the estimates of volumes are obtained after applying the inverse 3-D transform on their thresholded spectra, and then are returned in their original location. Overlapping estimates are finally aggregated through an adaptive convex combination using (15)–(16) as in [12]. This implementation can be interpreted either as the V-BM3D algorithm [13] with the block matching performed only along the temporal dimension, or as the V-BM4D algorithm [14] without the 4-D nonlocal grouping.

3) PSD Normalization: Without loss of generality, both PSDs Ψ_{RND} and Ψ_{FPN} are normalized with respect to their highest frequency coefficient. In Fig. 2, the highest frequency coefficients are located at the bottom corner, diametrically opposite to the DC coefficients. Observe in the figure that the magnitude of the highest-frequency coefficients is among the

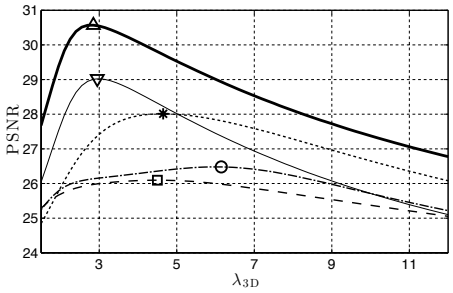


Fig. 8. Average PSNR (dB) obtained by WR (dashed line), CR (dot-dashed line), WRWF (dotted line), RF3D (thin solid line), and E-RF3D (thick solid line) as a function of the threshold factor λ_{3D}. The markers denote the global maxima.

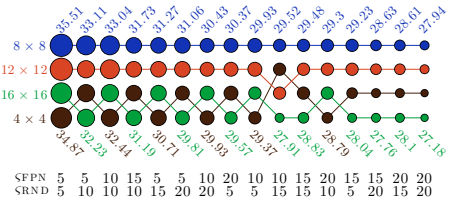


Fig. 9. Effect of different block sizes on the PSNR performance for the proposed method under different noise conditions. The area of the disks represents the average PSNR over different test videos for a specific noise level and block size, and each color represents a particular block size. The disks in each column are ordered in decreasing PSNR value from top to bottom; for each noise level we also report the best and worst PSNR value.

smallest of their respective PSDs Ψ_{RND} and Ψ_{FPN}; thus, the values of ς_{RND} and ς_{FPN} constitute only a rough quantitative indication of the actual strength of the two noise components, whose average standard-deviation can in fact be much larger than ς_{RND} and ς_{FPN}.

4) Parameter Settings: We set the maximum temporal extent of the spatiotemporal volumes to $H = t_{h^+} - t_{h^-} = 9$ with the reference block located in the middle, the size of the the 2-D blocks to $N \times N = 8 \times 8$, and the threshold factor to $\lambda_{3D} = 2.7$. As transform $\mathcal{T}_{3D}$ we utilize a separable 3-D DCT of size $N \times N \times H$.

The factor λ_{3D} is crucial: a too small or too large value may cause undersmoothing or oversmoothing of the data. In Fig. 8 we show the average PSNR obtained by the different methods for the denoising of the considered test videos and noise levels as λ_{3D} varies. We exclude *Miss America* from such average-value analyses because most of the sequence consists of a large smooth stationary background and thus its PSNR remains high even when a large λ_{3D} causes oversmoothing. The chosen $\lambda_{3D} = 2.7$ approximately yields the PSNR peak for both RF3D and E-RF3D; conversely, for WR, CR, and WRWF the best λ_{3D} needs to be larger (4.5, 6.15, and 4.65, respectively) to compensate the deficiencies of their assumed observation models. Note that $\lambda_{3D} = 2.7$ is equal to that used in [12] and is also not far from the universal threshold $\sqrt{2\log(NNH)}$ [22].

Both BM4D [15] and V-BM3D [13] modulate their filtering strength with the standard deviation σ_{AWGN} of the i.i.d. Gaussian noise assumed to corrupt the data; however, because of the mismatch between the AWGN model and the actual observations (1), there is no ideal value of σ_{AWGN}. We aim to compare the proposed algorithm against the best possible BM4D and V-BM3D results; thus, we use "oracle" σ^*_{AWGN} values that maximize the output PSNR individually in each experiment. Details are given in the Appendix.

The block size 8×8 is widely used in many image-processing applications because it enables the use of fast transform implementation (e.g., DCT or FFT) and also allows for a good data sparsification (e.g., BM3D denoising [12] or JPEG/MPEG compression). The denoising performances of the proposed E-RF3D using different block sizes are evaluated in Fig. 9; the performance is measured as average PSNR over the test sequences considered in our experiments, again minus Miss America, using $\lambda_{3D} = 2.7$. The area of each disk is proportional to the average PSNR (bigger disks indicate higher PSNR), and each color represents a particular block size. The disks in each column are ordered in descending PSNR value, and as one can clearly see, the best performance is always attained by 8×8 blocks (blue disks) with PSNR improvements ranging between 0.5dB and 1.5dB with respect to the worst case. Note that also V-BM3D as well as all others considered algorithms employ 8×8 blocks as basic data structures, whereas BM4D uses cubes of size $4 \times 4 \times 4$.

Our single-threaded MATLAB implementation[1] of the proposed algorithm used for the reported experiments processes a CIF-resolution sequence (i.e. 352×288) at approximately 1 frame per second on an Intel® i7-2640M CPU at 2.80-GHz.

B. Synthetic Data

The synthetic noisy sequences are generated according to the observation model (1) with the PSDs defined in (2) and (3) and shown in Fig. 2; ς_{RND} and ς_{FPN} are both simulated to remain constant in time. In order to present the best possible performances, every compared method use the optimized value of λ_{3D} discussed in Section VI-A4.

1) Joint Random and Fixed-Pattern Noise Removal: The PSNR denoising results under static and drifting FP are reported in Table I and Table II, respectively. Table II only includes E-RF3D because the other methods only exploit the PSD of the FPN, and not the actual realization FP, and thus are unaffected by the drift. In fact, the PSNR of such methods under static or drifting FP only differ by ± 0.1dB. Observe that a drift in the FP complicates the estimation (18), and thus the results of E-RF3D reported in Table II are not always as good as those obtained in case of static FP.

Referring to the PSNR results in Table I, RF3D and E-RF3D consistently outperform the results obtained under the less

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

TABLE I
PSNR (DB) DENOISING PERFORMANCE OF V-BM3D [13], BM4D [15], AND THE PROPOSED RF3D AND E-RF3D APPLIED TO DATA CORRUPTED BY SYNTHETIC NOISE AS IN (1) HAVING DIFFERENT COMBINATIONS OF ς_{FPN} AND ς_{RND}. THE SAME DATA IS ALSO FILTERED ASSUMING WHITE RANDOM NOISE (WR), COLORED RANDOM NOISE (CR), OR WHITE RANDOM AND WHITE FIXED-PATTERN NOISE (WRWF). THE FP IS STATIC IN TIME.

	Video	Foreman				Coastguard				Miss America				Flower Garden			
	Resolution	352×288				176×144				360×288				352×240			
	Frames	300				300				150				150			
ς_{FPN}	**Filter**	ς_{RND}															
		5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
5	V-BM3D	33.89	33.20	32.11	30.88	32.11	31.47	30.59	29.58	37.22	36.99	36.58	35.75	32.25	30.09	28.25	26.73
	BM4D	33.18	32.72	31.84	30.83	32.27	31.66	30.77	29.86	35.64	36.11	36.10	35.45	31.37	29.18	27.29	25.75
	WR	34.41	33.26	31.94	30.80	32.27	31.26	30.18	29.16	35.85	37.30	37.04	36.20	27.02	25.83	24.65	23.58
	CR	34.42	32.73	31.22	30.03	32.03	30.90	29.77	28.75	37.91	37.65	36.92	36.08	26.55	25.27	24.09	23.04
	WRWF	35.32	33.71	32.32	31.15	33.45	31.96	30.79	29.76	37.68	37.32	36.98	36.10	31.36	29.19	27.35	25.87
	RF3D	36.14	34.52	33.16	32.00	34.02	32.75	31.65	30.68	38.14	37.39	37.10	36.48	32.23	30.04	28.26	26.87
	E-RF3D	**38.52**	**35.44**	**33.53**	**32.15**	**35.74**	**33.83**	**32.22**	**31.03**	**38.80**	**38.17**	**37.38**	**36.54**	**33.12**	**30.42**	**28.44**	**26.92**
10	V-BM3D	29.87	29.77	29.67	29.50	28.35	28.27	28.14	27.96	34.83	34.75	34.62	34.46	28.04	27.41	26.61	25.73
	BM4D	29.12	29.12	29.11	29.02	27.70	27.68	27.67	27.57	34.36	34.31	34.12	33.80	26.52	25.99	25.29	24.49
	WR	29.40	29.84	30.03	29.89	28.09	28.25	28.21	27.99	29.03	30.55	32.25	33.97	25.46	24.78	23.95	23.12
	CR	30.72	30.58	30.00	29.30	28.77	28.68	28.32	27.83	32.01	33.92	34.81	34.90	25.20	24.39	23.47	22.61
	WRWF	31.92	31.33	30.68	30.09	29.95	29.26	28.68	28.20	34.82	34.55	34.35	34.30	27.94	27.02	25.96	24.95
	RF3D	33.01	32.34	31.55	30.81	30.55	30.04	29.47	28.97	36.30	35.79	35.28	34.76	28.64	27.80	26.82	25.89
	E-RF3D	**37.10**	**34.78**	**33.12**	**31.82**	**33.01**	**32.17**	**31.22**	**30.34**	**36.74**	**36.45**	**35.87**	**35.61**	**30.76**	**29.27**	**27.81**	**26.43**
15	V-BM3D	27.83	27.81	27.77	27.72	26.23	26.20	26.16	26.10	32.94	32.91	32.84	32.75	25.01	24.79	24.51	24.13
	BM4D	26.54	26.55	26.59	26.67	25.18	25.19	25.21	25.26	32.52	32.47	32.37	32.20	23.19	23.05	22.81	22.51
	WR	25.62	26.18	26.79	27.31	24.62	25.02	25.43	25.73	25.07	25.97	27.18	28.51	23.75	23.41	22.92	22.38
	CR	27.46	27.96	28.15	28.02	25.80	26.20	26.41	26.40	27.77	29.39	31.08	32.15	23.75	23.28	22.66	22.00
	WRWF	29.68	29.34	28.98	28.64	27.77	27.36	26.97	26.64	32.44	32.28	32.16	32.06	25.40	24.95	24.36	23.73
	RF3D	31.06	30.64	30.13	29.59	28.58	28.27	27.89	27.51	34.32	34.02	33.71	33.35	26.05	25.69	25.20	24.65
	E-RF3D	**35.24**	**33.93**	**32.56**	**31.42**	**30.84**	**30.70**	**30.19**	**29.44**	**34.84**	**34.61**	**34.24**	**33.91**	**29.07**	**27.99**	**26.99**	**25.99**
20	V-BM3D	26.50	26.50	26.48	26.46	24.83	24.83	24.80	24.78	31.46	31.45	31.40	31.33	22.72	22.63	22.50	22.32
	BM4D	26.24	26.23	26.18	26.12	23.34	23.35	23.40	23.42	30.93	30.91	30.84	30.73	20.76	20.72	20.64	20.53
	WR	22.73	23.16	23.81	24.47	21.96	22.31	22.79	23.27	22.37	22.92	23.75	24.75	22.11	21.95	21.71	21.39
	CR	24.81	25.40	26.02	26.37	23.34	23.82	24.36	24.74	24.91	26.00	27.49	28.91	22.42	22.14	21.74	21.27
	WRWF	28.15	27.87	27.59	27.35	26.19	25.93	25.65	25.41	30.55	30.43	30.35	30.29	23.53	23.26	22.90	22.49
	RF3D	29.78	29.45	29.02	28.60	27.23	27.02	26.73	26.44	**32.74**	**32.52**	**32.31**	**32.07**	24.20	24.01	23.74	23.42
	E-RF3D	**33.77**	**32.98**	**31.93**	**30.93**	**29.98**	**30.03**	**29.40**	**28.86**	**32.74**	**32.52**	**32.31**	**32.07**	**27.60**	**26.94**	**26.17**	**25.43**

TABLE II
PSNR (DB) DENOISING PERFORMANCE OF E-RF3D APPLIED TO DATA CORRUPTED BY SYNTHETIC NOISE AS IN (1) HAVING DIFFERENT COMBINATIONS OF ς_{FPN} AND ς_{RND}. THE FP PRESENTS A DRIFT IN TIME. IN THIS CONDITION V-BM3D, BM4D WR, CR, WRWF, AND RF3D OBTAIN RESULTS COMPARABLE (± 0.1DB) TO THE ONES REPORTED IN TABLE I, AND THUS ARE NOT SHOWN.

	Video	Foreman				Coastguard				Miss America				Flower Garden			
	Resolution	352×288				176×144				360×288				352×240			
	Frames	300				300				150				150			
ς_{FPN}	**Filter**	ς_{RND}															
		5	10	15	20	5	10	15	20	5	10	15	20	5	10	15	20
5	E-RF3D	37.87	35.10	33.32	32.00	35.30	33.43	31.95	30.80	38.02	37.89	37.28	36.52	32.88	30.30	28.35	26.90
10	E-RF3D	35.61	34.07	32.61	31.43	31.97	31.40	30.53	29.69	36.37	36.11	35.77	35.34	30.46	28.95	27.58	26.32
15	E-RF3D	33.28	32.76	31.75	30.76	29.70	29.67	29.18	28.59	34.36	34.29	34.20	33.79	28.57	27.58	26.64	25.71
20	E-RF3D	31.31	31.26	30.74	30.04	28.29	28.33	28.10	27.65	32.73	32.55	32.33	32.09	26.87	26.47	25.75	24.97

accurate WR, CR, and WRWF assumptions with a substantial PSNR improvement in almost every experiment. Similarly, the state-of-the-art V-BM3D and BM4D filters (which we remark are designed for AWGN) are outperformed by the RF3D and E-RF3D methods. This demonstrates the importance of correctly modeling and appropriately filtering the two different components of the noise. It is interesting to notice that whenever ς_{FPN} is large enough (≥ 10), the PSNR of WR and CR increase as ς_{RND} increases. This apparent counterintuitive behavior is explained by the fact that neither WR nor CR model the FPN component, which may accumulate in the temporal DC plane of the 3-D volume spectrum. Such accumulation is particularly significant when motion is absent, as shown by (13), and corresponds to DC-plane coefficients having much larger noise variance than the rest of the spectrum. WR and CR make no distinction between DC-plane coefficients and AC coefficients, thus an increase of the RND noise component results in a higher filtering strength, which partly compensates

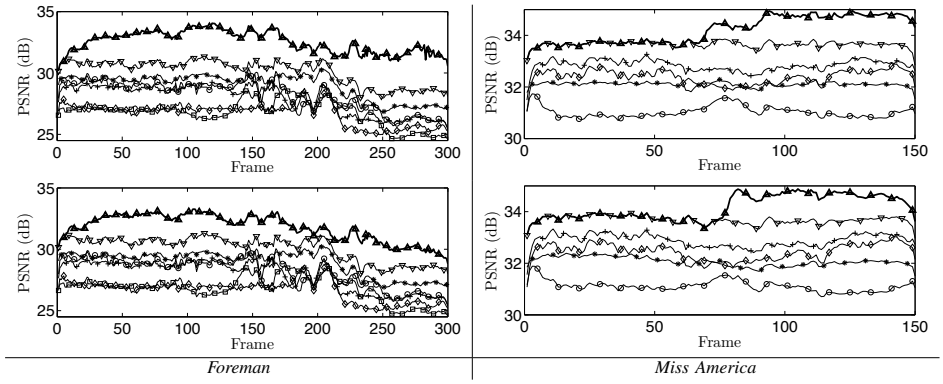


Fig. 10. Frame-by-Frame PSNR (dB) output of the videos *Foreman* and *Miss America* corrupted by synthetic noise having $\varsigma_{\text{RND}} = \varsigma_{\text{FPN}} = 15$ with either static FP (top row) or drifting FP (bottom row). We show the results of V-BM3D ($+$), BM4D ($\diamond$), WR ($\square$), CR ($\circ$), WRWF ($*$), RF3D ($\triangledown$), and E-RF3D ($\triangle$).

their model deficiency. The sequence *Flower Garden* is an exception: being a fast moving scene there is no accumulation of FPN and thus the PSNR naturally decreases with the increase of ς_{RND}. An additional remark about Table I regards the results of RF3D and E-RF3D for *Miss America* under high levels of ς_{FPN}: since the sequence presents little motion, E-RF3D is challenged to get a reliable estimate of the FP under strong FPN, and thus it is not able to provide the same performance gain as that of the other cases. As a matter of fact RF3D and E-RF3D provide the same PSNR results at $\varsigma_{\text{FPN}} = 20$.

Fig. 10 shows the frame-by-frame PSNR of *Foreman* and *Miss America* corrupted by random and fixed-pattern noise having $\varsigma_{\text{RND}} = \varsigma_{\text{FPN}} = 15$. *Miss America* and the first half of *Foreman* have low motion activity, whereas the second half of *Foreman* exhibits a high motion activity because of a fast transition in the scene. In good accord with the numerical results of Table I and Table II, E-RF3D ($\triangle$) always outperform the results obtained under WR ($\square$), CR ($\circ$), and WRWF ($*$) assumptions, as well as those of V-BM3D ($+$) and BM4D ($\diamond$). RF3D ($\triangledown$) is in few cases marginally inferior to V-BM3D ($+$). The advantage of the enhanced fixed-pattern suppression is clearly visible in all experiments, with the immediate and substantial PSNR improvement after the first estimate of the FP is subtracted (around the 10th frame in *Foreman* and between the 50th and the 75th frame in *Miss America*).

In Fig. 11, we show a denoised frame from *Foreman* and *Miss America* corrupted by synthetic noise having $\varsigma_{\text{RND}} = \varsigma_{\text{FPN}} = 15$, as well as the FP estimate obtained by E-RF3D. The noise-free data is shown in Fig. 7. Under the WR and CR assumptions the filter is unable to properly remove the FPN component, whose residual artifacts can be easily spotted within the denoised frames. In the WRWF results, we notice a good suppression of the random noise, but the structures of the FPN are still clearly visible. Conversely, RF3D and E-RF3D generate more visually pleasant images, as the artifacts of the FPN are dramatically reduced and many high-frequency features, such as the hair and facial features of *Foreman* or the wrinkles in the clothes of *Miss America*, are nicely preserved. The results obtained by the V-BM3D and BM4D algorithms are separately presented in Fig. 12: as one can clearly see, the visual quality is significantly inferior those of RF3D and E-RF3D because of the remaining artifacts due to the FPN and the excessive loss of details.

2) Separate Random and Fixed-Pattern Noise Removal: The proposed filter is designed to jointly remove the random and fixed-pattern noise components, but for this set of experiments we modify it such that the two noise components are suppressed one at a time in two cascading passes. In other words the modified filter is applied twice on the observed data, first suppressing the random noise and then the FPN, or viceversa. From Fig. 13 it can be seen that whenever the FPN is suppressed before the random noise, the visual quality of the denoised videos is comparable or even slightly better to that obtained by the joint denoising strategy (at the obvious expense of a doubled computational load). The improvement is due to the assumption of zero random noise made in the first pass: if $\varsigma_{\text{RND}} = 0$ the number M of frames required for the FP estimation is small and thus the FP estimate can be obtained faster. Conversely, the reversed schema, implemented by suppressing the FPN after the random noise, is not as effective. In fact, as can be seen from the cheek of *Foreman* in Fig. 13, the corresponding denoising results exhibit significant FP artifacts.

3) Additive White Gaussian Noise Removal: In the final set of experiments using synthetic noise, we evaluate the proposed method against sequences corrupted solely by i.i.d. additive (white) Gaussian random noise with standard deviation σ_{AWGN}, which is assumed to be known. The proposed RF3D operates according to the WR assumption with $\hat{\varsigma}_{\text{RND}} = \sigma_{\text{AWGN}}$. The

Fig. 11. From top to bottom: denoising results of WR, CR, WRWF, RF3D, E-RF3D, and the FP estimate obtained from E-RF3D for *Foreman* and *Miss America* corrupted by synthetic noise having $\varsigma_{\text{RND}} = \varsigma_{\text{FPN}} = 15$.

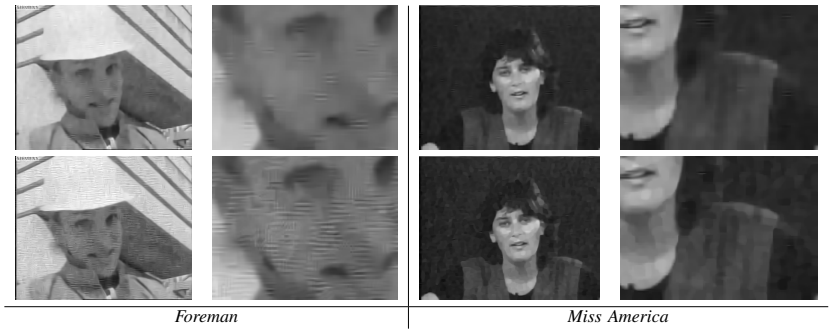


Fig. 12. From top to bottom: denoising results of V-BM3D (*Foreman* 27.77 dB, *Miss America* 32.84 dB) and BM4D (*Foreman* 26.59 dB, *Miss America* 32.37 dB). The synthetic correlated noise is characterized by $\varsigma_{\text{RND}} = \varsigma_{\text{FPN}} = 15$.

Fig. 13. Denoising results for *Foreman* and *Miss America* corrupted by synthetic correlated noise having $\varsigma_{\text{RND}} = \varsigma_{\text{FPN}} = 15$ using E-RF3D to separately remove the two noise components. Top: first suppression of random noise and then FPN (*Foreman* 31.23dB, *Miss America* 33.37dB); bottom first suppression of the FPN and then random noise (*Foreman* 32.02dB, *Miss America* 34.29dB). For comparison, as can be seen in Table I, E-RF3D with joint-noise suppression provides 32.56dB for *Foreman* and 34.24dB for *Miss America*.

TABLE III
PSNR (DB) DENOISING PERFORMANCE OF V-BM3D, BM4D, AND RF3D FOR DATA CORRUPTED BY I.I.D. GAUSSIAN NOISE WITH STANDARD DEVIATION σ_{AWGN}.

σ_{AWGN}	Video Res. Frames	Foreman 352×288 300	Coastg. 176×144 300	Miss Am. 360×288 150	Fl. Gard. 352×240 150
5	V-BM3D	39.84	38.33	41.50	36.53
	BM4D	39.77	38.87	**42.02**	36.09
	RF3D	**40.27**	**39.43**	41.98	**36.58**
10	V-BM3D	36.55	34.82	39.64	**32.15**
	BM4D	36.38	35.31	**40.28**	31.39
	RF3D	**36.88**	**35.77**	40.19	32.06
20	V-BM3D	33.40	31.76	37.95	**28.30**
	BM4D	33.27	32.13	38.33	27.27
	RF3D	**33.72**	**32.36**	**38.40**	28.00
40	V-BM3D	29.99	28.28	35.46	**24.34**
	BM4D	30.39	29.08	36.03	23.40
	RF3D	**30.61**	**29.09**	**36.23**	24.21

rationale of these experiments is to compare RF3D against V-BM3D and BM4D on data where the latter two methods operate in ideal conditions; the results for different values of σ_{AWGN} are reported in Table III. From the table we can notice that the best-performing method is not the same for all experiments: while RF3D yields the best results in most of the cases, it also sometimes falls behind. The gap between the highest and lower PSNR values is at most 1.1dB, and typically much smaller; overall, these three methods perform comparably. Thus, the significant advantage (often several dB) of RF3D and especially E-RF3D in the case of correlated and fixed-pattern noise reported in Table I is a result of a correct modeling of the observed data, and not of an intrinsically more powerful algorithm.

C. LWIR Thermography Data

In this section we demonstrate the appropriateness of the proposed method through the denoising of two real LWIR thermography sequences acquired using a FLIR Tau 320 camera: the first sequence, *Matteo*, is characterized by high

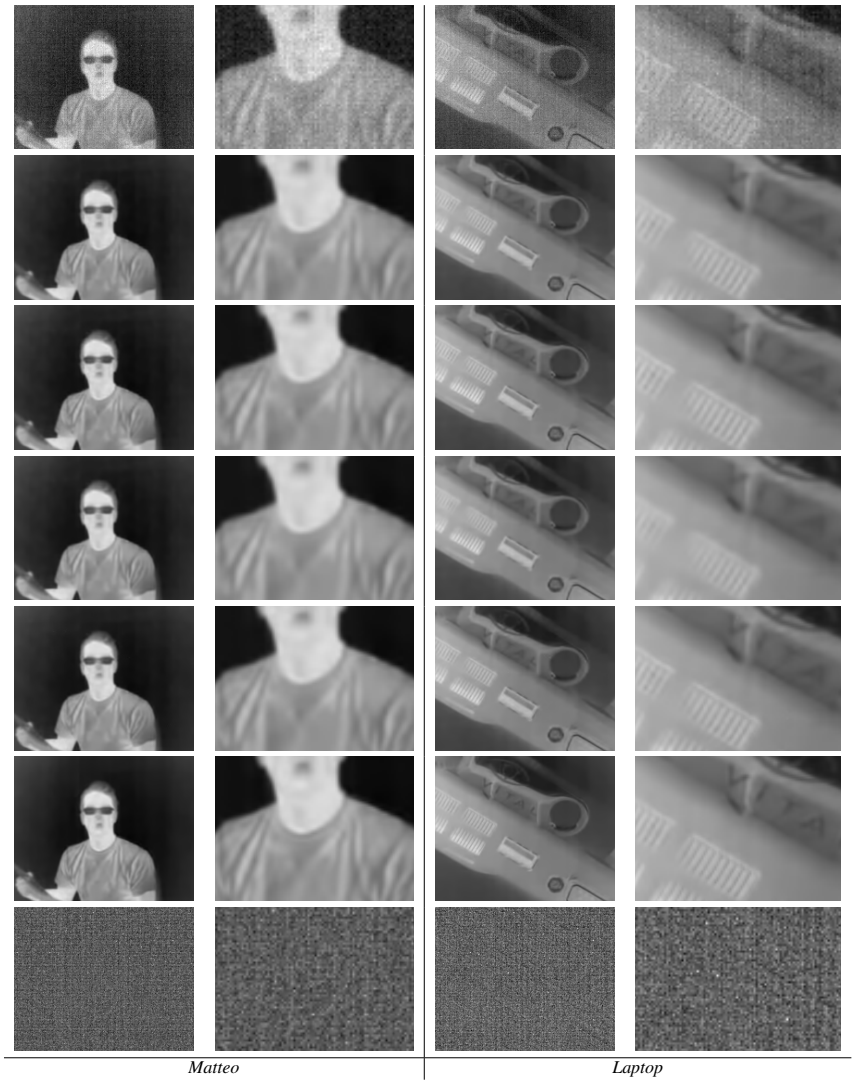


Fig. 14. From top to bottom: denoising results of WR, CR, WRWF, RF3D, E-RF3D, and the FP estimate obtained from E-RF3D for LWIR thermography sequences *Matteo* and *Laptop* acquired by a FLIR Tau 320 camera.

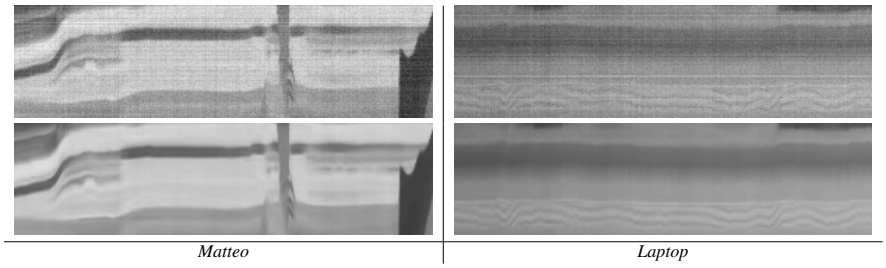


Fig. 15. Temporal cross-section of the noisy (top row) and E-RF3D denoised (bottom row) *Matteo* and *Laptop* sequences acquired by a FLIR Tau 320 camera. Both sequences consists of 300 frames. The artifacts of the FPN and the random noise are evident from the roughly constant streaks in time (horizontal direction) and space (vertical direction), respectively.

motion activity, whereas the second, *Laptop*, contains a more static scene[2]. The noise in the acquired data is characterized by $\varsigma_{\text{RND}} \approx 2.3$ and $\varsigma_{\text{FPN}} \approx 1.5$ over a $[6010, 6100]$ range, which corresponds to $\varsigma_{\text{RND}} \approx 6.5$ and $\varsigma_{\text{FPN}} \approx 4.3$ for a $[0, 255]$ range.

Objective assessments cannot be made because the ground-truth is not available, however, referring to Fig. 14, we can observe that under the WR, CR and WRWF assumptions the filter is not able to remove the noise, and that the best visual quality is obtained by the proposed RF3D and its enhancement E-RF3D. In particular, E-RF3D provides the best FPN suppression, which is evident from smooth areas such as the background of *Matteo*, and the best detail preservation, as can be seen from the folds in the tee-shirt of *Matteo* or the grid and letters in *Laptop*.

In the last row of Fig. 14 we show the FP estimate obtained from E-RF3D. As can be noticed, in the case of the static sequence *Laptop* part of the signal leaks into the residuals and is accumulated into the FP estimate. This is explained by the difficulty of unambiguously distinguishing the static information of the signal from the pattern of the FPN without the aid of motion (as described in Section V-A). In such cases the estimate of the FP (18) is likely to be less accurate, and thus isolating the noise component may be challenging. However, in spite of this mild leakage, the quality of the E-RF3D estimate is clearly superior to that of the compared methods (including RF3D), with better preservation of details and suppression of noise. In Fig. 15, we illustrate the effects of the random and fixed-pattern noise from the temporal cross-section of *Matteo* and *Laptop* (i.e. the horizontal dimension represents time, and the vertical dimension represents a particular cross-section of each frame). The effects of the noise structure of the FPN and RND can be respectively noticed from the horizontal and vertical streaks in the noisy data, whereas in the denoised counterparts these artifacts are effectively removed while preserving the fine (temporal) details, such as the three "claws" in the second half of *Matteo* and the "waves" in *Laptop*.

[2]This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. This includes the raw and filtered LWIR sequences of *Matteo* and *Laptop* as uncompressed AVI format movie clips. The material as GZIP Tar Archive file is 143 MB in size.

TABLE IV
MINIMUM (LEFT VALUE IN EACH CELL) AND MAXIMUM (RIGHT VALUE IN EACH CELL) VALUES OF THE ORACLE σ^*_{AWGN} PARAMETERS OF BM4D AND V-BM3D FOR EACH COMBINATION OF NOISE SCALING FACTORS ς_{FPN} AND ς_{RND}.

ς_{FPN}	Filter	ς_{RND} 5		10		15		20	
		min	max	min	max	min	max	min	max
5	V-BM3D	10	26	14	27	19	30	25	38
	BM4D	10	28	14	29	19	33	25	41
10	V-BM3D	16	57	19	60	23	60	27	59
	BM4D	17	189	19	160	23	152	27	150
15	V-BM3D	24	93	26	92	30	91	32	90
	BM4D	25	274	27	272	30	268	32	264
20	V-BM3D	34	124	35	123	39	120	40	120
	BM4D	35	385	37	380	39	371	40	352

VII. CONCLUSION

The contribution of this work is twofold. First, we developed an observation model for data corrupted by a combination of two spatially correlated components, i.e. random and fixed-pattern noise, each having its own non-flat PSD. This observation model can characterize several imaging sensors, and is particularly successful in describing the output of LWIR imagers. Second, we embed such observation model within a filtering framework based on 3-D spatiotemporal volumes built by stacking a sequence of blocks along the motion trajectories of the video. The volumes are then sparsified by a decorrelating 3-D transform, and then filtered in 3-D transform domain through a shrinkage operator based on both the PSDs of the noise components and on the relative spatial position of the blocks in the volume. Extensive experimental analysis demonstrates the subjective and objective (PSNR) effectiveness of the proposed framework for the denoising of synthetically corrupted videos, as well as the high visual quality achieved by the filtering of real LWIR thermography sequences. We further showed the capabilities of online FP estimation and subtraction to improve the denoising results.

APPENDIX

The denoising results of V-BM3D and BM4D in Table I are obtained with a default implementation of those algorithms

[13], [15] and an "oracle" value σ^*_{AWGN} of the assumed noise standard deviation. In particular, for each video and for each separate combination of ς_{RND} and ς_{FPN} under either static or drifting FPN, we have optimized σ^*_{AWGN} such that it yields the maximum PSNR value in each individual experiment. Due to length limitation and for the sake of illustration simplicity, in Table IV we report only the minimum and maximum of such optimum σ^*_{AWGN} values for all combination of noise scaling factors. As can be clearly seen, the difference between the maximum and minimum values notably increases with ς_{FPN}, thus indicating the impossibility of compensating the mismatch in the observation model by a simple tuning of the filter's parameters. Also, note how the maximum values tend to be very large in order to compensate the accumulated FPN in the volume spectra as quantified in (13).

REFERENCES

[1] A. F. Milton, F. R. Barone, and M. R. Kruer, "Influence of nonuniformity on infrared focal plane array performance," *Optical Engineering*, vol. 24, no. 5, pp. 245 855–245 855, Aug. 1985.

[2] M. T. Eismann and C. Schwartz, "Focal plane array nonlinearity and nonuniformity impacts to target detection with thermal infrared imaging spectrometers," in *Proceedings of the SPIE Infrared Imaging Systems: Design, Analysis, Modeling, and Testing*, vol. 3063, Jun. 1997, pp. 164–173.

[3] A. El Gamal, B. A. Fowler, H. Min, and X. Liu, "Modeling and estimation of FPN components in CMOS image sensors," in *Proceedings of the SPIE Solid State Sensor Arrays: Development and Applications*, vol. 3301, 1998, pp. 168–177.

[4] M. J. Schulz and L. V. Caldwell, "Nonuniformity correction and correctability of infrared focal plane arrays," in *Proceedings of the SPIE Infrared Imaging Systems: Design, Analysis, Modeling, and Testing*, vol. 2470, May 1995, pp. 200–211.

[5] A. Kumar, S. Sarkar, and R. P. Agarwal, "A novel algorithm and hardware implementation for correcting sensor non-uniformities in infrared focal plane array based staring system," *Infrared Physics & Technology*, vol. 50, no. 1, pp. 9–13, Mar. 2007.

[6] P. M. Narendra, "Reference-free nonuniformity compensation for IR imaging arrays," in *Proceedings of the SPIE Smart Sensors*, vol. 252, Jan. 1980, pp. 10–17.

[7] D. A. Scribner, K. A. Sarkady, J. T. Caulfield, M. R. Kruer, G. Katz, C. J. Gridley, and C. Herman, "Nonuniformity correction for staring IR focal plane arrays using scene-based techniques," in *Proceedings of the SPIE Infrared Detectors and Focal Plane Arrays*, vol. 1308, Apr. 1990, pp. 224–233.

[8] J. Harris and C. Yu-Ming, "Nonuniformity correction of infrared image sequences using the constant-statistics constraint," *IEEE Transactions on Image Processing*, vol. 8, no. 8, pp. 1148–1151, Aug 1999.

[9] S. N. Torres, J. E. Pezoa, and M. M. Hayat, "Scene-based nonuniformity correction for focal plane arrays by the method of the inverse covariance form," *Applied Optics*, vol. 42, no. 29, pp. 5872–5881, Oct. 2003.

[10] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral-spatial adaptive total variation model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 3660–3677, Oct. 2012.

[11] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4729–4743, Aug. 2014.

[12] 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, Aug. 2007.

[13] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *Proceedings of the European Signal Processing Conference*, Sep. 2007. [Online]. Matlab code available: http://www.cs.tut.fi/~foi/GCF-BM3D/

[14] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, Sep. 2012.

[15] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 119–133, Jan. 2013. [Online]. Matlab code available: http://www.cs.tut.fi/~foi/GCF-BM3D/

[16] E. P. Simoncelli and B. Olshausen, "Natural image statistics and neural representation," *Annual Review of Neuroscience*, vol. 24, pp. 1193–1216, May 2001.

[17] J. S. De Bonet, "Noise reduction through detection of signal redundancy," Rethinking Artificial Intelligence, MIT AI Lab, Tech. Rep., 1997.

[18] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[19] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision*, vol. 86, pp. 1–32, Jan. 2010.

[20] H. Ji, S. Huang, Z. Shen, and Y. Xu, "Robust video restoration by joint sparse and low rank matrix approximation," *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1122–1142, Nov. 2011.

[21] P. Milanfar, "A tour of modern image filtering: New insights and methods, both practical and theoretical," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 106–128, Jan. 2013.

[22] D. Donoho, I. Johnstone, and I. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1993.

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

[24] 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.

[25] L. Jovanov, A. Pizurica, S. Schulte, P. Schelkens, A. Munteanu, E. Kerre, and W. Philips, "Combined wavelet-domain and motion-compensated video denoising based on video codec motion estimation methods," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 3, pp. 417–421, Mar. 2009.

[26] Z. Wang and Q. Li, "Statistics of natural image sequences: temporal motion smoothness by local phase correlations," in *Proceedings of the SPIE Human Vision and Electronic Imaging*, vol. 7240, Jan. 2009, pp. 1–12.

[27] R. Kleihorst, R. Lagendijk, and J. Biemond, "Noise reduction of image sequences using motion compensation and signal decomposition," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 274–284, 1995.

[28] A. Buades, B. Coll, and J. M. Morel, "Denoising image sequences does not require motion estimation," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance*, Sep. 2005, pp. 70–74.

[29] J. Boulanger, C. Kervrann, and P. Bouthemy, "Space-time adaptation for patch-based image sequence restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1096–1102, Jun. 2007.

[30] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 27–35, Jan. 2009.

[31] E. Dubois and S. Sabri, "Noise reduction in image sequences using motion-compensated temporal filtering," *IEEE Transactions on Communications*, vol. 32, no. 7, pp. 826–831, Jul. 1984.

[32] C. Liu and W. T. Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in *Proceedings of the European conference on Computer vision*, 2010, pp. 706–719.

[33] M. Bertero, T. Poggio, and V. Torre, "Ill-posed problems in early vision," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 869–889, Aug. 1988.

[34] J. Brailean, R. Kleihorst, S. Efstratiadis, A. Katsaggelos, and R. Lagendijk, "Noise reduction filters for dynamic image sequences: A review," *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1272–1292, Sep. 1995.

[35] A. Foi and M. Maggioni, "Methods and systems for suppressing noise in images," Patent Application US 13/943,035, Filed Jul. 16, 2013.

[36] F. R. Hampel, "The influence curve and its role in robust estimation," *Journal of the American Statistical Association*, vol. 69, no. 346, pp. 383–393, Jun. 1974.

[37] D. Donoho and I. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, Dec. 1995.

[38] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Transactions on Image Processing*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3440-9
ISSN 1459-2045