



TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Aram Danielyan

**Block-based Collaborative 3-D Transform Domain
Modeling in Inverse Imaging**



Julkaisu 1145 • Publication 1145

Tampereen teknillinen yliopisto. Julkaisu 1145
Tampere University of Technology. Publication 1145

Aram Danielyan

Block-based Collaborative 3-D Transform Domain Modeling in Inverse Imaging

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 TB104, at Tampere University of Technology, on the 5th of August 2013, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2013

ISBN 978-952-15-3101-9 (printed)
ISBN 978-952-15-3214-6 (PDF)
ISSN 1459-2045

Abstract

The recent developments in image and video denoising have brought a new generation of filtering algorithms achieving unprecedented restoration quality. This quality mainly follows from exploiting various features of natural images. The non-local self-similarity and sparsity of representations are key elements of the novel filtering algorithms, with the best performance achieved by adaptively aggregating multiple redundant and sparse estimates. In a very broad sense, the filters are now able, given a perturbed image, to identify its plausible representative in the space or manifold of possible solutions. Thus, they are powerful tools not only for noise removal, but also for providing accurate adaptive regularization to many ill-conditioned inverse imaging problems.

In this thesis we show how the image modeling of the well known Block-matching 3-D transform domain (BM3D) filter can be exploited for designing efficient algorithms for image reconstruction.

First, we formalize the BM3D-modeling in terms of the overcomplete sparse frame representation. We construct analysis and synthesis BM3D-frames and study their properties, making BM3D-modeling available for use in variational formulations of image reconstruction problems.

Second, we demonstrate that standard variational formulations based on single objective optimization, such as Basis Pursuit Denoising and its various extensions, cannot be used with the imaging models generating non-tight frames, such as BM3D. We propose an alternative sparsity promoting problem formulation based on the generalized Nash equilibrium (GNE).

Finally, using BM3D-frames we develop practical algorithms for image deblurring and super-resolution problems. To the best of our knowledge, these algorithms provide results which are the state of the art in the field.

Acknowledgements

This thesis is based on part of the research work which has been carried out by the author between years 2008 and 2012, at the Department of Signal Processing of Tampere University of Technology (TUT), Finland.

I would like to express my sincere gratitude to my supervisor, professor Karen Egiazarian, for his inspiring ideas, advices and constant support. I am also indebted to my advisors, professors Vladimir Katkovnik and Alessandro Foi, for sharing their profound knowledge, giving example how to be constructively critical and how to achieve high-quality research.

My special thanks go to professor Jaakko Astola, director of the Signal Processing Algorithm Group (SPAG) and of Tampere International Center for Signal Processing (TICSP), who offered me the opportunity to start working at TUT.

The work presented in this thesis has been done within research projects of the Academy of Finland's Finnish Centre of Excellence program and Tampere Doctoral Programme in Information Science and Engineering (TISE). Without their financial support, gratefully acknowledged, this research would have not been possible.

Finally, I would like to thank all my friends from TUT, for the enjoyable atmosphere they create; my father, for encouraging me to study signal processing; and my wife Elizaveta, for her love, understanding and support.

Tampere, May 2013

Aram Danielyan

Contents

Abstract	iii
Acknowledgements	v
Contents	vii
Introduction to the thesis	ix
Outline of the thesis	ix
Link to publications	x
List of publications and author's contribution	x
Notation and conventions	xii
1 Sparsity-based image reconstruction	1
1.1 Sparsity-based models in inverse problems	1
1.1.1 Observation model	2
1.1.2 Analysis-based formulation	2
1.1.3 Synthesis-based formulation	3
1.1.4 Relaxed analysis formulation	4
1.2 Methods for solving sparse inverse problems	5
1.3 Dictionaries for sparse image approximations	6
1.3.1 Non-adaptive dictionaries	7
1.3.2 Learned dictionaries	7
1.3.3 Adaptive dictionaries based on fixed transforms	8
1.4 Block-matching based 3-D transform domain image modeling	9
1.4.1 BM3D-frames	10
2 Denoising with non-tight frames: from Basis Pursuit Denoising to generalized Nash equilibrium	13
2.1 Basis Pursuit denoising with non-tight frames	13
2.1.1 Problem formulation	13
2.1.2 ADMM algorithm for BP denoising problem	14
2.1.3 Convergence	15
2.1.4 Discussion	16
2.1.5 Criticism	18
2.2 Denoising as a Generalized Nash equilibrium problem	22
2.2.1 Problem formulation	22

2.2.2	Algorithm for solving GNE problem	23
2.2.3	Convergence	23
2.2.4	Discussion	24
2.3	Experiments	24
2.4	Conclusions	26
3	Deblurring	33
3.1	Proposed approach	35
3.1.1	Gaussian data: IDD-BM3D algorithm	35
3.1.2	Poissonian data: PIDD-BM3D algorithm	36
3.2	Implementation of IDD-BM3D and PIDD-BM3D algorithms	39
3.3	Selection of regularization parameters	40
3.4	Experiments	41
3.4.1	Gaussian data	41
3.4.2	Poissonian data	42
4	Super-resolution	55
4.1	Fusion via collaborative filtering	57
4.2	SR algorithm with BM3D-modeling	58
4.3	Implementation	60
4.3.1	Initialization	60
4.3.2	Computing the inverse	61
4.4	Discussion	62
4.4.1	Reconstruction from noise-free data	62
4.4.2	Reconstruction from noisy data	65
4.4.3	Relation to the IFSR algorithm	65
	Conclusions to the thesis	69
C.1	Overview	69
C.2	Future research	69
	Bibliography	71
	Publications	79

Introduction to the thesis

Outline of the thesis

Chapter 1 contains an overview of the sparsity based models and methods used in image reconstruction problems. We present the ideas behind the sparsity based models, review the methods for solving inverse problems under sparsity constraints and discuss construction of effective dictionaries rendering sparse image models. We shortly present the concepts of the Block-matching 3-D transform domain (BM3D) image modeling and its interpretation as a non-tight frame [DKE12].

The goal of Chapter 2 is to demonstrate the main ideas laying behind the image reconstruction algorithms we propose in the next chapters. First, on the example of Basis Pursuit denoising problem, we show that for non-tight frames sparsity promoting models based on a single objective minimization lead to the reconstructions where the strength of the regularization varies for different spatial positions. Then, we propose an alternative sparsity promoting model based on generalized Nash equilibrium (GNE), develop the algorithm for solving it, and show its effectiveness in the simulated experiments.

In Chapter 3 we review the image deblurring algorithms proposed in our papers [DKE12] and [DFKE11], where the problem is formulated as a search for the generalized Nash equilibrium, which would balance the fit to the observation model and complexity of the estimate in the BM3D-frame domain. This approach has been found extremely successful, leading, to the best of our knowledge, to the state-of-the-art results in the field.

In Chapter 4 we discuss the role of BM3D-modeling in super-resolution. We show that 3-D transform domain modeling can be treated not only as a regularization tool, but also as a robust data fusion approach. This important observation will help us to explain the efficacy of the SR algorithm based on iterative BM3D filtering [DFKE08c] and also to suggest more efficient ways of implementing it. Finally, we develop a new SR algorithm based on the BM3D-modeling, deriving it as a solution of a GNE problem. Its key difference from [DFKE08c] is the way how the different 3-D spectral components are treated. Further, in contrast to [DFKE08c], the new algorithm is directly designed to handle noisy data.

Conclusive remarks are given at the end of the thesis.

Link to publications

The five publications included in this thesis represent two series of works, devoted to the development of reconstruction algorithms for different inverse imaging problems. These two series were originally developed independently, but as we show in Chapter 4, there are deep connections between them.

First series starts with publication [DFKE08d], which considers *image upsampling* problem (Chapter 4, p. 55) and proposes reconstruction algorithm based on iterative BM3D filtering. This algorithm has been extended in [DFKE08c] to deal with more general *video super-resolution* problem (Chapter 4, p. 55). The book chapter [DFKE10] summarizes the results of the previous two publications and the results of [EFK07] on compressive sensing reconstruction, presenting a unified view on the methods for solving these problems (author of the current thesis is not among the authors of [EFK07]).

Second series is devoted to the image *deblurring* problem. In [DKE12] and [DFKE11] we develop methods for deblurring images corrupted respectively with Gaussian (Chapter 3, p. 35) and Poisson noise (Chapter 3, p. 36). The reconstruction technique developed in [DKE12] is general and applicable to a wide class of linear inverse imaging problems. Particularly, we use it to develop a new super-resolution algorithm (Chapter 4, p. 58).

List of publications and author's contribution

The main contribution of this compound thesis is contained in the publications listed below. However material in Chapters 2 and 4 is novel and has not been published before.

- [DKE12]: A. Danielyan, V. Katkovnik, and K. Egiazarian, “BM3D frames and variational image deblurring,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, April 2012.
- [DFKE11]: A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, “Deblurring of Poissonian images using BM3D frames,” *Proc. Wavelets and Sparsity XIV*, SPIE, vol. 8138, San Diego, September 2011.
- [DFKE10]: A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, “Spatially adaptive filtering as regularization in inverse imaging: compressive sensing, super-resolution, and upsampling,” *Super-Resolution Imaging*, P. Milanfar, Ed., pp. 123–153. CRC Press, 2010.
- [DFKE08c]: A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, “Image and video super-resolution via spatially adaptive block-matching filtering,” *Proc. International Workshop on Local and Non-Local Approximations in Image Processing, LNLA 2009*, Lausanne, Switzerland, August 2008, pp. 53–60.
- [DFKE08d]: A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, “Image upsampling via spatially adaptive block-matching filtering,” *Proc. European*

Signal Processing Conference, EUSIPCO2 008, Lausanne, Switzerland, August 2008.

The author of the thesis is the main contributor of the work presented in [DKE12]-[DFKE08d]. The contribution of all coauthors is nevertheless essential and without it, none of these publications would have been possible.

All co-authors have confirmed their agreement on the above statement.

Notation and conventions

The symbols \mathbb{R} , \mathbb{Z} , and \mathbb{N} indicate, respectively, the real numbers, the integer numbers, and the natural numbers.

We use lowercase boldface letters for denoting vectors, e.g. $\mathbf{x} \in \mathbb{R}^N$, and uppercase bold letters for matrices corresponding to linear operators, e.g., $\mathbf{A} : \mathbb{R}^N \rightarrow \mathbb{R}^M$. Components of the vector are referred using regular letters and subscripts, e.g., v_i indicates the i -th component of \mathbf{v} . If notation already has a subscript, e.g., \mathbf{v}_t^k we put the component index outside the brackets $(\mathbf{v}_t^k)_i$.

Matrices representing images are denoted by normal lowercase letters $y : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$. *Vectorized matrix* is a column vector constructed from a matrix by stacking its columns. The conjugate transpose of a matrix or vector is denoted by the superscript T .

The central dot \cdot stands for a “mute variable” or “mute argument”.

We use notation $\|\cdot\|_p$, $p > 0$, for functionals $\|\mathbf{x}\|_p = \left(\sum_i^N |x_i^p|\right)^{1/p}$, $\mathbf{x} \in \mathbb{R}^N$. Additionally, we use $\|\cdot\|_0$ to denote the functional which gives the number of non-zero elements of its argument. For $p \geq 1$, $\|\cdot\|_p$ are the well known l_p -norms of \mathbb{R}^N . For $p < 1$, functionals $\|\cdot\|_p$ do not define norms, since they do not fulfill the triangle inequality condition. However, in the text we sometimes abuse the terminology and refer to $\|\cdot\|_p$ as l_p -norms also when $p < 1$.

The inner-product of a vector space is denoted by $\langle \cdot, \cdot \rangle$. The Fourier transform is denoted by \mathcal{F} .

The bar decoration $\bar{\cdot}$ is used to denote true values, while the hat $\hat{\cdot}$ denotes estimated values (e.g., “ \hat{y} is the estimate of \bar{y} ”).

$\mathcal{N}(\mu, \sigma^2)$ and $\mathcal{P}(\lambda)$ respectively denote the normal (i.e., Gaussian) distribution with mean μ and variance σ^2 and the Poisson distribution with mean (and variance) λ . The notation $z \sim \mathcal{P}(\lambda)$ means that the random variable z is distributed according to a $\mathcal{P}(\lambda)$ distribution, which implies that the probability of z being equal to k is $P(z = k) = e^{-\lambda} \frac{\lambda^k}{k!}$, $k \in \mathbb{N}$. Similarly, if $z \sim \mathcal{N}(\mu, \sigma^2)$, we have that the probability density of z is $p(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$, $z \in \mathbb{R}$.

For the images, unless differently noted, we assume that the data-range is $[0, 255]$, where 0 and 255 correspond, respectively, to black and white.

We use the following standard criteria functions to assess the objective quality of an estimate \hat{y} of $\bar{y} : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$:

$$\begin{aligned}
 (\text{signal-to-noise ratio}) \text{ SNR} &= 20 \log_{10} \left(\frac{\|\bar{y}\|_2}{\|\bar{y} - \hat{y}\|_2} \right), \\
 (\text{improvement in SNR}) \text{ ISNR} &= 20 \log_{10} \left(\frac{\|\bar{y} - z\|_2}{\|\bar{y} - \hat{y}\|_2} \right), \\
 (\text{peak SNR}) \text{ PSNR} &= 20 \log_{10} \left(\frac{255 \cdot N^2}{\|\bar{y} - \hat{y}\|_2} \right), \\
 (\text{mean squared error}) \text{ MSE} &= \frac{\|\bar{y} - \hat{y}\|_2^2}{N^2}. \\
 (\text{mean absolute error}) \text{ MAE} &= \frac{\|\bar{y} - \hat{y}\|_1}{N^2}.
 \end{aligned}$$

Additionally, to measure the noisiness of a blurred image, we use the blurred SNR (BSNR), defined as

$$(\text{blurred SNR}) \text{ BSNR} = 20 \log_{10} \left(\frac{\|\bar{y}_{\text{blur}} - \text{mean}(\bar{y}_{\text{blur}})\|_2}{\|\bar{y}_{\text{blur}} - z\|_2} \right),$$

where \bar{y}_{blur} is the noise-free blurred image and z is the noisy blurred image.

1-D, 2-D, 3-D	One-, Two-, Three-Dimensional
AC	Alternating Current (non-constant components)
AWGN	Additive White Gaussian Noise
BSNR	Blurred Signal-to-Noise Ratio
B-DCT	Block Discrete Cosine Transform (e.g., 8×8 2-D DCT)
BM3D	Block-Matching 3-D transform domain modeling or filter
CCD	Charge-Coupled Device
CDF	Cumulative Distribution Function
dB	decibel
DC	Direct Current (constant component)
DCT	Discrete Cosine Transform
DST	Discrete Sine Transform
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
GNE	Generalized Nash equilibrium
ICI	Intersection of Confidence Intervals
i.i.d.	independent and identically distributed
ISNR	Improvement in Signal-to-Noise Ratio
LPA	Local Polynomial Approximation
LS	Least Squares
ML	Maximum Likelihood
MSE	Mean Squared Error
NLM	Non-local Means
PCA	Principal Component Analysis
PDF	Probability Density Function
PSF	Point-Spread Function
PSNR	Peak Signal-to-Noise Ratio
RI	Regularized Inverse
RWI	Regularized Wiener Inverse
SNR	Signal-to-Noise Ratio
SR	Super-resolution
std	standard deviation
TV	Total Variation
var	variance

Chapter 1

Sparsity-based image reconstruction

Many imaging problems belong to the class of *inverse problems*, where one aims to reconstruct the image from its indirect measurements. To recover the image, the inverse of the measurement operator should be applied to the measured data. The main difficulty of the inverse problems is that typically the measurement operator has no exact inverse, and hence the problem has no unique solution. Unfortunately not any solution can be considered as a reasonable estimate of the true image. But, the fact that we can differentiate whether solution is reasonable or not indicates that we have an *a-priori knowledge* of how the reasonable estimate should look like. This knowledge formalized in the form of mathematical *model* can be utilized as a constraint, to sieve the solution with desired properties from the set of all possible solutions. In fact, the problem of devising an efficient mathematical *image model* is the one of the main problems in image processing. While the general criteria for assessing image quality are well known (e.g. sharp edges, absence of artefacts, preservation of textures), their translation into mathematical language is not a trivial task. More, it is not enough to give the mathematical formulation of the criteria, the resulting mathematical problems should be tractable both in theory and in practice.

In this chapter we give an overview of the sparsity based methods used in image reconstruction problems. We present the ideas behind the sparsity based models, review the methods for solving inverse problems under sparsity constraints and discuss construction of effective dictionaries rendering sparse image models.

1.1 Sparsity-based models in inverse problems

One of the reasons which motivated use of the sparsity priors in imaging was the observation, that certain transforms (such as DFT, DCT and various types of wavelets) have energy compaction property for natural images. This means that

images can be reasonably well *approximated*¹ using only relatively small part of the basis elements. In other words, natural images can be *sparsely approximated* in these bases. On the other hand, unstructured signals, such as noise, cannot be sparsely approximated. For example, the energy of the AWGN is spread uniformly over all basis elements of any orthonormal transform. Hence, sparsity can be used as a criterion to differentiate the natural images from unstructured noise-like images.

1.1.1 Observation model

We consider a general *linear* inverse problem, where we try to estimate the original vector, from the set of its linear measurements corrupted by noise. Assuming noise to be additive white Gaussian (AWGN), we write the observation model as

$$\mathbf{z} = \mathbf{A}\bar{\mathbf{y}} + \boldsymbol{\eta} \quad (1.1)$$

where $\mathbf{z} \in \mathbb{R}^N$ is the vector of measurements, $\bar{\mathbf{y}} \in \mathbb{R}^N$ is the unknown true vector, $\boldsymbol{\eta} \sim \mathcal{N}(0_N, \sigma \mathbf{I}_{N \times N})$ is the noise and \mathbf{A} is the $N \times N$ matrix representing the *known* linear measurement (or degradation) operator. Additionally, we consider an overcomplete dictionary $\{\phi_i \in \mathbb{R}^N\}_{i=1}^M$, $M \geq N$ which we assume to provide sparse approximation of the vector $\bar{\mathbf{y}}$. The analysis and synthesis operators (matrices) corresponding to the dictionary $\{\phi_i\}_{i=1}^M$ we denote respectively by Φ and Ψ . Unless different is stated, we assume the synthesis operator to be defined as the pseudoinverse of the analysis operator $\Psi = (\Phi^T \Phi)^{-1} \Phi^T$.

The sparsity prior can be incorporated in the formulation of the reconstruction problem in a number of different ways. Below we describe several such possibilities, starting from the two most basic ones: the *analysis-* and *synthesis-based* formulations.

1.1.2 Analysis-based formulation

The analysis based formulation states, that among all candidate solutions explaining the observation \mathbf{z} with high enough probability, we should take one having sparsest representation in the dictionary $\{\phi_i\}_{i=1}^M$.

Applied to the problem (1.1), the analysis formulation takes the form:

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}} \|\Phi \mathbf{y}\|_0 \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2 \leq \epsilon, \quad (1.2)$$

where the constraint follows from the condition that the likelihood of observing \mathbf{z} in the model (1.1) for the particular \mathbf{y} should be higher than a certain value δ

$$p(\mathbf{z}|\mathbf{y}) = \frac{1}{(2\pi)^{N/2} \sigma^N} \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2\right) > \delta. \quad (1.3)$$

¹We are going to differentiate terms "approximation" and "representation". We use *approximation*, for approximation with nonzero error. To indicate exact approximation, without error, we use term *representation*.

Often, it is more convenient to consider a similar formulation, which instead involves unconstrained optimization

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}} \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2 + \tau \|\Phi\mathbf{y}\|_0. \quad (1.4)$$

Strictly speaking (1.4) and (1.2) are not equivalent since l_0 -norm is not a convex functional. Nevertheless, both formulations provide qualitatively similar results promoting sparse solutions.

Problem (1.4) can be interpreted as finding Maximum-A-posteriori-Probability (MAP) estimate

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} p(\mathbf{y}|\mathbf{z}).$$

Starting from the improper prior

$$p(\mathbf{y}) = \exp(-\tau \|\Phi\mathbf{y}\|_0), \tau > 0,$$

and taking into account formula (1.3), which gives the conditional distribution $p(\mathbf{z}|\mathbf{y})$, we define the posterior probability of \mathbf{y} as

$$p(\mathbf{y}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{z})}.$$

Ignoring $p(\mathbf{z})$, which is independent of \mathbf{y} , and taking negative of the logarithm of $p(\mathbf{z}|\mathbf{y})p(\mathbf{y})$ we obtain

$$\begin{aligned} \hat{\mathbf{y}} &= \arg \max_{\mathbf{y}} \frac{p(\mathbf{z}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{z})} = \arg \min_{\mathbf{y}} \{-\log(p(\mathbf{z}|\mathbf{y})p(\mathbf{y}))\} \\ &= \arg \min_{\mathbf{y}} \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2 + \tau \|\Phi\mathbf{y}\|_0, \end{aligned}$$

which is exactly our problem (1.4).

1.1.3 Synthesis-based formulation

Let $\boldsymbol{\omega} \in \mathbb{R}^M$ be a vector of coefficients generating a signal $\Psi\boldsymbol{\omega} \in \mathbb{R}^N$. The synthesis based formulation states, that we should search for the sparsest $\boldsymbol{\omega}$, among those whose generated signals $\Psi\boldsymbol{\omega}$ have sufficiently high probability to explain the observation \mathbf{z} :

$$\hat{\mathbf{y}} = \Psi \arg \min_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\|_0 \quad \text{subject to} \quad \|\mathbf{z} - \mathbf{A}\Psi\boldsymbol{\omega}\|_2^2 \leq \epsilon. \quad (1.5)$$

Similar to the analysis case, there is an unconstrained optimization formulation

$$\hat{\mathbf{y}} = \Psi \arg \min_{\boldsymbol{\omega}} \left\{ \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\Psi\boldsymbol{\omega}\|_2^2 + \tau \|\boldsymbol{\omega}\|_0 \right\}. \quad (1.6)$$

From the first glance the analysis- and synthesis-based formulations may look identical, but in fact, this is true only in the case when $M = N$ and $\{\phi_i\}_{i=1}^M$ is an orthonormal basis. In the general case, the difference is easy to spot, since in the analysis-based formulation optimization is performed over the N -dimensional space, while in the synthesis formulation the search space is M -dimensional. A detailed discussion of the nontrivial connections between these two formulations can be found in [EMR07].

1.1.4 Relaxed analysis formulation

Portilla [Por09] suggested an extension of the analysis based model (1.4), motivated by the observation that wavelet representations of the most of the natural images produce *compressible* rather than sparse coefficient vectors. The vector is called compressible if it can be represented as a sum of a sparse vector plus a non-sparse residual which has small energy. Following the compressibility assumption, \mathbf{y} is represented as

$$\Phi \mathbf{y} = \boldsymbol{\omega} + \mathbf{r},$$

where $\boldsymbol{\omega} \in \mathbb{R}^M$ is the sparse component and $\mathbf{r} \in \mathbb{R}^M$ is the non-sparse residual. In general, $\boldsymbol{\omega}$ and \mathbf{r} are not independent, but in order to simplify the modeling they are assumed to be such. Moreover, it is assumed that \mathbf{r} is distributed as zero mean white Gaussian noise with variance σ_r . While these assumptions are arguable, they make the resulting reconstruction problem easier to solve, yet benefiting from the extra flexibility of the model compared to (1.4). Thus, we are given the following prior distributions:

$$\begin{aligned} p(\mathbf{z}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2\right), \\ p(\mathbf{y}|\boldsymbol{\omega}) &\propto \exp\left(-\frac{1}{2\sigma_r^2} \|\boldsymbol{\omega} - \Phi \mathbf{y}\|_2^2\right), \\ p(\boldsymbol{\omega}) &\propto \exp(-\tau \|\boldsymbol{\omega}\|_0). \end{aligned}$$

Here formula for $p(\mathbf{y}|\boldsymbol{\omega})$ follows from gaussianity of \mathbf{r} , while $p(\boldsymbol{\omega})$ reflects the assumption of $\boldsymbol{\omega}$ being sparse. Noting that $p(\mathbf{z}|\mathbf{y}, \boldsymbol{\omega}) = p(\mathbf{z}|\mathbf{y})$ and defining $p(\mathbf{y}, \boldsymbol{\omega}) = p(\mathbf{y}|\boldsymbol{\omega})p(\boldsymbol{\omega})$ we obtain the expression for the joint PDF

$$\begin{aligned} p(\mathbf{y}, \boldsymbol{\omega}, \mathbf{z}) &= p(\mathbf{z}|\mathbf{y}, \boldsymbol{\omega}) p(\mathbf{y}, \boldsymbol{\omega}) \\ &= p(\mathbf{z}|\mathbf{y}) p(\mathbf{y}, \boldsymbol{\omega}) = p(\mathbf{z}|\mathbf{y}) p(\mathbf{y}|\boldsymbol{\omega}) p(\boldsymbol{\omega}). \end{aligned}$$

The reconstruction is formulated as a problem to find a pair of vectors $(\hat{\mathbf{y}}, \hat{\boldsymbol{\omega}})$ which will maximize the joint probability $p(\mathbf{y}, \boldsymbol{\omega}, \mathbf{z})$.

$$(\hat{\mathbf{y}}, \hat{\boldsymbol{\omega}}) = \arg \max_{\mathbf{y}, \boldsymbol{\omega}} p(\mathbf{y}, \boldsymbol{\omega}, \mathbf{z}) = \arg \min_{\mathbf{y}, \boldsymbol{\omega}} \tau \|\boldsymbol{\omega}\|_0 + \frac{1}{2\sigma_r^2} \|\Phi \mathbf{y} - \boldsymbol{\omega}\|_2^2 + \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2. \quad (1.7)$$

The last problem can be interpreted from slightly different view, better illustrating its relation to the analysis formulation (1.4). We start from (1.4) performing variable splitting, i.e., substituting $\boldsymbol{\omega} \triangleq \Phi \mathbf{y}$ and considering $\boldsymbol{\omega}$ to be an independent variable. Then the constrained minimization with respect to the variables \mathbf{y} and $\boldsymbol{\omega}$

$$(\hat{\mathbf{y}}, \hat{\boldsymbol{\omega}}) = \arg \min_{\mathbf{y}, \boldsymbol{\omega}} \tau \|\boldsymbol{\omega}\|_0 + \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2, \text{ subject to } \boldsymbol{\omega} = \Phi \mathbf{y}, \quad (1.8)$$

is an equivalent form of the problem (1.4).

The (1.7) then can be viewed as an analysis formulation (1.8) where the equality constraint is relaxed being replaced by the inequality constraint

$$(\hat{\mathbf{y}}, \hat{\boldsymbol{\omega}}) = \arg \min_{\mathbf{y}, \boldsymbol{\omega}} \tau \|\boldsymbol{\omega}\|_0 + \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2, \text{ subject to } \|\boldsymbol{\omega} - \Phi \mathbf{y}\|_2^2 \leq \delta.$$

In the similar manner one can construct also a relaxed synthesis model

$$(\hat{\mathbf{y}}, \hat{\boldsymbol{\omega}}) = \arg \min_{\mathbf{y}, \boldsymbol{\omega}} \tau \|\boldsymbol{\omega}\|_0 + \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2, \text{ subject to } \|\boldsymbol{\Psi}\boldsymbol{\omega} - \mathbf{y}\|_2^2 \leq \delta.$$

1.2 Methods for solving sparse inverse problems

How to solve problems formulated above? Presence of the l_0 -norm, which is non-differentiable and non-convex function, restricts direct application of most of the standard optimization techniques. The naive approach, using combinatorial search is computationally too complex, which limits its use only to the cases when the dictionary size is relatively small $M < 10^2$. At the same time, the typical dimensionality of the dictionaries in the imaging are $M \sim 10^4$ - 10^7 . Solving problems of such size poses also technical problems: on most of the current computers directly storing dictionary matrices in the computer memory is practically impossible. Considered problems stimulated development of new specialized methods for large scale problem solving. These methods can be roughly divided in three classes: greedy search algorithms, algorithms based on convex relaxation of the problems and algorithms based on iterative shrinkage.

Greedy algorithms were the first ones proposed to find sparse solutions in different optimization problems. The Matching Pursuit (MP) [MZ93] and its extensions, Orthogonal Matching Pursuit (OMP) [PRRK93] and Stagewise OMP [DTDS12], are the key algorithms of this class.

The greedy algorithm performs iteratively. It starts from the null vector as the initial approximation. At each iteration the support of the approximation vector is extended with those dictionary elements showing highest correlation with the approximation error of the previous iteration. The process stops when the approximation error drops below a certain level or complexity of the approximation reaches the maximum allowed limit.

The main drawback of the greedy algorithms is that being suboptimal they may stack at the local minima. This is particularly an issue when the complexity of the solution increases. Another weak point of the greedy algorithms is their slow convergence speed.

Convex relaxation. Convex relaxation for sparse vector approximation problems was introduced independently in signal processing [CDS01] as Basis Pursuit Denoising (BPD) problem and in the statistical estimation theory [Tib96] as Least Absolute Shrinkage and Selection Operator (LASSO). The idea of this approach is to replace the non-convex l_0 -norm with l_1 -norm, which is the closest to l_0 convex l_p -norm that promotes sparsity. The resulting l_1 - l_2 problems:

$$\arg \min_{\mathbf{y}} \frac{1}{2\sigma^2} \|\mathbf{z} - \mathbf{y}\|_2^2 + \tau \|\Phi\mathbf{y}\|_1 \quad (1.9)$$

and

$$\arg \min_{\boldsymbol{\omega}} \frac{1}{2\sigma^2} \|\mathbf{z} - \boldsymbol{\Psi}\boldsymbol{\omega}\|_2^2 + \tau \|\boldsymbol{\omega}\|_1 \quad (1.10)$$

corresponding respectively to the analysis- and synthesis-based formulations (1.4) and (1.6), can be solved with the standard convex optimization tools, such as

steepest descent, conjugate gradient or interior-point methods. Convexity of the criterion function ensures that the optimum of the problem is unique and global.

Relaxed problems in the general case are not equivalent to their counterparts based on l_0 -norm. Equivalence holds under certain conditions (see e.g. [Tro06]), which, however, are quite restrictive.

Iterative-shrinkage algorithms. The methods based on iterative shrinkage (IS) constitute a broad class of algorithms (e.g., [FN03], [DDDM04], [FN05], [CW05], [FBDN07] and many others) which are known for their remarkable convergence speed for large scale l_p - l_2 ($p \leq 1$) problems. While being derived from different considerations all IS algorithms share two common operations: shrinkage (thresholding) and projection on the subspace defined by the range of the analysis operator associated with the dictionary.

The classical example of the IS type method, is the Iterative Soft Thresholding (IST) algorithm for solving (1.10) [DDDM04], [FN05]. It is given by the recursive formula

$$\boldsymbol{\omega}_{t+1} = Th_{\sigma^2\tau/c}^{soft} \left(\frac{1}{c} \boldsymbol{\Phi} \left(\mathbf{z} - \boldsymbol{\Phi}^T \boldsymbol{\omega}_t \right) + \boldsymbol{\omega}_t \right), \quad (1.11)$$

where c is the step size and Th is the shrinkage operator. Shrinkage operator is the solution of the sparse approximation problem

$$Th_{\tau}(\mathbf{x}) = \arg \min_{\boldsymbol{\omega}} \tau \|\boldsymbol{\omega}\|_p + \frac{1}{2} \|\boldsymbol{\omega} - \mathbf{x}\|_2^2, \quad (1.12)$$

where $\mathbf{x} \in \mathbb{R}^M$ is the vector being approximated. In two particular cases: $p = 0$ and $p = 1$, problem (1.12) has close form solutions, given respectively by the hard and soft thresholding formulas

$$Th_{\tau}^{hard}(\mathbf{x}) = \begin{cases} x_i, & |x_i| > \sqrt{2\tau} \\ 0, & |x_i| \leq \sqrt{2\tau} \end{cases}, i = 1, \dots, M, \quad (1.13)$$

$$Th_{\tau}^{soft}(\mathbf{x}) = \begin{cases} \text{sign}(x_i)(|x_i| - \tau), & |x_i| > \tau \\ 0, & |x_i| \leq \tau \end{cases}, i = 1, \dots, M. \quad (1.14)$$

Hence, at each iteration (1.11) the prediction error $\mathbf{z} - \boldsymbol{\Phi}^T \boldsymbol{\omega}_t$ is projected to the range of analysis operator $\boldsymbol{\Phi}$, scaled by the step size parameter and added to the previous estimate $\boldsymbol{\omega}_t$. The sum is then thresholded to obtain sparser estimate. For $p = 1$, the IST algorithm is proven to converge to the global minimum of the problem (1.10). Convergence properties for $p = 0$ are studied in [BD08] and [BD09].

1.3 Dictionaries for sparse image approximations

So far we concentrated on the formulations for sparse recovery problems and methods to solve them, circumventing the main question: how to construct a dictionary that will render the sparse approximations?

Earlier we mentioned that some of the well known bases have energy compactation property. Despite this fact, representations of many natural images with these bases are not sparse. In general, the number of elements in the basis is too limited to generate sparse representations for wide classes of images, particularly for

natural images. One needs to consider redundant dictionaries with a number of elements essentially larger than the dimensionality of the approximated images.

The dictionaries used image processing can be roughly divided into three groups according to their construction methods:

- non-adaptive or fixed,
- learned dictionaries,
- adaptive dictionaries which are based on fixed transforms.

1.3.1 Non-adaptive dictionaries

The non-adaptive (fixed) transforms are typically designed to possess particular mathematical properties and are targeted to model certain types of structures in signals. E.g., Haar wavelets are good for representing sharp discontinuities, contourlets effectively represent smooth curves, Block DCT transform is known to well represent textures, etc. Such a targeted design limits the ability of fixed transforms to sparsify signals, restricting it to the target class only. Since in general natural images contain structures of different types, edges, textures, smooth areas, use of fixed transforms can be considered as a suboptimal choice.

Nevertheless, despite apparent drawbacks, fixed redundant transforms such as undecimated Haar wavelets, Dual Tree Complex Wavelets, Curvelets, Contourlets and others find extensive use in publications, where authors concentrate on reconstruction algorithms rather than effective sparse representations (e.g., [PSWS03], [ABDF10]). Known mathematical properties and availability of efficient implementations make them convenient tools for testing various algorithms.

To mitigate the limitations of individual fixed dictionaries, it has been proposed to use dictionaries made by merging several fixed dictionaries having complementary features. Such, Portilla demonstrated [Por09] that for images with a moderate amount of texture, using combination of Dual Tree Complex Wavelets (DTCW) [Kin01] and Translation Invariant Haar Pyramid (TIHT) [GCMP08] can significantly improve reconstruction quality compared to the cases when DTCW or TIHT are used alone.

1.3.2 Learned dictionaries

The fail of attempts to construct a universal "all purpose" dictionary synthetically, from mathematical models, stimulated the research in the direction of constructing dictionaries analytically, by learning them from natural images. Given the training set of sample images, the goal of the learning process is to construct a dictionary which will sparsely approximate images from the training set. The dictionary learning can be posed as an optimization problem:

$$\min_{\Psi, \{\omega_k\}_{k=1}^K} \sum_k \|\omega_k\|_0, \text{ subject to } \|\mathbf{y}_k - \Psi \omega_k\|_2^2 \leq \epsilon, \quad k = 1, \dots, K, \quad (1.15)$$

where $\{\mathbf{y}_k \in \mathbb{R}^n\}_{k=1}^K$ is the training set of K image patches of size $\sqrt{n} \times \sqrt{n}$, and we are looking for an $M \ll K$ element dictionary Ψ that will minimize the complexity of the approximation coefficients $\{\omega_k\}_{k=1}^K$.

There are number of heuristic algorithms proposed to solve (1.15). Here we briefly mention the two main ones. First algorithm, named Method of Optimal Directions (MOD) [EAH00] solves (1.15), by performing alternating optimization with respect to the dictionary Ψ and approximation vectors $\{\omega_k\}_{k=1}^K$. The dictionary update step is a simple quadratic optimization problem:

$$\begin{aligned}\Psi_{(t+1)} &= \arg \min_{\Psi} \sum \left\| \mathbf{y}_k - \Psi \omega_k^{(t)} \right\|_2^2 \\ &= \arg \min_{\Psi} \sum \left\| \mathbf{Y} - \Psi \Omega_{(t)} \right\|_F^2 = \mathbf{Y} \Omega_{(t)}^T \left(\Omega_{(t)} \Omega_{(t)}^T \right)^{-1},\end{aligned}$$

where \mathbf{Y} and $\Omega_{(t)}$ are matrices made respectively of column vectors \mathbf{y}_k and $\omega_k^{(t)}$, and $\|\cdot\|_F$ is the Frobenius norm. The minimizations with respect to the individual approximation vectors ω_k are sparse synthesis problems

$$\omega_k^{(t+1)} = \arg \min_{\omega_k} \|\omega_k\|_0, \text{ subject to } \left\| \mathbf{y}_k - \Psi_{(t)} \omega_k \right\|_2^2 \leq \epsilon, 1 \leq k \leq K$$

which are solved with pursuit algorithms.

The more advanced learning technique K-SVD [AEB06], also relies on alternating optimization, but unlike MOD at the dictionary update step the atoms are updated sequentially. The update of each atom is done by removing the atom from the dictionary, forming a matrix of approximation errors vectors and then applying SVD to find the new atom which will minimize the error while keeping representation vectors sparse.

Using MOD and K-SVD, dictionaries can be trained both from uncorrupted set of images, as well from distorted, for example, noisy images [AEB06].

The learned dictionaries have also some drawbacks:

- Unlike most of the fixed dictionaries (such as FFT, DCT, x-lets), the learned dictionaries are unstructured. As a result, there are no fast implementations for performing analysis/synthesis with learned dictionaries, the dictionaries are stored and operated explicitly as big matrices.
- Dictionary training is a computationally intensive task. It is tractable only for relatively small image patches $\sqrt{n} \leq 30$. The resulting dictionaries cannot be applied to the whole image. Nevertheless, this limitation does not seem to be critical for image restoration, since we can always consider image as a linear combination of patches.

1.3.3 Adaptive dictionaries based on fixed transforms

These dictionaries can be considered as an intermediate step between fixed and learned dictionaries. The typical adaptive dictionary emerges from smaller size fixed subdictionaries combined in a data adaptive manner. The adaptation typically follows either low level signal features, such as direction, scale and shape (e.g. Shape-Adaptive DCT [FKE07], Local Polynomial Approximations with Intersection of Confidence Intervals (LPA-ICI) [Foi05], kernel regression [TFM07]),

or higher level features, such as self-similarity (Non-local Means [BCM05], Block-Matching and 3-D filtering [DFKE07], [DKE12])².

Applying fixed subdictionaries adaptively allows to obtain sparser signal representation compared to one obtained with fixed dictionaries. At the same time, unlike learned dictionaries, adaptive dictionaries do not need to be stored and manipulated as huge matrices. The analysis and synthesis operations with adaptive dictionaries are performed by applying small size subdictionaries in the order defined by the data adaptation rule. Additionally, the speed of analysis/synthesis operations can essentially benefit from existence of fast computation schemes for the subdictionaries.

It has been demonstrated [KFEA10] that the best adaptive dictionaries, such as one in Block-Matching and 3-D filtering can provide restoration quality very close to those provided by learned dictionaries.

1.4 Block-matching based 3-D transform domain image modeling

In this thesis we are going to consider methods for sparse image reconstruction with the particular type of adaptive dictionaries, which emerges from the image modeling known as Block-Matching based 3-D transform domain modeling (BM3D) [DFKE07]. BM3D is a nonlocal image modelling technique based on adaptive high-order groupwise models. This technique is well known for its ability to provide highly sparse, redundant image/video representations. The detailed discussion of the BM3D modelling can be found in [DFKE07], [KFEA10] or [Dab10]. Here we briefly recall its main concepts.

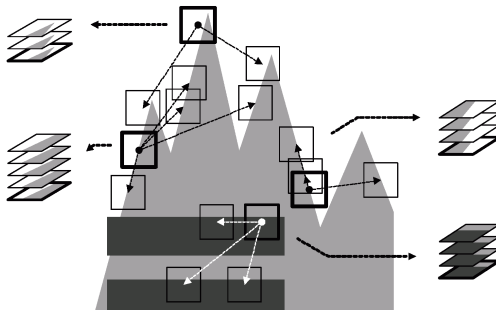


Figure 1.1: Illustration of grouping in an artificial image.

The modeling is split in two steps: analysis and synthesis.

Analysis. Similar image blocks found in the image are collected in groups. Blocks in each group are stacked together to form 3-D data arrays (see Fig. 1.1), which are decorrelated using an invertible 3D transform.

²The Local Polynomial Approximations, Kernel Regression and Non-local Means image modellings do not assume in any form that the images can be sparsely approximated with the corresponding dictionaries. Those dictionaries are mentioned in the text solely as important examples of data-adaptive, redundant dictionaries.

The blocking imposes a localization of the image on small pieces where simpler models may fit the observations. It has been demonstrated that a higher sparsity of the signal representation and a lower complexity of the model can be achieved using joint groupwise 3D transforms instead of conventional blockwise 2D transforms [Dab10].

In what follows, we refer to the union of all group spectra elements as the *groupwise spectrum*. Since blocks overlap, the number of elements in the groupwise spectrum is much larger than the image size. Thus, BM3D analysis provides an *overcomplete* data representation.

Synthesis. The inverse transform is applied to each group spectrum, providing estimates for each block in the group. These blockwise estimates are returned to their original positions, and the final image reconstruction is calculated as a weighted average of all of the obtained estimates, a procedure known as *aggregation*.

1.4.1 BM3D-frames

Let \mathbf{Y} be a $\sqrt{N} \times \sqrt{N}$ array representing the image and \mathbf{y} be the corresponding \mathbb{R}^N -vector built from the columns of \mathbf{Y} . To each $\sqrt{N_{bl}} \times \sqrt{N_{bl}}$ square image block we assign a unique index equal to the index of its upper-left corner element (pixel) in \mathbf{y} . We denote a vector of elements of j -th block \mathbf{Y}_j by \mathbf{y}_j and define \mathbf{P}_j as an $N_{bl} \times N$ matrix of indicators $[0, 1]$ showing which elements of \mathbf{y} belong to the j -th block, so that $\mathbf{y}_j = \mathbf{P}_j \mathbf{y}$. For the sake of a notational simplicity, we assume that the number of blocks in each group is fixed and equal to K . Let $J_r = \{j_{r,1}, \dots, j_{r,K}\}$ be the set of indices of the blocks in the r -th group, then grouping is completely defined by the set $J = \{J_r : r = 1, \dots, R\}$, where R is a total number of the groups. It is assumed that for each pixel there is at least one block which enters in a group and contains the pixel. The final image estimate is defined as the weighted mean of the groupwise estimates using weights $g_r > 0$.

It has been shown [DKE12], that for a fixed grouping the BM3D analysis/synthesis operations can be given in the matrix form linking the image \mathbf{y} and its groupwise spectrum vector $\boldsymbol{\omega} \in \mathbb{R}^M$ by the forward and backward transforms

$$\boldsymbol{\omega} = \boldsymbol{\Phi} \mathbf{y}, \mathbf{y} = \boldsymbol{\Psi} \boldsymbol{\omega}.$$

Here $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ represent respectively the analysis and synthesis operators.

Proposition 1 [DKE12]. *The following equations hold for the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$:*

$$\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \sum_r \sum_{j \in J_r} \mathbf{P}_j^T \mathbf{P}_j > 0, \quad (1.16)$$

$$\boldsymbol{\Psi} \boldsymbol{\Psi}^T = \sum_r g_r^2 \sum_{j \in J_r} \mathbf{P}_j^T \mathbf{P}_j \mathbf{W}^{-2} > 0, \quad (1.17)$$

$$\boldsymbol{\Psi} \boldsymbol{\Phi} = \mathbf{I}_{N \times N}, \quad (1.18)$$

where $\mathbf{W} = \sum_r g_r \sum_{j \in J_r} \mathbf{P}_j^T \mathbf{P}_j$.

As it follows from Proposition 1, the rows of Φ constitute a frame $\{\phi_n\}$ in \mathbb{R}^N . Indeed, let us verify the frame inequality. Using the analysis formula $\omega = \Phi \mathbf{y}$ and (1.16) we obtain

$$\sum_n |\langle \phi_n, \mathbf{y} \rangle|^2 = \omega^T \omega = \mathbf{y}^T \Phi^T \Phi \mathbf{y} = \mathbf{y}^T \sum_r \sum_{j \in I_r} \mathbf{P}_j^T \mathbf{P}_j \mathbf{y}.$$

If a and b are respectively minimum and maximum values of the diagonal matrix $\sum_r \sum_{j \in I_r} \mathbf{P}_j^T \mathbf{P}_j$, then for any $\mathbf{y} \in \mathbb{R}^N$ holds the frame inequality

$$a \|\mathbf{y}\|^2 \leq \sum_n |\langle \phi_n, \mathbf{y} \rangle|^2 \leq b \|\mathbf{y}\|^2. \quad (1.19)$$

The frame $\{\phi_n\}$ is not tight because, in general, $a \neq b$. This follows from the fact that the (j, j) element on the diagonal of the matrix $\sum_r \sum_{j \in I_r} \mathbf{P}_j^T \mathbf{P}_j$ is equal to the number of grouped blocks containing the j -th pixel. For different pixels this numbers are different, since pixels from the blocks possessing higher similarity to other blocks participate in a larger number of groups.

Similarly, using (1.17) we can show that columns of Ψ constitute a non-tight frame $\{\psi_n\}$. It follows from equation (1.18) that $\{\phi_n\}$ is dual to $\{\psi_n\}$. In general, $\Psi \neq (\Phi^T \Phi)^{-1} \Phi^T$ and $\{\psi_k\}$ is an *alternative dual* frame due to the presence of the group weights. The equality $\Psi = (\Phi^T \Phi)^{-1} \Phi^T$ takes place only when all weights g_r are equal. $\{\psi_k\}$ then becomes the *canonical dual* frame³.

We would like to emphasize that since groups and weights are selected data adaptively, the constructed frames are also data adaptive.

³In this thesis, dealing with frames, we follow the terminology used in the book by Christensen [Chr03].

Chapter 2

Denoising with non-tight frames: from Basis Pursuit Denoising to generalized Nash equilibrium

The goal of this chapter is to demonstrate the motivation and main ideas laying behind the image reconstruction algorithms we propose in the next chapters.

First, on the example of BP denoising problem, we show that, for non-tight frames sparsity promoting models based on a single objective minimization lead to the reconstructions were the strength of the regularization varies for different spatial positions. Then, we propose an alternative sparsity promoting model based on generalized Nash equilibrium (GNE), develop the algorithm for solving it, and show its effectiveness in the simulated experiments.

2.1 Basis Pursuit denoising with non-tight frames

2.1.1 Problem formulation

Let

$$\{\phi_m : \phi_m \in \mathbb{R}^N\}_{m=1}^M, M > N \quad (2.1)$$

be a *non-tight* frame in \mathbb{R}^N , $\Phi : \mathbb{R}^N \rightarrow \mathbb{R}^M$ and $\Psi : \mathbb{R}^M \rightarrow \mathbb{R}^N$ are, respectively, the analysis and synthesis operators associated with the frame. We assume that the frame is such that $\Phi^T \Phi$ is a diagonal matrix¹. Notice, that since the frame is assumed to be non-tight, at least one diagonal element should differ from the others: $\nexists \alpha$ s.t. $\Phi^T \Phi = \alpha \mathbf{I}_{N \times N}$. Unless otherwise stated, we assume, that Ψ corresponds to the canonical dual frame of $\{\phi_m\}_{m=1}^M$ and hence it is defined as the pseudoinverse of Φ , i.e.

$$\Psi = (\Phi^T \Phi)^{-1} \Phi^T.$$

¹Particularly this always holds for BM3D-frames see (1.16) in Proposition 1.

We consider the analysis based formulation of the Basis Pursuit (BP) denoising problem (1.9)

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}} \frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 + \tau \|\Phi \mathbf{y}\|_1, \quad (2.2)$$

where $\mathbf{z} \in \mathbb{R}^N$ is the noisy observation, $\hat{\mathbf{y}} \in \mathbb{R}^N$ is the solution, and $\tau \in \mathbb{R}^+$ is a regularization parameter. In this chapter we limit ourselves, considering only the analysis based formulation, nevertheless the qualitative results we are going to obtain stay valid also for the synthesis based formulation.

2.1.2 ADMM algorithm for BP denoising problem

While we could solve problem (2.2) with one of the standard algorithms mentioned in the previous chapter, we prefer to derive the algorithm based on variable splitting and Augmented Lagrangian techniques. The resulting algorithm will be an instance of the so-called alternating direction method of multipliers (ADMM) [EB92]. We prefer the ADMM-type algorithm, since its structure makes easier understanding the problems arising due to the use of non-tight frames in reconstruction.

Our derivation will repeat the derivation of the SALSA algorithm [ABDF10]. Instead of solving (2.2) directly, we transform it into constrained optimization problem of two variables. We introduce new auxiliary variable $\boldsymbol{\omega} \in \mathbb{R}^M$, and reformulate (2.2) as

$$\hat{\mathbf{y}} = \arg \min_{\mathbf{y}, \boldsymbol{\omega}} \frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 + \tau \|\boldsymbol{\omega}\|_1 \quad \text{subject to } \boldsymbol{\omega} = \Phi \mathbf{y}. \quad (2.3)$$

The idea behind this approach, known as *variable splitting*, is that minimization of the split problem can be done by alternately minimizing (2.3) with respect to the one of the variables.

The standard constrained optimization techniques, such as Quadratic Penalty or Augmented Lagrangian (AL) can be applied [NW06] to solve (2.3). We confine ourselves to the AL technique, which is nowadays widely used for minimization of convex functionals under linear-equality constraints ([ABDF10], [ABDF11]).

The AL criterion for (2.3) takes the form:

$$L_a(\mathbf{y}, \boldsymbol{\omega}, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 + \tau \cdot \|\boldsymbol{\omega}\|_1 + \frac{\gamma}{2} \|\boldsymbol{\omega} - \Phi \mathbf{y}\|_2^2 + \gamma \langle \boldsymbol{\omega} - \Phi \mathbf{y}, \boldsymbol{\lambda} \rangle, \quad (2.4)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^M$ is a vector of the Lagrange multipliers, $\gamma > 0$ is a parameter and the subscript 'a' of L_a indicates the analysis formulation. The saddle-point problem

$$\left(\hat{\mathbf{y}}, \hat{\boldsymbol{\omega}}, \hat{\boldsymbol{\lambda}} \right) = \arg \min_{\mathbf{y}, \boldsymbol{\omega}} \max_{\boldsymbol{\lambda}} L_a(\mathbf{y}, \boldsymbol{\omega}, \boldsymbol{\lambda}), \quad (2.5)$$

associated with the Lagrangian L_a , provides the solution of the constrained optimization problem (2.3) [NW06].

Finding the saddle point requires joint minimization of L_a with respect to the variables $\mathbf{y}, \boldsymbol{\omega}$ and maximization with respect to $\boldsymbol{\lambda}$. The idea behind the

ADMM method is that this joint optimization can be replaced by the alternating optimization with respect to the variables \mathbf{y} , $\boldsymbol{\omega}$ and $\boldsymbol{\lambda}$. Applied to (2.5), it results in the following iterative scheme:

Repeat for $t = 1, 2, \dots$

$$\mathbf{y}_t = \arg \min_{\mathbf{y}} L_a(\mathbf{y}, \boldsymbol{\omega}_{t-1}, \boldsymbol{\lambda}_{t-1}), \quad (2.6)$$

$$\boldsymbol{\omega}_t = \arg \min_{\boldsymbol{\omega}} L_a(\mathbf{y}_t, \boldsymbol{\omega}, \boldsymbol{\lambda}_{t-1}), \quad (2.7)$$

$$\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} + (\boldsymbol{\omega}_t - \Phi \mathbf{y}_t), \quad (2.8)$$

until convergence.

Here maximization with respect to $\boldsymbol{\lambda}$ is produced as a step (2.8) in the direction of the gradient $\nabla_{\boldsymbol{\lambda}} L_a$.

Solutions of (2.6) and (2.7) can be easily obtained. Indeed, since L_a is quadratic with respect to \mathbf{y} , the solution of (2.6) is given by the formula

$$\mathbf{y}_t = (\mathbf{I} + \gamma \Phi^T \Phi)^{-1} (\mathbf{z} + \gamma \Phi^T (\boldsymbol{\omega}_{t-1} + \boldsymbol{\lambda}_{t-1})). \quad (2.9)$$

To solve (2.7), we first regroup terms in L_a

$$L_a(\mathbf{y}_t, \boldsymbol{\omega}, \boldsymbol{\lambda}_{t-1}) = \frac{1}{2} \|\mathbf{z} - \mathbf{y}_t\|_2^2 + \tau \|\boldsymbol{\omega}\|_1 + \frac{\gamma}{2} \|\boldsymbol{\omega} - (\Phi \mathbf{y}_t - \boldsymbol{\lambda}_{t-1})\|_2^2 - \frac{\gamma}{2} \|\boldsymbol{\lambda}_{t-1}\|_2^2.$$

Since the first and the last terms do not depend on $\boldsymbol{\omega}$, problem (2.6) reduces to the optimization

$$\boldsymbol{\omega}_t = \arg \min_{\boldsymbol{\omega}} \tau \|\boldsymbol{\omega}\|_1 + \frac{\gamma}{2} \|\boldsymbol{\omega} - (\Phi \mathbf{y}_t - \boldsymbol{\lambda}_{t-1})\|_2^2, \quad (2.10)$$

which has well known analytical solution given by the soft thresholding operator (1.14):

$$\boldsymbol{\omega}_t = Th_{\tau/\gamma}^{soft}(\Phi \mathbf{y}_t - \boldsymbol{\lambda}_{t-1}).$$

Following (2.6)-(2.8) and using (2.9) and (2.10) we define the Algorithm 1 for solving BP denoising problem. In each iteration it first updates the image estimate using the linear filtering (2.9). Then, the difference between the spectrum $\Phi \mathbf{y}_t$ and $\boldsymbol{\lambda}_{t-1}$ is thresholded. Finally, the Lagrange multipliers are updated in the direction of the gradient $\boldsymbol{\omega}_t - \Phi \mathbf{y}_t$. Process is iterated until some convergence criterion is satisfied.

2.1.3 Convergence

The main motivation of the AL technique is to replace a constrained optimization with a simpler saddle-point problem. The equivalence of these two problems is not a given fact. The classical results stating equivalence are formulated for the convex and differentiable functions [Ber96]. Since l_p -norms with $p \leq 1$ are non-differentiable these results are inapplicable. Nevertheless, for the l_1 -norm the equivalence can be shown, provided that the constraints in the problem are linear. In the recent paper [TW09] the equivalence statement has been proven for the total variation penalty. This proof remains valid for any convex and non-differentiable

Algorithm 1 ADMM-type algorithm for analysis-based BP denoising.

1: **input:** $\mathbf{z}, \sigma, \tau, \gamma$
2: *construct* Φ using \mathbf{z} or any initial estimate \mathbf{y}_{init}
3: **initialization:**
4: set: $t = 0, \mathbf{y}_0 = \mathbf{z}, \omega_0 = \Phi\mathbf{z}, \lambda_0 = 0$
5: **repeat**
6: $t = t + 1$
7: $\mathbf{y}_t = (\mathbf{I} + \gamma\Phi^T\Phi)^{-1} (\mathbf{z} + \gamma\Phi^T(\omega_{t-1} + \lambda_{t-1}))$
8: $\omega_t = Th_{\tau/\gamma}^{soft}(\Phi\mathbf{y}_t - \lambda_{t-1})$
9: $\lambda_t = \lambda_{t-1} + (\omega_t - \Phi\mathbf{y}_t)$
10: **until** convergence
11: **output:** $\hat{\mathbf{y}} = \mathbf{y}_t$ (or equivalently $\hat{\mathbf{y}} = (\Phi^T\Phi)^{-1} \Phi^T\omega_t$)

penalties, in particular, for the l_1 -norm based penalties. For the considered analysis-based problem, the equivalence is stated in the following form:

$(\hat{\mathbf{y}}, \hat{\omega})$ is a solution of the analysis problem (2.3), if and only if, there exists a saddle point $(\hat{\mathbf{y}}, \hat{\omega}, \hat{\lambda})$ (2.5) of the AL (2.4).

Practically it means that the saddle-point of the AL optimization can be used in order to obtain the solutions of the considered constrained optimization problems.

The convergence properties of the Algorithm 1 are formulated in the following proposition, which is the particular case of the Proposition 2 [DKE12].

Proposition 2

If the saddle point $(\hat{\mathbf{y}}, \hat{\omega}, \hat{\lambda})$ of $L_a(\mathbf{y}, \omega, \lambda)$ (2.5) exist, then $\mathbf{y}_t \rightarrow \hat{\mathbf{y}}, \omega_t \rightarrow \hat{\omega}, \lambda_t \rightarrow \hat{\lambda}$. On the other hand, if no such a saddle point exists, then at least one of the sequences $\{\mathbf{y}_t\}$ or $\{\lambda_t\}$ must be unbounded.

Proof. See the proof of Proposition 2 in [DKE12]. ■

2.1.4 Discussion

An example of image denoising with a BM3D-frame using Algorithm 1 is presented in Figure 2.1(c). In this example noise variance is equal to 50, block size of the BM3D-frame is 8, group size is 16 and the step size between the reference blocks is set to 1. Block matching is performed on the estimate obtained from the hard thresholding step of the BM3D filter [DFKE07] and parameters τ and γ were selected such to ensure best quality of the reconstructed image in terms of PSNR.

We can see that though in most parts of the image noise is fairly suppressed, there are number of areas where residual noise is clearly noticeable. This observation can be explained if we look closer at the formula (2.9). Multiplying its right side by $(\Phi^T\Phi)^{-1}(\Phi^T\Phi)$ and taking into account that $\Phi^T\Phi$ is a diagonal matrix, after simplifications we obtain:

$$\begin{aligned} \mathbf{y}_t &= (\Phi^T\Phi)^{-1}(\Phi^T\Phi)(\mathbf{I} + \gamma\Phi^T\Phi)^{-1}(\mathbf{z} + \gamma\Phi^T(\omega_{t-1} + \lambda_{t-1})) \\ &= \left((\Phi^T\Phi)^{-1} + \gamma\mathbf{I} \right)^{-1} \left((\Phi^T\Phi)^{-1}\mathbf{z} + \gamma(\Phi^T\Phi)^{-1}\Phi^T(\omega_{t-1} + \lambda_{t-1}) \right), \end{aligned}$$

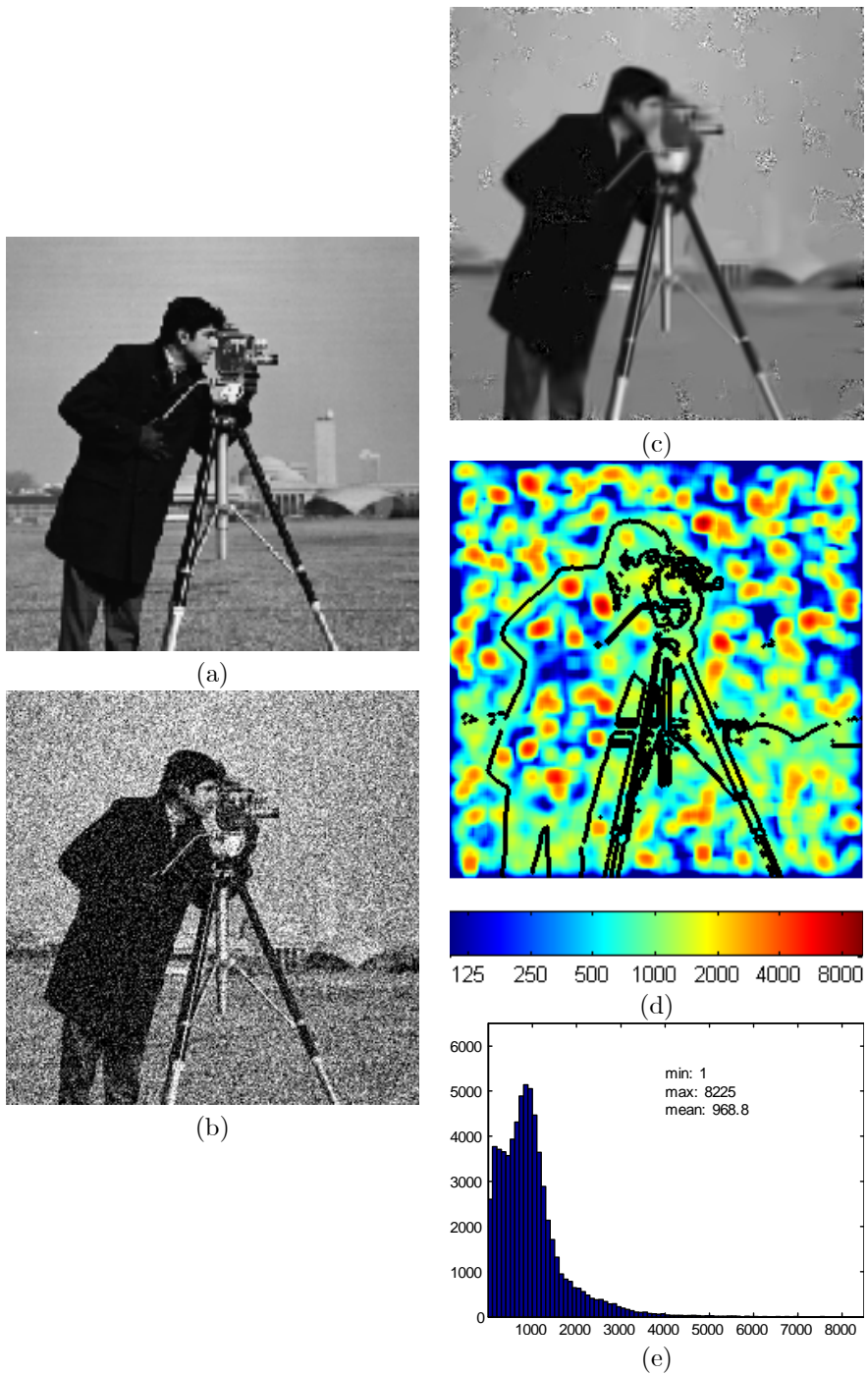


Figure 2.1: Example of denoising with Algorithm 1. (a) True image, (b) noisy image $\sigma = 50$, (c) denoised image, (d) visualization of the diagonal elements of matrix $\Phi^T \Phi$ (edges of the original image are superimposed for the convenience of comparison), (e) histogram of the diagonal elements of matrix $\Phi^T \Phi$.

or substituting $(\Phi^T \Phi)^{-1} \Phi^T = \Psi$:

$$\mathbf{y}_t = \left((\Phi^T \Phi)^{-1} + \gamma \mathbf{I} \right)^{-1} \left((\Phi^T \Phi)^{-1} \mathbf{z} + \gamma \Psi (\boldsymbol{\omega}_{t-1} + \boldsymbol{\lambda}_{t-1}) \right). \quad (2.11)$$

Formula (2.11) states, that each pixel of the image estimate \mathbf{y}_t is obtained as a weighted average of the corresponding pixels of the observation \mathbf{z} and the estimate obtained from the spectrum $\Psi(\boldsymbol{\omega}_{t-1} + \boldsymbol{\lambda}_{t-1})$. For i -th pixel the weight given to the observation component is the reciprocal of the i -th diagonal element of matrix $\Phi^T \Phi$, while weight of the spectral estimate is γ . Since we assume the frame to be non-tight $\Phi^T \Phi \neq a\mathbf{I}$ and hence the averaging proportion (and hence regularization level), varies from pixel to pixel. Pixels reconstructed with a weaker regularization demonstrate higher variance, while pixels with a stronger regularization demonstrate higher bias. This relation becomes apparent if we visualize the diagonal of $\Phi^T \Phi$ as an image (Fig. 2.1(d)), where intensity of each pixel is corresponding to the value of the respective diagonal element of $\Phi^T \Phi$. Comparing images (c) and (d) in Fig. 2.1 we can see that areas with high residual noise correspond to the areas where elements of $\Phi^T \Phi$ have small values.

The difference in the regularization level can be ignored if it is small. Unfortunately, for BM3D-frames this is not the typical case. As it was mentioned in the previous chapter, the matrix $\Phi^T \Phi$ is entirely defined by the block-matching; its i -th diagonal element shows total number of times i -th pixel appears in different groups. Experiments demonstrate that variation of this parameter can be very large, up to several thousand times. Figure 2.1(e) shows the histogram calculated from the diagonal of matrix $\Phi^T \Phi$ used to obtain reconstruction in Fig. 2.1(c). The ratio between the maximum and the minimum diagonal elements, for this particular case, is of the order of 10^5 .

It is worth mentioning that variation level of the elements in matrix $\Phi^T \Phi$ relatively weakly depends on the quality of the image on which block matching is performed. This is illustrated in Fig.2.2, where we show histograms calculated from the diagonal elements of the matrix $\Phi^T \Phi$ corresponding to the two cases, when block matching is done on the noisy and true images. We can see that even in the case when matching is done on the ground truth image the variation is still significant.

Uneven regularization creates a problem for selecting optimal values for parameters γ and τ . To avoid appearance of the residual noise in the regions with weak regularization, one needs to select much higher values of the parameters than otherwise would be optimal for the regions with strong regularization. This results in oversmoothing in the areas with strong regularization.

2.1.5 Criticism

We would like to mention that the problem of uneven regularization exhibits only when the frame is not tight. Indeed, for the tight frame bounds a and b in (1.19) are equal, and hence $\Phi^T \Phi = a\mathbf{I}$. Substituting this equality in (2.11), we obtain

$$\mathbf{y}_t = (\mathbf{I} + \gamma a)^{-1} (\mathbf{z} + \gamma a \Psi (\boldsymbol{\omega}_{t-1} + \boldsymbol{\lambda}_{t-1})),$$

which means that the proportion in which the observation and the current estimate are averaged is the same for all pixels. Based on this observation, one may

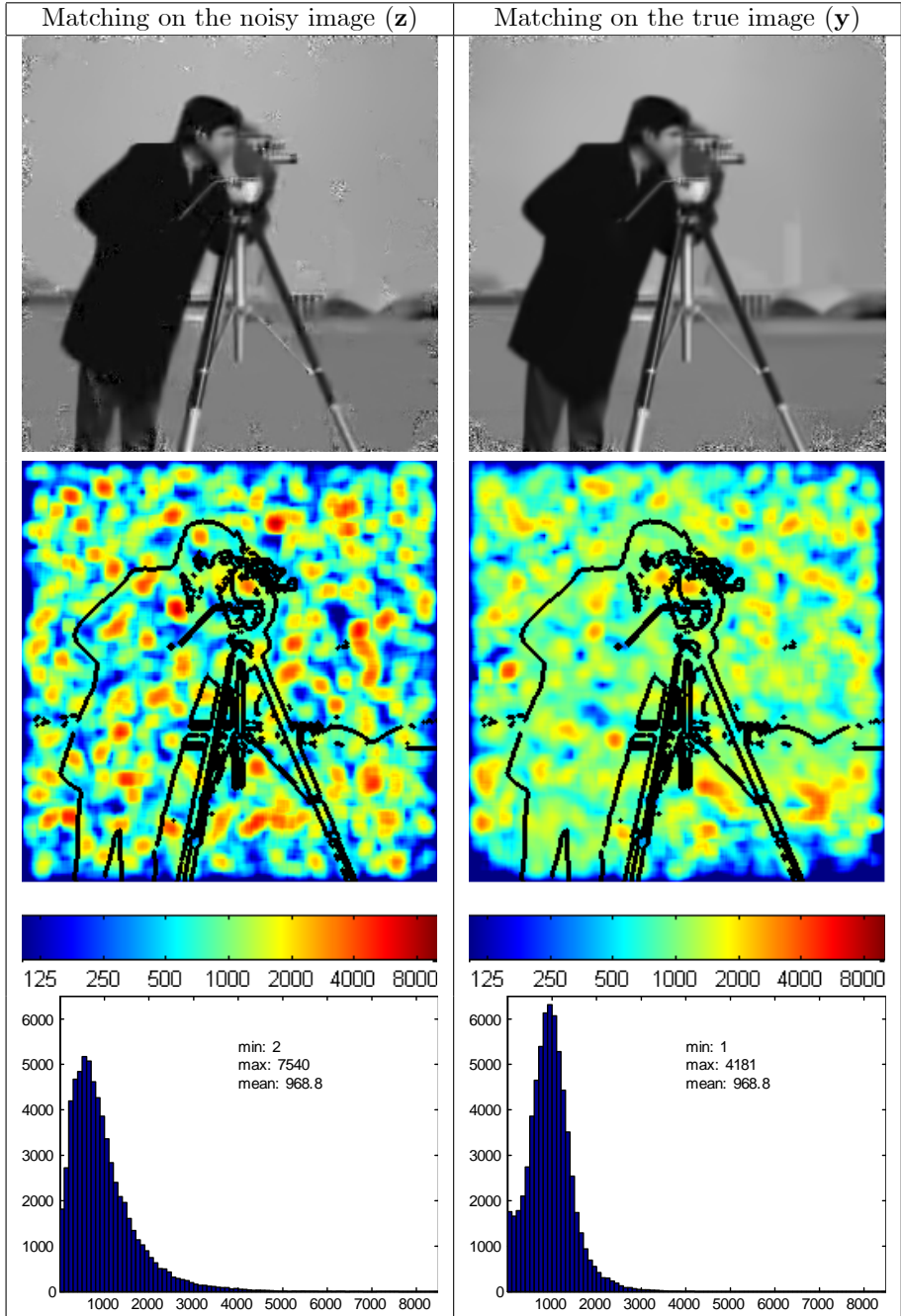


Figure 2.2: Influence of the quality of the image used for block matching on the variation of the diagonal elements of matrix $\Phi^T \Phi$.

Left column corresponds to the matching on the noisy image \mathbf{z} , right column - to the matching on the ground truth \mathbf{y} . From top to bottom: denoised image, visualization of the diagonal elements of matrix $\Phi^T \Phi$ and the histogram of the diagonal elements.

naively suggest using the tight analog of the frame (obtained by normalization) to overcome the problem of uneven regularization. Below we demonstrate, that unfortunately, normalization results in catastrophic degradation of sparsification properties of the frame.

Proposition 3 *Given an arbitrary frame $\{\phi_m\}_{m=1}^M$, its tight analog $\{\tilde{\phi}_m\}_{m=1}^M$ can be obtained as*

$$\tilde{\phi}_m = (\Phi^T \Phi)^{-\frac{1}{2}} \phi_m \quad (2.12)$$

with the corresponding analysis

$$\tilde{\Phi} = \Phi (\Phi^T \Phi)^{-\frac{1}{2}},$$

and synthesis

$$\tilde{\Psi} = (\Phi^T \Phi)^{-\frac{1}{2}} \Phi^T$$

operators. The bounds of the frame $\{\tilde{\phi}_m\}_{m=1}^M$ are equal to 1.

Proof. If $\Phi^T \Phi$ is a diagonal matrix we have

$$\begin{aligned} \|\tilde{\Phi} \mathbf{y}\|_2^2 &= \langle \tilde{\Phi} \mathbf{y}, \tilde{\Phi} \mathbf{y} \rangle = \mathbf{y}^T \tilde{\Phi}^T \tilde{\Phi} \mathbf{y} = \\ &= \mathbf{y}^T \left((\Phi^T \Phi)^{-\frac{1}{2}} \right)^T \Phi^T \Phi (\Phi^T \Phi)^{-\frac{1}{2}} \mathbf{y} = \mathbf{y}^T \mathbf{I} \mathbf{y} = \|\mathbf{y}\|_2^2. \end{aligned}$$

The proof for the general case when $\Phi^T \Phi$ is not a diagonal matrix, can be found in [Chr03] (see Theorem 5.3.4). ■

To compare sparsifying properties of the original non-tight frame and its normalized tight counterpart obtained from (2.12), we perform an experiment where noise-free images are reconstructed from K most significant coefficients of their representations in frames $\{\phi_m\}_{m=1}^M$ and $\{\tilde{\phi}_m\}_{m=1}^M$. In Figure 2.3 the RMSE of the reconstructed images are presented for different values of K . We can see that compared to the original frame its normalized tight counterpart demonstrates much worse signal sparsification properties, requiring about 100 – 1000 times more coefficients in order to reconstruct the images with the same RMSE. The explanation of such a behavior can be found if we look at the reconstruction of Cameraman image presented in Figure 2.3. While image obtained using the non-tight frame can be considered as a reasonable estimate of the original image, the reconstruction obtained with the tight frame suffers from strong artefacts. Particularly, the smooth area of the sky is corrupted by oscillating patterns. This allows us to conclude, that at least at the spatial locations corresponding to the sky, the tight frame lacks elements suitable for accurately representing constant signal. At the same time, such elements (corresponding to the DC element of the transform) are present in the original non-tight frame. Thus, the normalization process changes frame elements making them unsuitable for representing natural images.

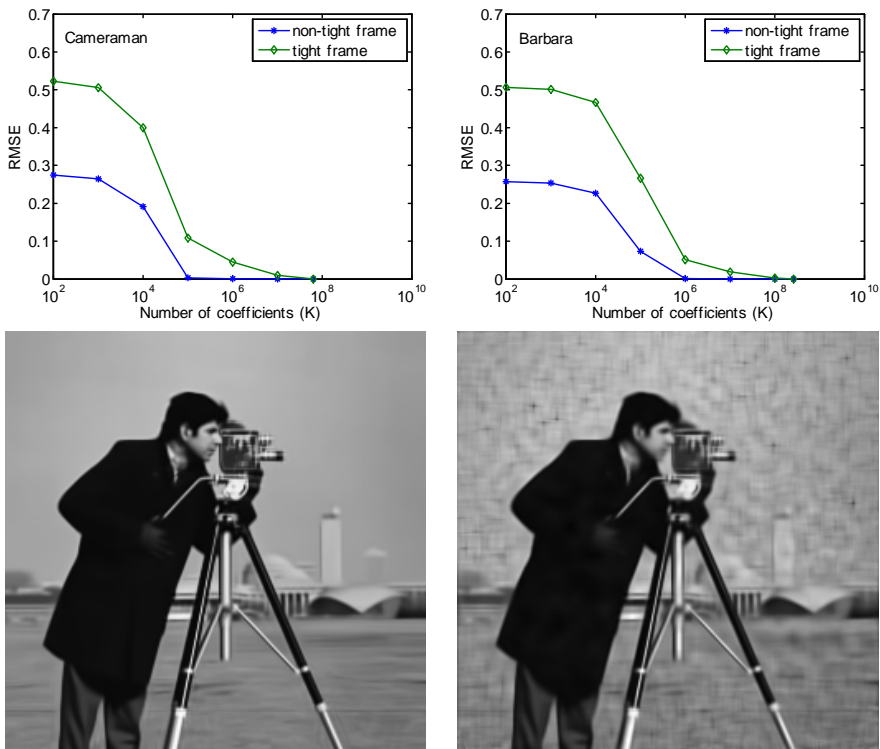


Figure 2.3: Top row: RMSE of the images reconstructed from the K largest coefficients using original non-tight and normalized tight frames. Left Cameraman (256×256), right Barbara (512×512). Redundancy factor ~ 1024 . Bottom row: Cameraman image reconstructed from 10^6 largest coefficients using original non-tight (left) and normalized tight (right) frames. PNSR values are 30.57dB and 13.49dB respectively.

Signal norms in the image and spectrum domains are linked through the matrix $\Phi^T \Phi$. Its negative impact on the reconstruction is inevitable² as long as variational criterion combines norms both for the image and its spectrum. Hence, uneven regularization roots in the very formulation of the BP denoising problem. We conclude, that for the non-tight frames originating from patch based image models (such as BM3D), formulating denoising in the BP form (2.2) is inefficient. We

²We could use more standard solution algorithms to demonstrate the problem of uneven regularization. For example, consider the Iterative Soft Thresholding algorithm (1.11). Applying identity $\Phi^T = (\Phi^T \Phi) (\Phi^T \Phi)^{-1} \Phi^T = (\Phi^T \Phi) \Psi^T$ to the IST recursion formula

$$\omega_t = Th_{\tau/c}^{soft} \left(\frac{1}{c} \Phi \left(\mathbf{z} - \Phi^T \omega_{t-1} \right) + \omega_{t-1} \right)$$

we obtain

$$\omega_t = Th_{\tau/c}^{soft} \left(\frac{1}{c} \Phi \left(\mathbf{z} - (\Phi^T \Phi) \Psi^T \omega_{t-1} \right) + \omega_{t-1} \right).$$

We can immediately notice the uneven treatment of the different pixels in the calculation of the difference $\mathbf{z} - (\Phi^T \Phi) \Psi^T \omega_{t-1}$.

should seek an alternative sparsity promoting variational formulations. This will be the topic of the next section.

2.2 Denoising as a Generalized Nash equilibrium problem

2.2.1 Problem formulation

Let us formulate the denoising problem as the following constrained optimization:

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}} \frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 \text{ subject to } \|\mathbf{y} - \Psi\boldsymbol{\omega}^*\|_2^2 \leq \varepsilon_1, \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\|_1 \text{ subject to } \|\boldsymbol{\omega} - \Phi\mathbf{y}^*\|_2^2 \leq \varepsilon_2, \end{cases} \quad (2.13)$$

where $\varepsilon_1, \varepsilon_2 > 0$. Compared to the split formulation (2.3) of the BP denoising, here the strict analysis/synthesis constraints $\mathbf{y} = \Psi\boldsymbol{\omega}$ and $\boldsymbol{\omega} = \Phi\mathbf{y}$ are relaxed, and weaker (inequality) constraints are imposed.

Problem (2.13) can be replaced by the equivalent unconstrained one:

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}} L_1(\mathbf{y}, \boldsymbol{\omega}^*) \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}^*, \boldsymbol{\omega}), \end{cases} \quad (2.14)$$

where

$$L_1(\mathbf{y}, \boldsymbol{\omega}) = \frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 + \frac{\gamma}{2} \|\mathbf{y} - \Psi\boldsymbol{\omega}\|_2^2, \quad (2.15)$$

$$L_2(\mathbf{y}, \boldsymbol{\omega}) = \tau \|\boldsymbol{\omega}\|_1 + \frac{1}{2} \|\boldsymbol{\omega} - \Phi\mathbf{y}\|_2^2, \quad (2.16)$$

and γ, τ are constants selected accordingly to the values of $\varepsilon_1, \varepsilon_2$.

In terms of the game theory, the problem (2.14) can be interpreted as a game of two players identified, respectively, with two variables \mathbf{y} and $\boldsymbol{\omega}$ [LBS08], [FK10]. The interaction between the players is noncooperative, because minimization of $L_1(\mathbf{y}, \boldsymbol{\omega})$ with respect to \mathbf{y} typically results in increase of $L_2(\mathbf{y}, \boldsymbol{\omega})$ and minimization of $L_2(\mathbf{y}, \boldsymbol{\omega})$ with respect to $\boldsymbol{\omega}$ increases $L_1(\mathbf{y}, \boldsymbol{\omega})$. The equilibrium of this game, called *Generalized Nash equilibrium (GNE)*, defines the *fixed point* $(\mathbf{y}^*, \boldsymbol{\omega}^*)$ of the optimization.

The objective functions L_1 and L_2 allow the following interpretation. In L_1 the fidelity term $\frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2$ evaluates the deviation between the observation \mathbf{z} and its estimate \mathbf{y} . This fidelity is penalized by the norm $\|\mathbf{y} - \Psi\boldsymbol{\omega}\|_2^2$, which controls the divergence between the \mathbf{y} and its sparse approximation $\Psi\boldsymbol{\omega}$. The L_2 , in its turn, represents a trade-off between the error of $\boldsymbol{\omega}$ with respect to $\Phi\mathbf{y}$ and the complexity of $\boldsymbol{\omega}$ measured by the l_1 -norm. Hence the GNE provides a balance between the fit of the reconstruction \mathbf{y} to the observation \mathbf{z} and the complexity of the model $\|\boldsymbol{\omega}\|_1$. This can be contrasted with the analysis- and synthesis-based problem formulations where the balance is provided within a single criterion.

2.2.2 Algorithm for solving GNE problem

To solve (2.14) we consider the following iterative procedure:

$$\begin{cases} \mathbf{y}_t = \arg \min_{\mathbf{y}} L_1(\mathbf{y}, \boldsymbol{\omega}_{t-1}) \\ \boldsymbol{\omega}_t = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}_t, \boldsymbol{\omega}) \end{cases}, t = 1, \dots. \quad (2.17)$$

This procedure models the selfish behavior, where each variable minimizes only its own objective function. As it is shown in [DKE12], these iterations converge to the fixed point $(\mathbf{y}^*, \boldsymbol{\omega}^*)$ of (2.14).

Solutions of the minimization subproblems in (2.17) are easy to obtain. Minimization of the quadratic functional L_1 with respect to \mathbf{y} is given by the equation

$$\mathbf{y}_t = (\mathbf{I} + \gamma \mathbf{I})^{-1} (\mathbf{z} + \gamma \boldsymbol{\Psi} \boldsymbol{\omega}_{t-1}), \quad (2.18)$$

which is a simple weighted average between the observation \mathbf{z} and the estimate $\boldsymbol{\Psi} \boldsymbol{\omega}_{t-1}$.

Minimization of L_2 with respect to $\boldsymbol{\omega}$ is obtained by thresholding with the threshold parameter τ :

$$\boldsymbol{\omega}_t = Th_{\tau}^{soft}(\boldsymbol{\Phi} \mathbf{y}_t). \quad (2.19)$$

Summarizing these two steps we obtain Algorithm 2 that solves the problem (2.14).

Algorithm 2 Denoising algorithm based on GNE formulation

- 1: **input:** \mathbf{z}, τ, γ
 - 2: **construct** $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ using \mathbf{z} or any initial estimate \mathbf{y}_{init}
 - 3: **initialization:**
 - 4: *set:* $t = 0, \mathbf{y}_0 = \mathbf{z}, \boldsymbol{\omega}_0 = \boldsymbol{\Phi} \mathbf{z}$
 - 5: **repeat**
 - 6: $t = t + 1$
 - 7: $\mathbf{y}_t = (\mathbf{I} + \gamma \mathbf{I})^{-1} (\mathbf{z} + \gamma \boldsymbol{\Psi} \boldsymbol{\omega}_{t-1})$
 - 8: $\boldsymbol{\omega}_t = Th_{\tau}^{soft}(\boldsymbol{\Phi} \mathbf{y}_t)$
 - 9: **until** convergence
 - 10: **output:** $\hat{\mathbf{y}} = \boldsymbol{\Psi} \boldsymbol{\omega}_t$
-

2.2.3 Convergence

The following proposition describes the convergence properties of the algorithm.

Proposition 4 *For any set of parameters τ, γ the sequence $(\mathbf{y}_t, \boldsymbol{\omega}_t)$ generated by Algorithm 2, converges to a fixed point $(\mathbf{y}^*, \boldsymbol{\omega}^*)$ defined by the equations (2.14), if a fixed point exists.*

Proof. See the proof of the more general Proposition 3 in [DKE12]. ■

Note, that it is not required that the fixed point is unique. Depending on a starting point $(\mathbf{y}_0, \boldsymbol{\omega}_0)$ the limit point of the algorithm can be different but should satisfy the fixed-point equations.

2.2.4 Discussion

Each solution of the GNE denoising problem (2.14) is a pair of vectors $(\mathbf{y}^*, \boldsymbol{\omega}^*)$. Both \mathbf{y}^* and $\boldsymbol{\Psi}\boldsymbol{\omega}^*$ can be considered as estimates of the true image $\bar{\mathbf{y}}$, but unlike BP denoising case, these two estimates are going to be different. Indeed, in the GNE formulation we do not enforce equality $\boldsymbol{\Phi}\mathbf{y}^* = \boldsymbol{\omega}^*$, instead it is only required that $\|\mathbf{y}^* - \boldsymbol{\Psi}\boldsymbol{\omega}^*\|_2^2 \leq \varepsilon_1$ and $\|\boldsymbol{\omega}^* - \boldsymbol{\Phi}\mathbf{y}^*\|_2^2 \leq \varepsilon_2$, $\varepsilon_1, \varepsilon_2 > 0$. How close estimates \mathbf{y}^* and $\boldsymbol{\Psi}\boldsymbol{\omega}^*$ are going to be and which of them will be better, depends on the choice of parameters τ and γ . $\boldsymbol{\Psi}\boldsymbol{\omega}^*$ can be considered preferable to \mathbf{y}^* , since \mathbf{y}^* contains noisy component through its $\frac{\gamma}{1+\gamma}\mathbf{z}$ summand. Hence it is rational to select parameters τ and γ such to maximize the quality of $\boldsymbol{\Psi}\boldsymbol{\omega}^*$ while ignoring \mathbf{y}^* . In Fig. 2.4 (e) and (f) we show estimates $\boldsymbol{\Psi}\boldsymbol{\omega}^*$ and \mathbf{y}^* resulting from the same denoising experiment. The presence of noise is clearly visible on \mathbf{y}^* . For the comparison we show estimates $\hat{\mathbf{y}}$ and $\boldsymbol{\Psi}\hat{\boldsymbol{\omega}}$ (Fig. 2.4 (c) and (d)), corresponding to denoising with Algorithm 1, which are practically identical.

In the Section 2.1.5 we criticized the BP denoising formulation for enforcing uneven regularization in the case of the non-tight frames. The NE based formulation (2.14) is free of this drawback. Indeed, unlike BP case, in the update step (2.18) the weighted averaging between the observation \mathbf{z} and $\boldsymbol{\Psi}\boldsymbol{\omega}_t$ is performed in the same proportion for all pixels. As a result, we are able to obtain better reconstruction free of the areas with strong residual noise. This can be verified from Fig. 2.4 (e) where we show the result of denoising *Camerman* image with Algorithm 2. For the reference we also show the result obtained with the BP denoising Algorithm 1 (Fig. 2.4 (c)).

So far as a sparsity promoting penalty we have considered only l_1 norm. We can replace l_1 with l_0 (pseudo-) norm in the GNE problem (2.14) and formally derive algorithm for it. This algorithm will repeat Algorithm 2 with the only difference that the soft thresholding will be replaced by the hard thresholding (1.13). Unfortunately, convergence analysis done for (2.14) cannot be extended to the l_0 -norm case. Convergence proof of Algorithm 2 relies on non-expansivity of the soft thresholding operator, while hard thresholding operator is not a non-expansive operator. Nevertheless, in practice, the algorithm for l_0 converges for reasonably selected parameters τ and γ .

2.3 Experiments

We compare proposed Algorithm 2 against the standard BM3D algorithm [DFKE07]. Both the soft and hard thresholding versions of the Algorithm 2 are evaluated. Additionally we consider the hard thresholding version of the algorithm with group weights chosen adaptively. In Table 2.1 we present the PSNR values of the denoised images obtained with each of the method. To simplify comparison of these data, in Figure 2.5 we plot the PSNR scores relative to the results of the BM3D algorithm [DFKE07] which are selected as the baseline.

The presented data demonstrate that the hard thresholding version of the iterative algorithm clearly outperforms the version which uses soft thresholding. Overall, results of the hard thresholding version of Algorithm 2 are close to the results of the BM3D algorithm, both according to the numerical criteria and visual

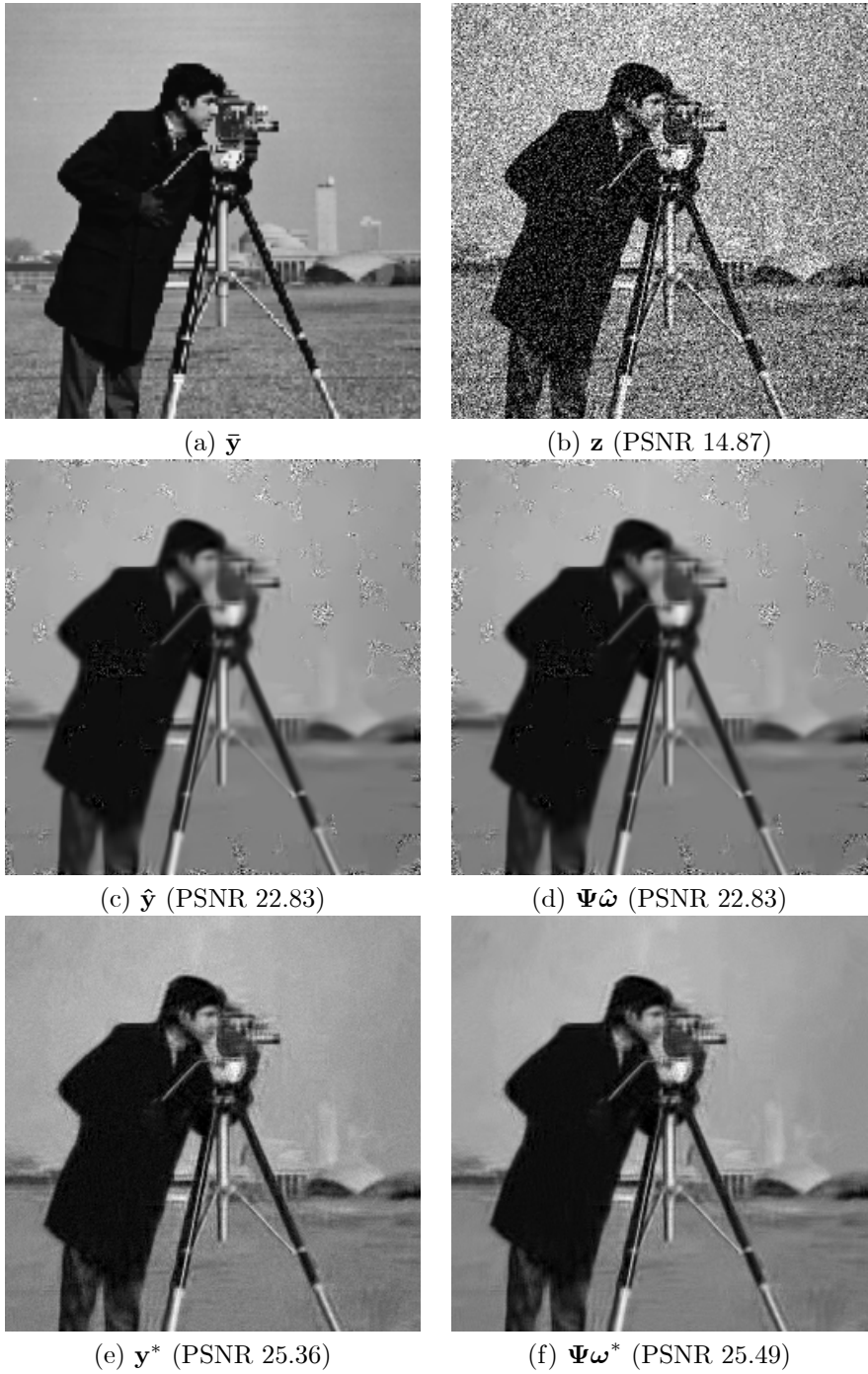


Figure 2.4: Comparison of the reconstructions corresponding to the GNE and BPD problem formulations. (a) true image, (b) noisy image $\sigma = 50$, (c)-(d) reconstructions corresponding to the BPD formulation obtained with Algorithm 1, (e)-(f) reconstructions corresponding to the GNE formulation obtained with Algorithm 2.

inspection. BM3D slightly overperforms in the low noise cases while the proposed algorithm performs slightly better in the high noise.

Judging solely from the PSNR values, one may argue the need of using adaptive group weights, since in many cases they are either not increasing or slightly lowering the PSNR. However, adaptive weights help to improve the visual appearance of the denoised images by suppressing (to some extent) the high frequency artefacts around the edges of objects. This improvement comes at the cost of slight oversmoothing the textured areas. While the oversmoothing decreases the PSNR, visually it is practically unnoticeable. Hence, the benefits that adaptive weights potentially bring pay off the risk of getting slightly oversmoothed images. This observation is illustrated in Figure 2.8, where we show two pairs of images corresponding to the extreme cases where adaptive group weights provide, respectively, the highest improvement and the highest reduction of PSNR compared to the denoising with unit weights. Note significant suppression of the artefacts around the face of cameraman in the first case and practically indistinguishable differences in the texture on the trousers of Barbara in the second case.

2.4 Conclusions

The results of the last experiments may seem to be discouraging, since the proposed iterative algorithm is unable to outperform the computationally less expensive (non-iterative) BM3D algorithm. But we should remember that BM3D implements more complex modeling which exploits not only sparsity, but also wiener filtering. In fact if we compare our results against the output of the hard-thresholding step of BM3D, we see a clear advantage of the proposed algorithm. Overall, we should admit that for the image denoising task, combined model utilized by BM3D is more efficient than the model generated by the GNE formulation (2.14). Nevertheless, in the next chapter we demonstrate that in the case of more difficult inverse problem - deblurring, situation changes to the opposite.

Method \ noise std. (σ)	5	10	25	50	75	100
<i>Cameraman (256 × 256)</i>						
BM3D (hard thresh. only)	38.20	33.94	29.14	25.55	23.45	21.91
BM3D (hard thresh. + wiener)	38.29	34.18	29.45	26.12	24.33	23.07
GNE (soft thresh., unit weights)	37.06	32.88	28.37	25.49	23.89	22.77
GNE (hard thresh., unit weights)	37.97	33.80	29.04	26.17	24.44	23.26
GNE (hard thresh., adaptive weights)	38.14	33.97	29.15	26.20	24.47	23.23
<i>Lena (512 × 512)</i>						
BM3D (hard thresh. only)	38.63	35.65	31.37	27.92	25.85	24.30
BM3D (hard thresh. + wiener)	38.72	35.93	32.08	29.05	27.26	25.95
GNE (soft thresh., unit weights)	38.40	35.27	31.42	28.58	26.95	25.79
GNE (hard thresh., unit weights)	38.75	35.85	32.05	29.10	27.36	26.11
GNE (hard thresh., adaptive weights)	38.74	35.83	32.02	29.04	27.30	26.05
<i>House (256 × 256)</i>						
BM3D (hard thresh. only)	39.58	36.35	32.27	28.42	25.83	23.97
BM3D (hard thresh. + wiener)	39.83	36.71	32.86	29.69	27.51	25.87
GNE (soft thresh., unit weights)	39.13	35.90	32.13	29.18	27.25	25.79
GNE (hard thresh., unit weights)	39.76	36.60	32.82	29.86	27.74	26.19
GNE (hard thresh., adaptive weights)	39.80	36.62	32.88	29.88	27.78	26.18
<i>Barbara (512 × 512)</i>						
BM3D (hard thresh. only)	38.12	34.62	29.92	25.95	23.71	22.24
BM3D (hard thresh. + wiener)	38.31	34.98	30.72	27.23	25.12	23.62
GNE (soft thresh., unit weights)	37.74	34.01	29.47	26.09	24.16	22.87
GNE (hard thresh., unit weights)	38.38	34.87	30.55	27.01	24.90	23.35
GNE (hard thresh., adaptive weights)	38.40	34.88	30.51	26.91	24.81	23.26
<i>Boats (512 × 512)</i>						
BM3D (hard thresh. only)	37.08	33.70	29.43	26.00	24.09	22.75
BM3D (hard thresh. + wiener)	37.28	33.92	29.91	26.78	25.12	23.97
GNE (soft thresh., unit weights)	36.84	33.25	29.20	26.41	24.85	23.77
GNE (hard thresh., unit weights)	37.33	33.86	29.82	26.86	25.19	24.04
GNE (hard thresh., adaptive weights)	37.32	33.82	29.78	26.80	25.16	23.97
<i>Hill (512 × 512)</i>						
BM3D (hard thresh. only)	36.92	33.40	29.46	26.39	24.62	23.36
BM3D (hard thresh. + wiener)	37.13	33.62	29.85	27.19	25.68	24.58
GNE (soft thresh., unit weights)	36.65	33.13	29.46	26.99	25.53	24.49
GNE (hard thresh., unit weights)	37.10	33.55	29.78	27.25	25.74	24.62
GNE (hard thresh., adaptive weights)	37.08	33.49	29.74	27.20	25.70	24.57
<i>Man (512 × 512)</i>						
BM3D (hard thresh. only)	37.64	33.70	29.20	26.13	24.37	23.10
BM3D (hard thresh. + wiener)	37.82	33.98	29.62	26.81	25.32	24.22
GNE (soft thresh., unit weights)	36.94	33.10	29.04	26.49	25.08	24.09
GNE (hard thresh., unit weights)	37.67	33.84	29.51	26.81	25.33	24.27
GNE (hard thresh., adaptive weights)	37.70	33.83	29.46	26.77	25.31	24.25
<i>Peppers (256 × 256)</i>						
BM3D (hard thresh. only)	38.09	34.52	29.68	25.90	23.67	21.97
BM3D (hard thresh. + wiener)	38.12	34.68	30.16	26.68	24.73	23.39
GNE (soft thresh., unit weights)	37.24	33.50	29.16	26.16	24.38	23.12
GNE (hard thresh., unit weights)	37.88	34.33	29.92	26.71	24.78	23.45
GNE (hard thresh., adaptive weights)	37.95	34.39	29.94	26.69	24.79	23.42

Table 2.1: PSNR performance of the considered denoising algorithms utilizing BM3D-modeling.

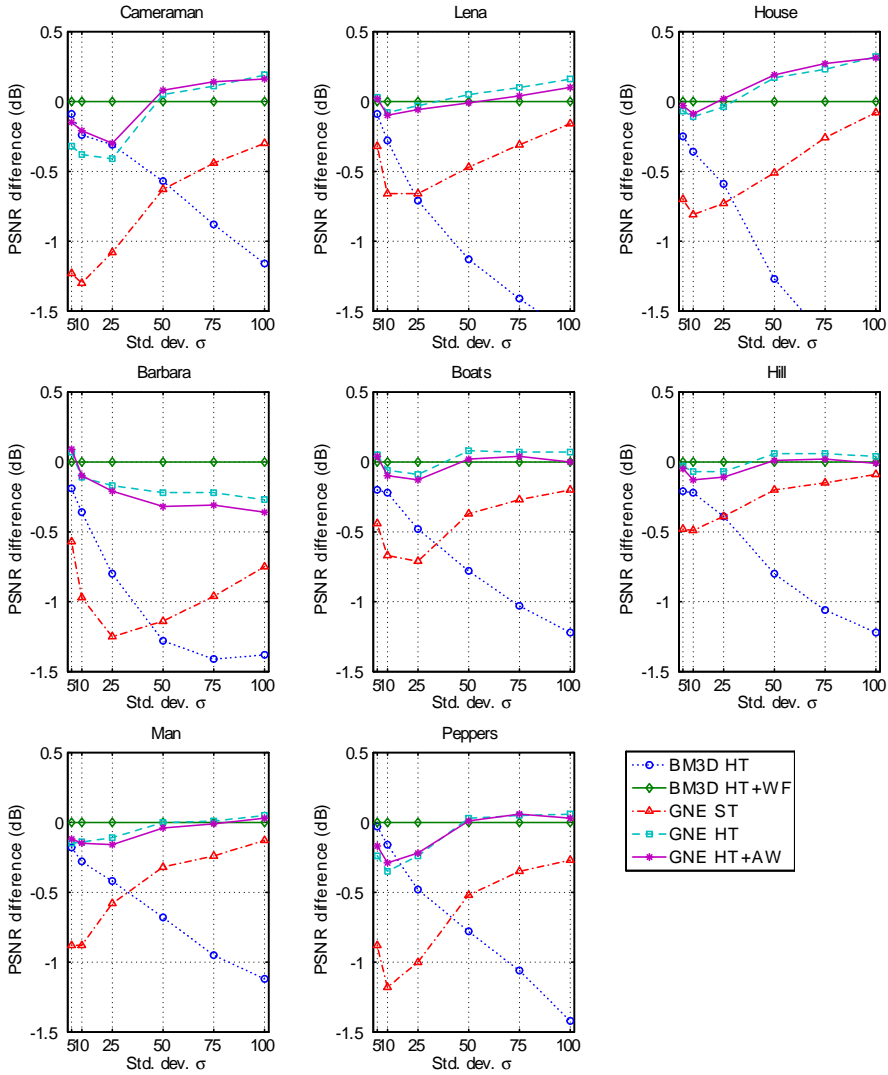


Figure 2.5: Relative PSNR performance of the considered denoising algorithms utilizing BM3D-modeling. The results of the BM3D denoising [DFKE07] algorithm are considered as the baselines.

BM3D HT is the BM3D denoising [DFKE07] algorithm including only hard thresholding step, BM3D HT+WF is the standard two-step BM3D denoising [DFKE07] algorithm including hard thresholding and wiener filtering steps. GNE ST -states for the Algorithm 2 with soft thresholding, GNE HT - Algorithm 2 with hard thresholding and GNE HT+AW states for the same algorithm with hard thresholding and group weights selected adaptively.





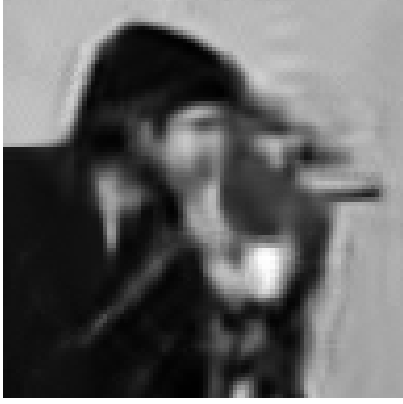
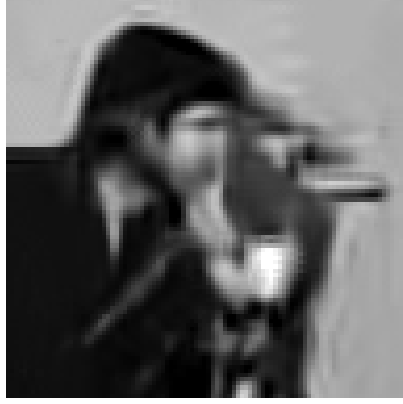
	BM3D HT+WF	GNE HT+AW
<i>Peppers</i> , $\sigma = 10$		
	PSNR: 34.68, SSIM: 0.9282	PSNR: 34.39, SSIM: 0.9266
<i>Cameraman</i> , $\sigma = 25$		
	PSNR: 29.45, SSIM: 0.8544	PSNR: 29.15, SSIM: 0.850
<i>Cameraman</i> , $\sigma = 100$		
	PSNR: 23.07, SSIM: 0.6924	PSNR: 23.23, SSIM: 0.7080

Figure 2.6: Zoomed fragments of the denoised images.

Left column (BM3D HT+WF): standard two-step BM3D denoising [DFKE07] algorithm including hard thresholding and wiener filtering steps. Right column (GNE HT+AW): Algorithm 2 with hard thresholding and group weights selected adaptively.



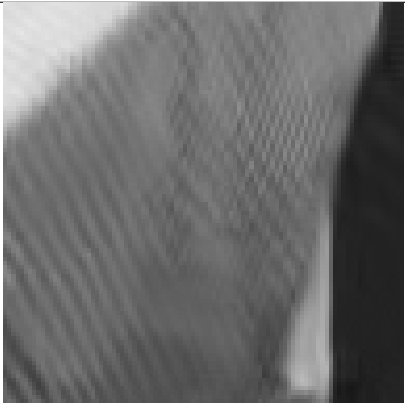
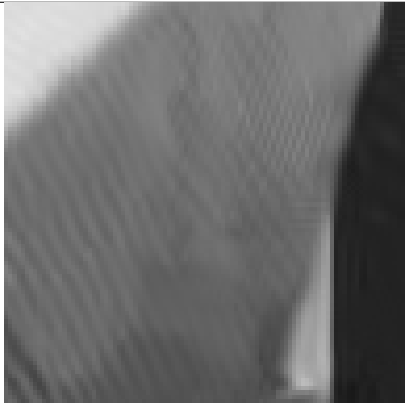
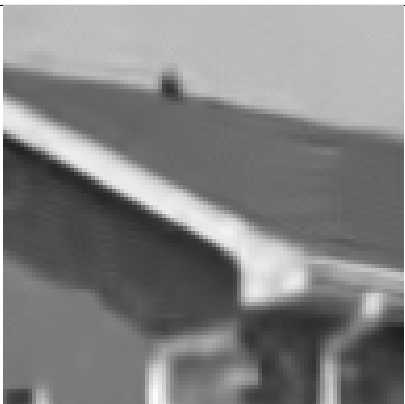

	BM3D HT+WF	GNE HT+AW
<i>Boat, $\sigma = 25$</i>		
	PSNR: 29.91, SSIM: 0.9058	PSNR: 29.78, SSIM: 0.9046
<i>Barbara, $\sigma = 50$</i>		
	PSNR: 27.23, SSIM: 0.8721	PSNR: 26.91, SSIM: 0.8633
<i>House, $\sigma = 50$</i>		
	PSNR: 29.69, SSIM: 0.8122	PSNR: 29.88, SSIM: 0.8194

Figure 2.7: Zoomed fragments of the denoised images (continued).

Left column (BM3D HT+WF): standard two-step BM3D denoising [DFKE07] algorithm including hard thresholding and wiener filtering steps. Right column (GNE HT+AW): Algorithm 2 with hard thresholding and group weights selected adaptively.





		GNE HT	
		unit weights	adaptive weights
<i>Cameraman</i> , $\sigma = 10$			
	PSNR: 33.80, SSIM: 0.9272	PSNR: 33.97, SSIM: 0.9313	
<i>Barbara</i> , $\sigma = 50$			
	PSNR: 27.01, SSIM: 0.8652	PSNR: 26.91, SSIM: 0.8633	

Figure 2.8: The effect of using adaptive group weights. Top row shows the benefit - significant suppression of the artefacts around the edges of objects. Bottom row shows the hardly noticeable drawback - slight oversmoothning of textures.

Chapter 3

Deblurring

Blur is a general term used to denote deterministic linear degradations which results in band-limiting the imaging process. The reasons causing blur can be various: relative motion of the object and the imaging device (motion blur), imperfections of the optical system, wrong focussing (out-of-focus blur), atmospheric turbulence, etc. Typically it is desirable to revert the action of blur. The process aiming to reconstruct the original image from its blurred observation is called *deblurring*.

The mathematical formulation of the deblurring problem is given differently depending whether the model of the blur is known or not. In the more complex case, the blurring model is unknown, and the problem is formulated as joint estimation of the original image and the blur operator. This is so-called *blind deblurring* problem [CE07]. Nevertheless, in many practical cases the blur operator is known or can be estimated from some other independent measurements. The deblurring is then reduces to the standard linear inverse problem. In what follows, we consider only problems where the blur operator is known. Additionally we will assume that the measured data is corrupted by noise with known distribution.

The deblurring problems are typically ill-posed, since the inverse of the band-limiting operation often does not exist. Even when the inverse exists it still may be ill-conditioned: small perturbations introduced by the presence of noise can be amplified by the inverse operator introducing significant distortions in the restored image. To overcome the ill-posedness, the reconstruction problem needs to be regularized. Historically, deblurring problems were treated using classical regularization techniques, such as Tikhonov regularization [TA77] or empirical Wiener filtering, [BB98]. Later, with advances in image modeling more sophisticated techniques were proposed. The modern deblurring algorithms can be roughly divided in two classes.

The algorithms from the first class are treating deblurring as a color noise removal problem. The general scheme of this approach is simple. In the first step Tikhonov regularized inverse is applied to obtain the initial estimate. The strength of the regularization is selected such, to be barely enough to perform the inverse. The resulting estimate contains significant amount of residual noise, which is then removed by the denoising filter. The advantage of the described ap-

proach compared to the standard Tikhonov regularization is that it allows to use much less regularization, which otherwise would be needed to suppress the noise using only Tikhonov regularization. Less regularization leads to more accurate inversion of the blur operator and better preservation of image details. This approach is particularly efficient in the case when only little regularization is needed to invert the blur operator, since in that case problem essentially reduces to denoising. Arguably, the most known algorithm of this class is the Fourier-Wavelet Regularized Deconvolution (ForWaRD) [NCB04]. It implements more advanced, two-step deblurring procedure. The first step implements Tikhonov inverse followed by shrinkage filtering in the discrete Fourier transform domain. The aim of this step is suppressing the colored noise. The output of the first step is then used as the pilot estimate in the second step, where deconvolution based on empirical Wiener filtering is followed by shrinkage in the wavelet domain. The last shrinkage is called to suppress the distortions (such as ringing) arising due to the imperfectness of the blur inversion process. The two-step reconstruction scheme implemented in ForWaRD was found to be quite efficient and, at the same time, very attracting by its simplicity, since any denoising filter which is adapted to work with colored noise can be plugged into the scheme producing a new deblurring algorithm. Several successful variations of the ForWaRD were proposed based on different denoising filters, such as Anisotropic LPA-ICI [KEA05], Shape-Adaptive DCT [FDKE06] and BM3D [DFKE08a].

The second class of deblurring algorithms is based on formulating deblurring as a variational problem, where the functional to be optimized consist of two terms: fidelity and penalty. The fidelity term reflects the likelihood that the considered solution can indeed be the generator of the observed data. The penalty, in its turn, expresses the probability that the considered solution is generated by the a-priori image model assumed by the algorithm. The functional is formed as a sum of the fidelity term plus penalty term multiplied by a scalar which is called *regularization parameter*. The regularization parameter controls the balance between the fit to the observed data and fit to the a-priori model. In Chapter 1 we already discussed several variational formulations arising from sparsity priors. From the vast number of other proposed priors it worth to mention the Total Variation (TV) prior. It has been first introduced in [ROF92] and since then became one of the most widely used priors. The TV prior limits the total variation of the signal by penalizing the l_1 -norm of its gradient. The TV penalty tends to preserve sharp discontinuities, such as edges, favouring piecewise constant signals. Overall it provides a reasonable model for cartoon-like images, while pictures containing regions with smooth transition and textures do not follow the minimum TV model.

In our paper [DKE12] we proposed a novel deblurring algorithm, where the problem is formulated as a search for the generalized Nash equilibrium, which would balance the fit to the observation model and complexity of the estimate in the BM3D-frame domain. This approach has been found extremely successful, leading, to the best of our knowledge, to the state-of-the-art results in the field. Below we briefly review the proposed approach, algorithm and demonstrate experimental results obtained with it. The extension to the Poisson noise case is also presented.

3.1 Proposed approach

Let $\mathbf{z}, \bar{\mathbf{y}} \in \mathbb{R}^N$ be vectors representing, respectively, the observed and true images, and \mathbf{A} be an $N \times N$ matrix representing the blur operator. We assume that the observation is noisy, i.e. \mathbf{z} is a sample of the random vector Z with known distribution, such that $E\{Z\} = \mathbf{A}\bar{\mathbf{y}}$.

Given an estimate \mathbf{y} we evaluate its misfit of to the observation model with the negative log-likelihood function $L_{\mathbf{z}}(\mathbf{y}) = -\log p(\mathbf{z}|\mathbf{y})$. Using the notations introduced in the previous chapters, we formulate deblurring as the GNE¹ problem

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}} L_1(\mathbf{y}, \boldsymbol{\omega}^*) \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}^*, \boldsymbol{\omega}) \end{cases}, \quad (3.1)$$

where

$$L_1(\mathbf{y}, \boldsymbol{\omega}) = L_{\mathbf{z}}(\mathbf{y}) + \frac{\gamma}{2} \|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\omega}\|_2^2, \quad (3.2)$$

$$L_2(\mathbf{y}, \boldsymbol{\omega}) = \tau \|\boldsymbol{\omega}\|_p + \frac{1}{2} \|\boldsymbol{\omega} - \boldsymbol{\Phi}\mathbf{y}\|_2^2, \quad (3.3)$$

$\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are respectively the analysis and synthesis BM3D-frames, and l_p -norm is either l_1 or l_0 . The algorithm solving (3.1) can be obtained by implementing alternating minimization procedure:

$$\begin{cases} \mathbf{y}_t = \arg \min_{\mathbf{y}} L_1(\mathbf{y}, \boldsymbol{\omega}_{t-1}) \\ \boldsymbol{\omega}_t = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}_t, \boldsymbol{\omega}) \end{cases}, \quad t = 0, 1, \dots \quad (3.4)$$

It can be shown (see proof of Proposition 3 in [DKE12]), that under certain conditions, iterations in (3.4) converge to a fixed point $(\mathbf{y}^*, \boldsymbol{\omega}^*)$ of (3.1).

3.1.1 Gaussian data: IDD-BM3D algorithm

In the case when noise is additive zero-mean white Gaussian the observation model is expressed as

$$\mathbf{z} = \mathbf{A}\bar{\mathbf{y}} + \sigma\boldsymbol{\eta}, \quad (3.5)$$

where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}_{N \times 1}, \mathbf{I}_{N \times N})$ and σ is the standard deviation of the noise. The negative log-likelihood function corresponding to the observation \mathbf{z} is

$$L_{\mathbf{z}}(\mathbf{y}) = \frac{1}{2} \|\mathbf{z} - \mathbf{A}\mathbf{y}\|_2^2$$

and therefore function $L_1(\mathbf{y}, \boldsymbol{\omega})$ in (3.2) is quadratic with respect to \mathbf{y} and $\boldsymbol{\omega}$. As a result, both minimization subproblems in (3.4) have close-form solutions. We omit the derivations (they can be found in [DKE12]), and present the final Algorithm 3 called Iterative Decoupled Deblurring with BM3D-frames (IDD-BM3D)².

¹Strictly speaking 3.1 is not a GNE problem. The formulation of the GNE requires that each variable should have its feasible set which depends on the other variables. In 3.1 the constraints defining feasible sets $\|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\omega}\| \leq \varepsilon_1$ and $\|\boldsymbol{\omega} - \boldsymbol{\Phi}\mathbf{y}\| \leq \varepsilon_2$ are incorporated into the cost functions

Algorithm 3 IDD-BM3D - Iterative Decoupled Deblurring with BM3D-frames

```

1: input:  $\mathbf{z}, \mathbf{A}, \hat{\mathbf{y}}_{\text{init}}, \tau, \gamma$ 
2: initialization:
3: using  $\hat{\mathbf{y}}_{\text{init}}$  construct operators  $\Phi$  and  $\Psi$ 
4: set:  $t = 0, \mathbf{y}_0 = \hat{\mathbf{y}}_{\text{init}}, \boldsymbol{\omega}_0 = \Phi \hat{\mathbf{y}}_{\text{init}}$ 
5: repeat
6:    $t = t + 1$ 
7:    $\mathbf{y}_t = (\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I})^{-1} (\mathbf{A}^T \mathbf{z} + \gamma \Psi \boldsymbol{\omega}_{t-1})$ 
8:    $\boldsymbol{\omega}_t = \text{Th}_\tau(\Phi \mathbf{y}_t)$ 
9: until convergence.
10: output:  $\hat{\mathbf{y}} = \Psi \boldsymbol{\omega}_t$ 

```

3.1.2 Poissonian data: PIDD-BM3D algorithm

The Poisson noise is inherent to the imaging systems where measurements are based on photon counting process such as CCD or CMOS camera sensors. The Poisson noise model assumes that the measured values are the realizations of independent random variables distributed according to the Poisson law with expectations equal to the value of the noise-free signal. Since the variance of Poissonian random variable is equal to its expectation, the Poisson noise is signal dependent and heteroscedastic. There are several approaches meant to deal with the data corrupted with heteroscedastic noise.

The first and the simplest approach is to perform image segmentation, considering small image patches of an arbitrary shape where pixel intensities are close to each other, such that the variance of the noise in the patch can be considered practically constant. The noise removal is then performed patchwise using any of the techniques designed for the homoscedastic noise. Applied to the deblurring problem, the segmentation approach was adopted in [FAT⁺05] and [FATK06] where it was used within the ForWaRD-type scheme.

Another popular approach, which was found to be particularly efficient in the context of sole denoising problem ([Foi09], [MF11]), consist in using a *variance stabilizing transform* (VST). The VST is a nonlinear function that being applied elementwise to the heteroscedastic data transforms it into approximately homoscedastic. The most known VST for Poisson noise is the Anscombe transform [Ans48], which converts Poissonian data into asymptotically Gaussian with unit variance. After applying Anscombe VST the resulting stabilized data can be treated with the methods designed for Gaussian noise. To obtain the final solution, the estimate of the stabilized data needs to be remapped to the original range through another transform known as *unbiased inverse* of the VST. The attempts

L_1 and L_2 .

²We would like to mention that while the IDD-BM3D and L0-Abs [Por09] algorithms look quite similar, they are derived from different variational formulations. For tight frames both formulations lead to the same algorithm. However, if one would try to derive an algorithm for non-tight frames using the variational formulation considered in [Por09], the resulting algorithm will be different from the IDD-BM3D and will suffer from the same uneven regularization problem which we discussed in Section 2.1.5.

to use the Anscombe transform for solving image deblurring problems were made in [CBFZ07] and [DFS09]. Unfortunately, none of these papers utilize the VST technique correctly. We discuss this issue in the footnote³ where we also present the correct derivation of the fidelity term based on the VST.

Finally, the reconstruction algorithm can be derived as a MAP solution from the penalized likelihood function. The basic maximum likelihood solution for the Poisson data can be computed using Richardson-Lucy iterative algorithm [MS06]. Its regularized versions using sparsity in wavelet domain and TV-penalty were proposed respectively in [MS06] and [DBFZ⁺06]. For the general tight-frame sparsity penalty, [FBD10] proposed an elegant approach based on variable splitting and alternating optimization. Similar variable splitting approach can also be found in two recent publications: [GRPMSM11] by group of Portilla, and in our paper [DFKE11]. [GRPMSM11] presents further development of the L0-Abs [Por09] algorithm and is designed for deblurring images corrupted by signal dependent Gaussian noise, while in [DFKE11] we use the variable splitting to adopt the general scheme (3.4) for solving Poisson deblurring problem. Below we present the algorithm developed in [GRPMSM11].

Let $\mathbf{z} \in \mathbb{N}_0^N$ ($\mathbb{N}_0 = \mathbb{N} \cup \{0\}$) represents the observed image. It is assumed that \mathbf{z} is a sample of the random vector Z with N independent Poissonian variables with the joint probability distribution

$$P(Z = \mathbf{z} | \mathbf{A}\bar{\mathbf{y}}) = \prod_{i=1}^N \frac{(\mathbf{A}\bar{\mathbf{y}})_i^{z_i} e^{-(\mathbf{A}\bar{\mathbf{y}})_i}}{z_i!}, \quad (3.6)$$

where $\bar{\mathbf{y}}, \mathbf{A}\bar{\mathbf{y}} \in \mathbb{R}_+^N$, ($\mathbb{R}_+ = \{x : x \in \mathbb{R}, x > 0\}$) is the true image, \mathbf{A} is the blur operator and the subindex i denotes the i -th component of the vector. We again would like to use GNE formulation (3.1) to define the deblurring problem which we are going to solve with the help of the iterative scheme (3.4). To do it we, first of all, need to compute the negative log-likelihood function corresponding to the

³Let $\bar{\mathbf{y}}, \mathbf{v} \in \mathbb{R}_+^N$ and $\mathbf{z} \in \mathbb{N}_0^N$ be respectively the true, blurred noise-free and noisy images, s.t.

$$\mathbf{v} = \mathbf{A}\bar{\mathbf{y}} \text{ and } z_i \sim \mathcal{P}(v_i), i = 1, \dots, N.$$

Also let $f, f_{inv} : \mathbb{R} \rightarrow \mathbb{R}$ denote respectively the VST and its unbiased inverse. Note, that $f_{inv} \neq f^{-1}$, where f^{-1} is the inverse function of f .

We have that (see [Foi09])

$$f_{inv}(E\{f(z_i)\}) = E\{z_i\}.$$

Applying f_{inv}^{-1} to the both sides of the equality we obtain

$$E\{f(z_i)\} = f_{inv}^{-1}(E\{z_i\}),$$

and recalling that $E\{z_i\} = v_i$ (since z_i is a Poissonian r.v.) we get

$$E\{f(z_i)\} = f_{inv}^{-1}(v_i).$$

Now, if f is such that approximately $f(z_i) = f_{inv}^{-1}(v_i) + \eta_i, \eta_i \sim \mathcal{N}(0, 1), \forall i$ (for example if f is the Anscombe transform), the fidelity term for \mathbf{v} can be derived as

$$L_{\mathbf{z}}(\mathbf{v}) = \frac{1}{2} \sum_{i=1}^N \left(f(z_i) - f_{inv}^{-1}(v_i) \right)^2.$$

In [CBFZ07] and [DFS09] the fidelity terms are derived using v_i and $f(v_i)$, respectively, instead of $f_{inv}^{-1}(v_i)$, making the final solution to be biased.

model (3.6) which is

$$L_{\mathbf{z}}(\mathbf{y}) = \sum_{i=1}^N [(\mathbf{A}\mathbf{y})_i - z_i \log((\mathbf{A}\mathbf{y})_i) + \log(z_i!)]. \quad (3.7)$$

Here we face the main problem, since unlike the Gaussian noise case $L_{\mathbf{z}}(\mathbf{y})$ is not quadratic and the optimization problem

$$\begin{aligned} \mathbf{y}_t &= \arg \min_{\mathbf{y}, \mathbf{y} > \mathbf{0}} L_{\mathbf{z}}(\mathbf{y}) + \frac{\gamma}{2} \|\mathbf{y} - \Psi\boldsymbol{\omega}_{t-1}\|_2^2 \\ &= \arg \min_{\mathbf{y}, \mathbf{y} > \mathbf{0}} \sum_{i=1}^N [(\mathbf{A}\mathbf{y})_i - z_i \log((\mathbf{A}\mathbf{y})_i)] + \frac{\gamma}{2} \|\mathbf{y} - \Psi\boldsymbol{\omega}_{t-1}\|_2^2, \end{aligned} \quad (3.8)$$

has no closed-form solution. In the last functional we omitted the term $\sum_{i=1}^N \log(z_i!)$ since it is independent of \mathbf{y} . One possible option to solve (3.8) is to apply a Richardson-Lucy-type iterative algorithm. But instead, following [FBD10], we perform variable splitting and try to solve it by an alternating optimization. We introduce an auxiliary variable $\mathbf{v} = \mathbf{A}\mathbf{y}$, and incorporate the constraint $\mathbf{v} = \mathbf{A}\mathbf{y}$ as a penalty term in the criterion

$$L'_1(\mathbf{y}, \boldsymbol{\omega}_t, \mathbf{v}) = \sum_{i=1}^N [v_i - z_i \log(v_i)] + \frac{\xi}{2} \|\mathbf{v} - \mathbf{A}\mathbf{y}\|_2^2 + \frac{\gamma}{2} \|\mathbf{y} - \Psi\boldsymbol{\omega}_{t-1}\|_2^2, \quad (3.9)$$

where $\xi > 0$. The fixed-point problem (3.1) is then reformulated as:

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}, \mathbf{v}} L'_1(\mathbf{y}, \boldsymbol{\omega}^*, \mathbf{v}), & \text{subject to } y_i \geq 0, v_i \geq 0, i = 1 \dots N \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}^*, \boldsymbol{\omega}), \end{cases} \quad (3.10)$$

where L_2 is given by (3.3).

Rigorous solution of (3.10), would require implementing minimization of (3.9) in a nested loop. After each update of $\boldsymbol{\omega}_t$, the new estimates for \mathbf{v}_t and \mathbf{y}_t need to be found by alternating minimization

$$\begin{cases} \mathbf{v}_k = \arg \min_{\mathbf{v}} L'_1(\mathbf{y}_{k-1}, \boldsymbol{\omega}_{t-1}, \mathbf{v}) \\ \mathbf{y}_k = \arg \min_{\mathbf{y}} L'_1(\mathbf{y}, \boldsymbol{\omega}_{t-1}, \mathbf{v}_k), \end{cases}, \quad k = 0, 1, \dots$$

where the weight of the penalty term ξ is gradually increased to ensure the equality $\mathbf{v}_t = \mathbf{A}\mathbf{y}_t$. In practice we do not need constraint $\mathbf{v} = \mathbf{A}\mathbf{y}$ to hold strictly, it is enough that \mathbf{v}_t and $\mathbf{A}\mathbf{y}_t$ remain close through the reconstruction process. For this reason we propose solving (3.10) with a simpler, one-loop alternating minimization procedure:

$$\begin{cases} \mathbf{v}_t = \arg \min_{\mathbf{v}} L'_1(\mathbf{y}_{t-1}, \boldsymbol{\omega}_{t-1}, \mathbf{v}) \\ \mathbf{y}_t = \arg \min_{\mathbf{y}} L'_1(\mathbf{y}, \boldsymbol{\omega}_{t-1}, \mathbf{v}_t) \\ \boldsymbol{\omega}_t = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}_t, \boldsymbol{\omega}) \end{cases}, \quad t = 0, 1, \dots \quad (3.11)$$

Minimization of L_1' is separable with respect to the components of \mathbf{v} . For each component $(\mathbf{v}_t)_i$ it takes form

$$(\mathbf{v}_t)_i = \arg \min_{\mathbf{v}_i, \mathbf{v}_i \geq 0} \left\{ v_i - z_i \log v_i + \frac{\xi}{2} (v_i - (\mathbf{A}\mathbf{y}_{t-1})_i)^2 \right\},$$

for which the non-negative solution is given by the formula

$$(\mathbf{v}_t)_i = \frac{1}{2} \left((\mathbf{A}\mathbf{y}_{t-1})_i - \frac{1}{\xi} + \sqrt{\left((\mathbf{A}\mathbf{y}_{t-1})_i - \frac{1}{\xi} \right)^2 + \frac{4}{\xi} z_i} \right).$$

Since L_1' is quadratic with respect to \mathbf{y} , its minimization results in the linear solution:

$$\mathbf{y}_t = \left(\mathbf{A}^T \mathbf{A} + \frac{\gamma}{\xi} \mathbf{I} \right)^{-1} \left(\mathbf{A}^T \mathbf{v}_t + \frac{\gamma}{\xi} \max \{ \Psi \boldsymbol{\omega}_{t-1}, 0 \} \right).$$

The elementwise maximum operation is applied to the $\Psi \boldsymbol{\omega}_{t-1}$ to ensure the non-negativity of the estimate \mathbf{y}_t .

The minimization of L_2 with respect to $\boldsymbol{\omega}$ is obtained by the soft or hard thresholding of $\Phi \mathbf{y}_t$.

Summarizing all steps we arrive to the Algorithm 4 derived in [DFKE11] and called Poissonian Iterative Decoupled Deblurring with BM3D-frames (PIDD-BM3D).

Algorithm 4 PIDD-BM3D - Poissonian Iterative Decoupled Deblurring with BM3D-frames

- 1: **input:** $\mathbf{z}, \mathbf{A}, \hat{\mathbf{y}}_{\text{init}}, \tau, \gamma, \xi$
 - 2: **initialization:**
 - 3: *using* $\hat{\mathbf{y}}_{\text{init}}$ *construct operators* Φ *and* Ψ
 - 4: *set:* $t = 0, \mathbf{y}_0 = \hat{\mathbf{y}}_{\text{init}}, \boldsymbol{\omega}_0 = \Phi \hat{\mathbf{y}}_{\text{init}}$
 - 5: **repeat**
 - 6: $t = t + 1$
 - 7: $(\mathbf{v}_t)_i = \frac{1}{2} \left((\mathbf{A}\mathbf{y}_{t-1})_i - \frac{1}{\xi} + \sqrt{\left((\mathbf{A}\mathbf{y}_{t-1})_i - \frac{1}{\xi} \right)^2 + \frac{4}{\xi} z_i} \right)$
 - 8: $\mathbf{y}_t = \left(\mathbf{A}^T \mathbf{A} + \frac{\gamma}{\xi} \mathbf{I} \right)^{-1} \left(\mathbf{A}^T \mathbf{v}_t + \frac{\gamma}{\xi} \max \{ \Psi \boldsymbol{\omega}_{t-1}, 0 \} \right)$
 - 9: $\boldsymbol{\omega}_t = Th_{\tau}(\Phi \mathbf{y}_t)$
 - 10: **until** convergence.
 - 11: **output:** $\hat{\mathbf{y}} = \Psi \boldsymbol{\omega}_t$
-

3.2 Implementation of IDD-BM3D and PIDD-BM3D algorithms

Our first step towards obtaining an efficient implementation of IDD-BM3D algorithm will be decoupling the blur inversion and thresholding/filtering operations.

Denoting $\tilde{\mathbf{y}}_t = \Psi \omega_t$, we rewrite steps 7 and 8 as

$$\mathbf{y}_t = (\mathbf{A}^T \mathbf{A} + \gamma \mathbf{I})^{-1} (\mathbf{A}^T \mathbf{z} + \gamma \tilde{\mathbf{y}}_{t-1}), \quad (3.12)$$

$$\tilde{\mathbf{y}}_t = \Psi T h_\tau (\Phi \mathbf{y}_t). \quad (3.13)$$

If blur is spatially invariant, it can be represented as a convolution of the true image against the blurring kernel⁴, and formula (3.12) can be computed in the Discrete Fourier Transform domain:

$$y_t = \mathcal{F}^{-1} \left(\frac{\mathcal{F}^*(h) \circ \mathcal{F}(z) + \gamma \mathcal{F}(\tilde{y}_{t-1})}{\mathcal{F}^*(h) \circ \mathcal{F}(h) + \gamma} \right).$$

Here y_t, \tilde{y}_t, z are $\sqrt{N} \times \sqrt{N}$ size images corresponding to the vectors $\mathbf{y}_t, \tilde{\mathbf{y}}_t, \mathbf{z}$, h is the 2-D blurring kernel, and operators \mathcal{F} and \mathcal{F}^{-1} are respectively the direct and inverse 2-D FFT. The product 'o' and division operations are performed elementwise.

Further, due to the particular structure of operators Φ and Ψ , the thresholding in BM3D-frame domain $\Psi T h_\tau (\Phi \mathbf{y}_t)$ can be computed by processing \mathbf{y}_t group by group, each time accessing only a small portion of the data corresponding to the particular group. This allows to avoid explicit manipulation with big vectors and matrices such as Ψ, Φ and ω_t .

The various image reconstruction experiments performed with BM3D-frames, including denoising experiments in Section 2.3 and deblurring experiments in Section VII A [DKE12], suggest that the best reconstruction quality is achieved when shrinkage is performed by hard thresholding and the synthesis frame Ψ is constructed with the group weights selected adaptively. This settings we consider to be the default for IDD-BM3D algorithm, and all experimental results which we report later in this chapter refer to this default implementation.

The above comments regarding the implementation of IDD-BM3D also apply to the Poissonian deblurring algorithm PIDD-BM3D.

3.3 Selection of regularization parameters

The problem of optimal selection of regularization parameters remains one of the key problems in inverse imaging. Lacking practical automatic estimation procedure, we suggest tuning the regularization parameters on a set of test images, and using the found parameter values for reconstruction of other images, under the same degradation conditions. Also, we wish to demonstrate that solutions of our GNE problems indeed are good estimates. Hence our target is to identify parameters which will result in stable convergence and, at the same time, provide high reconstruction quality. To find such parameters we designed an automatic procedure. We first fix the number of iterations taking it big enough to ensure convergence (e.g., in our experiments we use 200 iterations, while in most cases 50-100 iterations are enough to reach the convergence state). Then, for a given

⁴This function is usually termed *point spread function (PSF)*, reflecting the fact that the blurring kernel coincides with the image of a point source object.

set of regularization parameters, e.g., (τ, γ) , we evaluate the reconstruction by computing the criterion

$$F_{\bar{\mathbf{y}}}(\tau, \gamma) = PSNR_{\bar{\mathbf{y}}}(\hat{\mathbf{y}}_{t_{\text{final}}}, \bar{\mathbf{y}}) - 10 |PSNR(\hat{\mathbf{y}}_{t_{\text{final}}}, \bar{\mathbf{y}}) - PSNR(\hat{\mathbf{y}}_{0.75t_{\text{final}}}, \bar{\mathbf{y}})|, \quad (3.14)$$

where $\bar{\mathbf{y}}$ is the ground truth image and $\hat{\mathbf{y}}_{0.75t_{\text{final}}}, \hat{\mathbf{y}}_{t_{\text{final}}}$ are the estimates after performing respectively 75 and 100 percent of iterations. The criterion penalizes the final PSNR score by its change in the last 25 percent. In other words, we wish that PSNR do not change in the last 25 percent of iterations, indicating stable convergence. The optimal parameters are considered those maximizing $F_{\bar{\mathbf{y}}}(\tau, \gamma)$. We find them by computing

$$\arg \min_{\tau, \gamma} -F_{\bar{\mathbf{y}}}(\tau, \gamma)$$

using Nelder-Mead simplex minimization algorithm (implemented in Matlab's `fminsearch` function). To avoid overfitting parameters to a single image, we suggest minimizing the sum $-\sum_{i=1}^k F_{\bar{\mathbf{y}}_i}(\tau, \gamma)$ over a number of test images.

Note, that if we try to maximize PSNR without penalizing it, the found parameters, most probably, will correspond to the case when the reconstruction reaches the highest PSNR after t_{final} iterations, but then degrades rapidly if iterations continue. Which means that the algorithm did not reach the convergence state.

3.4 Experiments

We evaluate IDD-BM3D and PIDD-BM3D algorithms in simulated experiments, and compare them against the other algorithms known to produce best results in the field.

3.4.1 Gaussian data

We consider six deblurring scenarios used as the benchmarks in the number of publications (e.g., [GCMP08] and [DFKE08a]). The blur point-spread functions $h(x_1, x_2)$ (blurring kernels) and the noise variances σ^2 for each scenario are summarized in Table 3.1. In our experiments we use eight images: *Cameraman*, *Lena*, *House*, *Barbara*, *Boats*, *Hill*, *Man and Peppers*. The original, undistorted versions of the first four of these images were used as "oracles" in order to identify the optimal values for the parameters τ and γ of the IDD-BM3D algorithm. Optimization was performed separately for each deblurring scenario, and the obtained optimal values are used for all eight images.

For the comparison we also evaluate performance of four other algorithms. The first algorithm is the L0-Abs [Por09]; we use it with the tight frame obtained by concatenating the corresponding vectors of two individual Parseval frames: Dual Tree Complex Wavelet [Kin01], and Translation Invariant Haar Pyramid [GCMP08]. The next two algorithms: TVMM [OBDF09] and CGMK [CGMK10], both rely on TV priors; TVMM is an efficient solver for the classical, analysis-based TV prior, while CGMK is utilizing a more advanced modeling based on a

mixture of spatially weighted TV priors. Finally, the last algorithm which we are going to evaluate is BM3DDEB [DFKE08a], which is a ForWaRD-type, two-step algorithm. It uses BM3D-domain hard-thresholding and wiener filtering as the denoising for its first and second reconstruction steps, respectively.

To make our comparison fair, we used "oracle" images also to optimize regularization parameters of the L0-AbS algorithm. The default regularization parameters of algorithm BM3DDEB were already optimized in a similar manner by the authors of the algorithm. The remaining two algorithms, CGMK and TVMM, do not take any external parameters, instead they estimate optimal regularization parameters automatically.

The PSNR values of the reconstructed images are summarized in Table 3.2. To make comparison easier we plotted the same data in Figure 3.1. As we may see, in most cases the IDD-BM3D algorithm provides the best results. Exception is the case where the blur is very small while the noise level is relatively high (as in scenario 6). In this case, reconstruction is essentially reduces to the denoising problem, where BM3DDEB has some advantage due to the accurate modelling of the colored noise.

The visual quality of some of the restored images can be evaluated from Figures 3.2-3.4, where for the comparison we show results by the closest competitors [CGMK10], [Por09] and [DFKE08a]. Analyzing Figure 3.2 we can see that the proposed algorithm is able to suppress the ringing artefacts better than BM3DDEB and provides sharper image edges. The latter effect is achieved in particular due to the smaller block size used in IDD-BM3D compared to BM3DDEB. Figures 3.3 and 3.4 illustrate the advantage of the BM3D modeling, demonstrating that the algorithms based on the BM3D reconstruct shape and texture better than the algorithms based on the other types of modeling.

3.4.2 Poissonian data

We compare PIDD-BM3D with two Poissonian deblurring algorithms known to produce the best results in the field: PIDAL [FBD10] and DFS [DFS09]⁵. As in the experiments with Gaussian data, we consider several degradation scenarios which are described in Table 3.3.

Parameters of PIDD-BM3D algorithm are selected as follows. In all scenarios ξ is fixed and equal to 0.001. The other two parameters, τ, γ , are different for each scenario, but common for all test images. The optimal values for parameters τ and γ were found using the procedure described in Section 3.3, where 100×100 pixel fragment of Cameraman image was used as the test image.

PIDD-BM3D requires an initial estimate on which it can apply block-matching to construct groups, for that role we decided to test outputs of PIDAL and DFS algorithms. Note, that initialization images are used *only* for block-matching, the iterations are initialized with the observation \mathbf{z} .

Examining Table 3.4 which presents the numerical evaluation of the results, we notice that in most cases PIDD-BM3D is able to obtain estimates which have

⁵Images reconstructed with PIDAL-TV algorithm were kindly provided by Mario Figueiredo. Results of the DFS algorithm were reproduced using the code available online at <https://fadili.users.greyc.fr/demos/WaveRestore/downloads/PoissonDeconv100.tar.gz>.

significantly better MAE and PSNR scores than the estimates used for initialization. Note also consistency of the PIDD-BM3D algorithm: better initial estimates are translated in better reconstructed images.

In Figures 3.5-3.7 we show some of the reconstructed images. We can see that images obtained with PIDD-BM3D are sharper and contain more details than ones obtained with DFS and PIDAL-TV. For example, compare the reconstruction of the leg of a mushroom-shaped spine in Figure 3.5.

From the drawbacks, we should mention artefacts which PIDD-BM3D tends to create in the areas with uniform or slow varying intensity (e.g., see sky in *Cameraman* image or body of the neuron in *Neuron Phantom*). This phenomenon is known, it appears also in non-iterative denoising with BM3D algorithm and it is generally considered to arise due to the erroneous block-matching.

Scenario	PSF	σ^2
1	$1/(1+x_1^2+x_2^2)$, $x_1, x_2 = -7, \dots, 7$	2
2	$1/(1+x_1^2+x_2^2)$, $x_1, x_2 = -7, \dots, 7$	8
3	9×9 uniform	≈ 0.3
4	$[1\ 4\ 6\ 4\ 1]^T [1\ 4\ 6\ 4\ 1]/256$	49
5	Gaussian with $std = 1.6$	4
6	Gaussian with $std = 0.4$	64

Table 3.1: Blur PSF $h(x_1, x_2)$ and noise variance used in each Gaussian data deblurring scenario. All PSFs are normalized so that $\sum h = 1$.

Method\Scenario	1	2	3	4	5	6
<i>Cameraman (256 × 256)</i>						
BSNR	31.87	25.85	40.00	18.53	29.19	17.76
Input PSNR	22.23	22.16	20.76	24.62	23.36	29.82
BM3DDEB [DFKE08a]	8.20	6.46	8.37	3.35	3.72	4.70
L0-Abs [Por09]	7.67	5.77	8.96	2.99	3.58	3.40
TVMM [OBDF09]	7.41	5.17	8.54	2.57	3.36	1.30
CGMK [CGMK10]	7.80	5.49	9.15	2.80	3.54	3.33
IDD-BM3D [DKE12]	8.89	7.16	10.48	4.00	4.34	4.89
<i>Lena (512 × 512)</i>						
BSNR	29.89	23.87	40.00	16.47	27.18	15.52
Input PSNR	27.25	27.04	25.84	28.81	29.16	30.03
BM3DDEB [DFKE08a]	7.96	6.55	7.99	4.80	4.34	6.42
L0-Abs [Por09]	6.96	5.64	7.97	4.06	4.21	4.28
TVMM [OBDF09]	6.36	4.98	7.47	3.52	3.61	2.79
CGMK [CGMK10]	6.76	5.37	7.86	3.49	3.93	4.46
IDD-BM3D [DKE12]	7.97	6.61	8.92	4.97	4.85	6.34
<i>House (256 × 256)</i>						
BSNR	29.16	23.14	40.00	15.99	26.61	15.15
Input PSNR	25.61	25.46	24.11	28.06	27.81	29.98
BM3DDEB [DFKE08a]	9.34	8.22	10.94	5.20	4.59	7.34
L0-Abs [Por09]	8.60	7.09	11.48	4.56	4.79	4.79
TVMM [OBDF09]	7.98	6.57	10.39	4.12	4.54	2.44
CGMK [CGMK10]	8.31	6.97	10.75	4.48	4.97	4.59
IDD-BM3D [DKE12]	9.95	8.55	12.89	5.79	5.74	7.13
<i>Barbara (512 × 512)</i>						
BSNR	30.81	24.79	40.00	17.35	28.07	16.59
Input PSNR	23.34	23.25	22.49	24.22	23.77	29.78
BM3DDEB [DFKE08a]	7.88	4.13	5.91	2.05	1.29	5.85
L0-Abs [Por09]	3.23	1.68	3.81	0.78	0.83	2.55
TVMM [OBDF09]	3.10	1.33	3.49	0.41	0.75	0.59
CGMK [CGMK10]	2.45	1.34	3.55	0.44	0.81	0.38
IDD-BM3D [DKE12]	7.64	3.96	6.05	1.88	1.16	5.45

Table 3.2: Gaussian data deblurring. ISNR scores of the reconstructions obtained with different methods. Rows labeled "Input PSNR" show PSNRs of the input blurry images.

Method\Scenario	1	2	3	4	5	6
<i>Boats (512×512)</i>						
BSNR	29.37	23.35	40.00	16.06	26.70	15.26
Input PSNR	25.00	24.88	23.36	27.10	26.41	29.97
BM3DDEB [DFKE08a]	7.35	5.61	8.48	3.35	3.78	4.41
L0-Abs [Por09]	6.70	5.04	8.98	2.86	3.52	3.00
TVMM [OBDF09]	6.14	4.46	8.07	2.45	3.26	1.42
CGMK [CGMK10]	6.57	4.82	8.71	2.74	3.49	3.11
IDD-BM3D [DKE12]	7.69	5.96	9.77	3.60	4.10	4.46
<i>Hill (512×512)</i>						
BSNR	30.19	24.17	40.00	16.68	27.41	15.74
Input PSNR	26.51	26.33	25.04	27.74	27.83	29.99
BM3DDEB [DFKE08a]	5.77	4.46	6.97	2.94	2.75	4.07
L0-Abs [Por09]	5.39	4.06	7.26	2.76	2.84	2.86
TVMM [OBDF09]	5.02	3.62	6.73	2.48	2.59	1.67
CGMK [CGMK10]	5.30	3.79	7.14	2.51	2.76	3.40
IDD-BM3D [DKE12]	6.03	4.69	7.78	3.29	3.31	4.13
<i>Man (512×512)</i>						
BSNR	29.72	23.70	40.00	16.33	27.02	15.45
Input PSNR	25.95	25.80	24.51	27.65	27.41	29.99
BM3DDEB [DFKE08a]	6.38	4.82	6.92	3.09	3.22	4.45
L0-Abs [Por09]	5.90	4.42	7.30	2.69	3.04	3.05
TVMM [OBDF09]	5.74	4.17	7.01	2.52	3.03	1.78
CGMK [CGMK10]	5.85	4.34	7.37	2.65	3.24	3.62
IDD-BM3D [DKE12]	6.66	5.12	7.89	3.37	3.61	4.54
<i>Peppers (256×256)</i>						
BSNR	29.99	23.97	40.00	17.01	27.57	16.36
Input PSNR	22.60	22.53	21.33	24.77	23.89	29.83
BM3DDEB [DFKE08a]	9.58	7.88	10.05	3.31	3.30	5.23
L0-Abs [Por09]	7.45	5.24	11.13	1.57	2.05	3.38
TVMM [OBDF09]	7.76	5.02	10.10	2.03	2.59	1.37
CGMK [CGMK10]	8.88	5.17	10.36	2.19	2.69	3.30
IDD-BM3D [DKE12]	10.48	8.68	12.07	4.45	4.53	5.49

Table 3.2: Gaussian data deblurring (continued). ISNR scores of the reconstructions obtained with different methods. Rows labeled "Input PSNR" show PSNRs of the input blurry images.

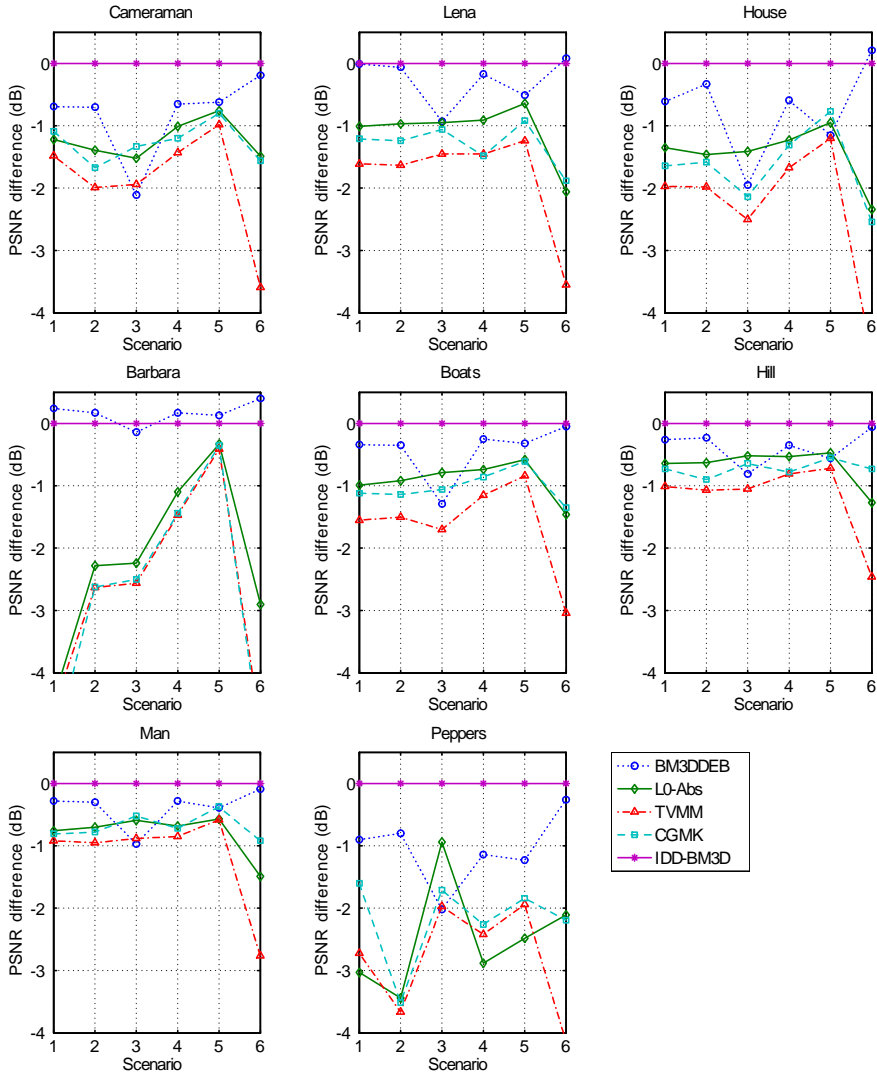


Figure 3.1: Relative PSNR performance of the considered deblurring algorithms. The results of the IDD-BM3D deblurring algorithm [DKE12] are considered as the baselines.



Figure 3.2: Deblurring of the *Cameraman* image, scenario 3. From left to right and from top to bottom are presented zoomed fragments of the following images: original, blurred noisy, reconstructed by CGMK [CGMK10] (ISNR 9.15), L0-Abs [Por09] (ISNR 8.96), DEB-BM3D [DFKE08a] (ISNR 8.37) and by proposed IDD-BM3D method (ISNR 10.48).



Figure 3.3: Deblurring of the *Man* image, scenario 2. From left to right and from top to bottom are presented zoomed fragments of the following images: original, blurred noisy, reconstructed by CGMK [CGMK10] (ISNR 4.34), L0-AbS [Por09] (ISNR 4.42), DEB-BM3D [DFKE08a] (ISNR 4.82) and by proposed IDD-BM3D method (ISNR 5.12).

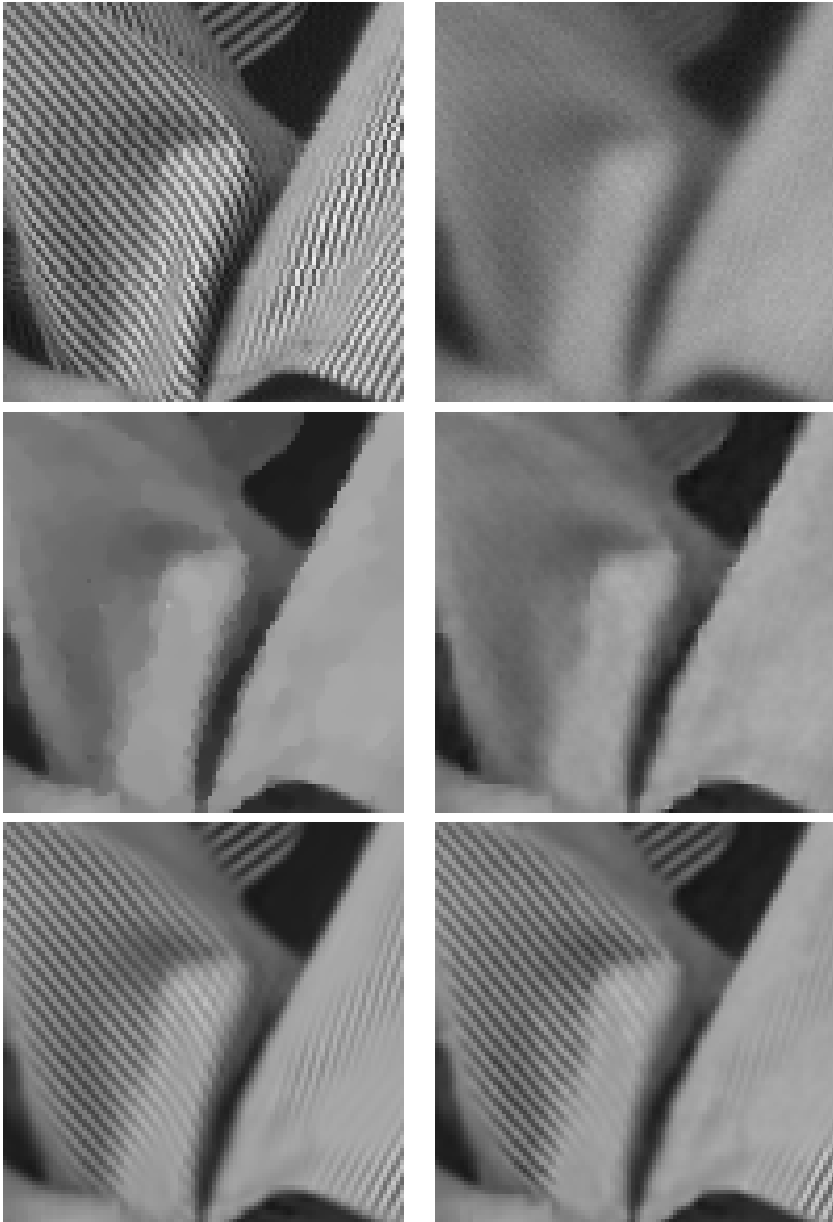


Figure 3.4: Deblurring of the *Barbara* image, scenario 2. From left to right and from top to bottom are presented zoomed fragments of the following images: original, blurred noisy, reconstructed by CGMK [CGMK10] (ISNR 1.34), L0-Abs [Por09] (ISNR 1.68), DEB-BM3D [DFKE08a] (ISNR 4.13) and by proposed IDD-BM3D method (ISNR 3.96).

Scenario	PSF	Maximum intensity
1	7×7 uniform	5
2	7×7 uniform	30
3	7×7 uniform	100
4	7×7 uniform	255

Table 3.3: Blur PSF $h(x_1, x_2)$ and maximum true image intensity M used in each Poisson data deblurring scenario. All PSFs are normalized so that $\sum h = 1$.

PSNR				
Method \ Scenario	1	2	3	4
<i>Neuron phantom (128 × 128)</i>				
DFS [DFS09]	20.90	24.40	24.40	26.46
PIDAL-TV [FBD10]	22.86	24.70	26.36	27.99
PIDD-BM3D (matching on DFS)	23.57	26.31	27.99	29.49
PIDD-BM3D (matching on PIDAL-TV)	24.46	26.65	28.19	29.71
<i>Cameraman (256 × 256)</i>				
DFS [DFS09]	10.46	21.42	21.59	22.31
PIDAL-TV [FBD10]	20.74	21.95	22.03	23.07
PIDD-BM3D (matching on DFS)	20.08	22.16	23.32	24.25
PIDD-BM3D (matching on PIDAL-TV)	21.24	22.59	23.49	24.47
<i>Cell (256 × 256)</i>				
DFS [DFS09]	22.15	27.60	27.71	24.39
PIDAL-TV [FBD10]	25.08	26.88	27.07	29.14
PIDD-BM3D (matching on DFS)	22.62	27.86	29.08	29.49
PIDD-BM3D (matching on PIDAL-TV)	24.97	27.81	29.16	30.55

MAE				
Method \ Scenario	1	2	3	4
<i>Neuron phantom (128 × 128)</i>				
DFS [DFS09]	0.2592	0.9184	3.4546	5.8221
PIDAL-TV [FBD10]	0.1834	0.7681	1.9737	3.9599
PIDD-BM3D (matching on DFS)	0.1611	0.6589	1.6904	3.6055
PIDD-BM3D (matching on PIDAL-TV)	0.1349	0.5883	1.6755	3.4645
<i>Cameraman (256 × 256)</i>				
DFS [DFS09]	1.1041	1.4555	4.7494	10.4984
PIDAL-TV [FBD10]	0.2747	1.3044	4.0740	9.1937
PIDD-BM3D (matching on DFS)	0.2592	1.3021	3.7298	8.3042
PIDD-BM3D (matching on PIDAL-TV)	0.2361	1.2118	3.5940	7.9363
<i>Cell (256 × 256)</i>				
DFS [DFS09]	0.1537	0.5810	1.7695	5.7309
PIDAL-TV [FBD10]	0.1223	0.5892	1.7860	3.9293
PIDD-BM3D (matching on DFS)	0.1328	0.4942	1.4593	3.7271
PIDD-BM3D (matching on PIDAL-TV)	0.1073	0.4961	1.4537	3.1991

Table 3.4: Poissonian data deblurring. PSNR and MAE scores of the reconstructions obtained with different methods. For PIDD-BM3D algorithm in the parenthesis is mentioned the estimate used for constructing groups of BM3D-frames, i.e., the image on which block-matching was performed. Scores are calculated over the central part of the images, omitting 10-pixel wide borders.

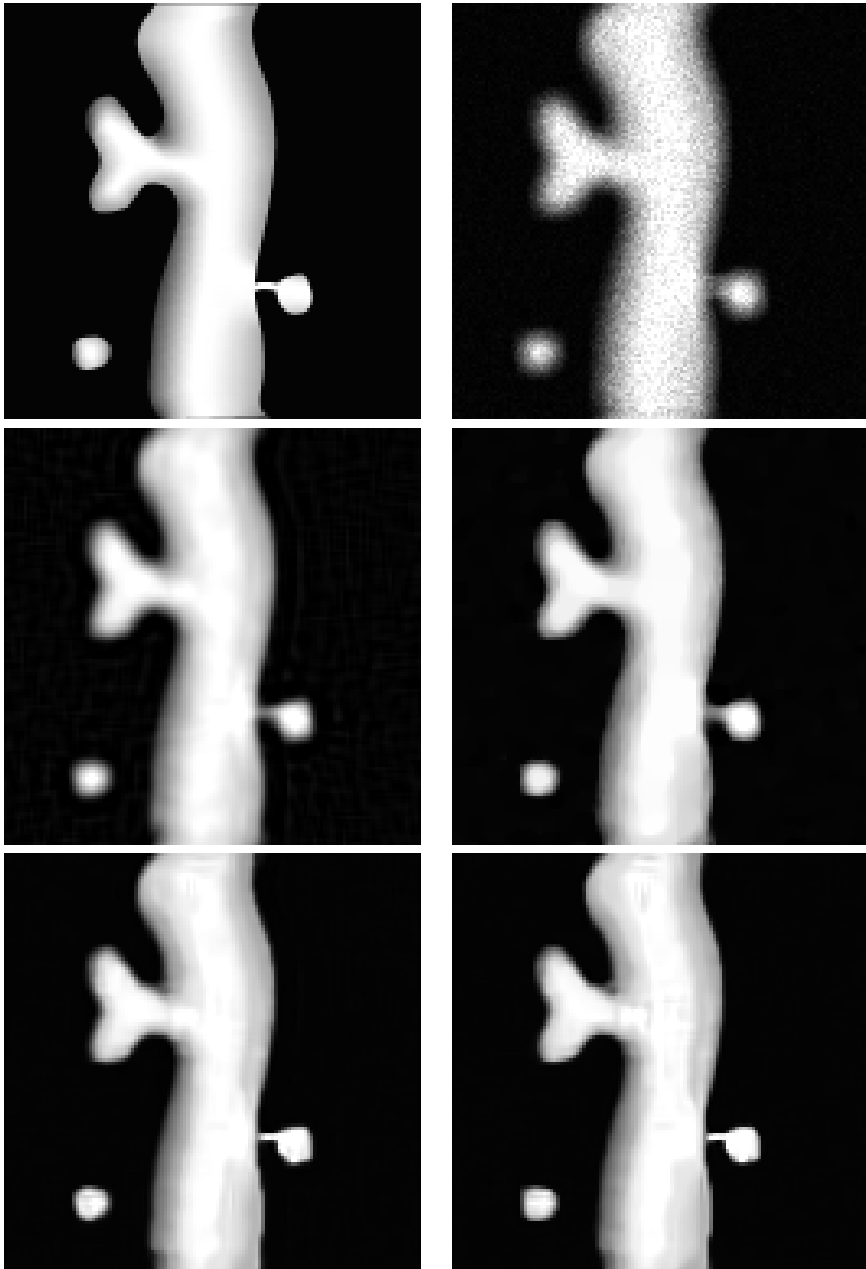


Figure 3.5: Deblurring of the *Neuron phantom* image, scenario 4. From left to right and from top to bottom are presented the following images: original, blurred noisy, reconstructed by DFS[DFS09] (PSNR 26.46, MAE 5.82), PICAL-TV[FBD10] (PSNR 27.99, MAE 3.96), PIDD-BM3D with constructing frames using the result of DFS (PSNR 29.49, MAE 3.60), PIDD-BM3D with constructing frames using the result of PICAL-TV (PSNR 29.71, MAE 3.46).



Figure 3.6: Deblurring of the *Cameraman* image, scenario 3. From left to right and from top to bottom are presented the following images: original, blurred noisy, reconstructed by DFS[DFS09] (PSNR 21.59, MAE 4.75), PIDAL-TV[FBD10] (PSNR 22.03, MAE 4.07), PIDD-BM3D with constructing frames using the result of DFS (PSNR 23.32, MAE 3.73), PIDD-BM3D with constructing frames using the result of PIDAL-TV (PSNR 23.49, MAE 3.59).

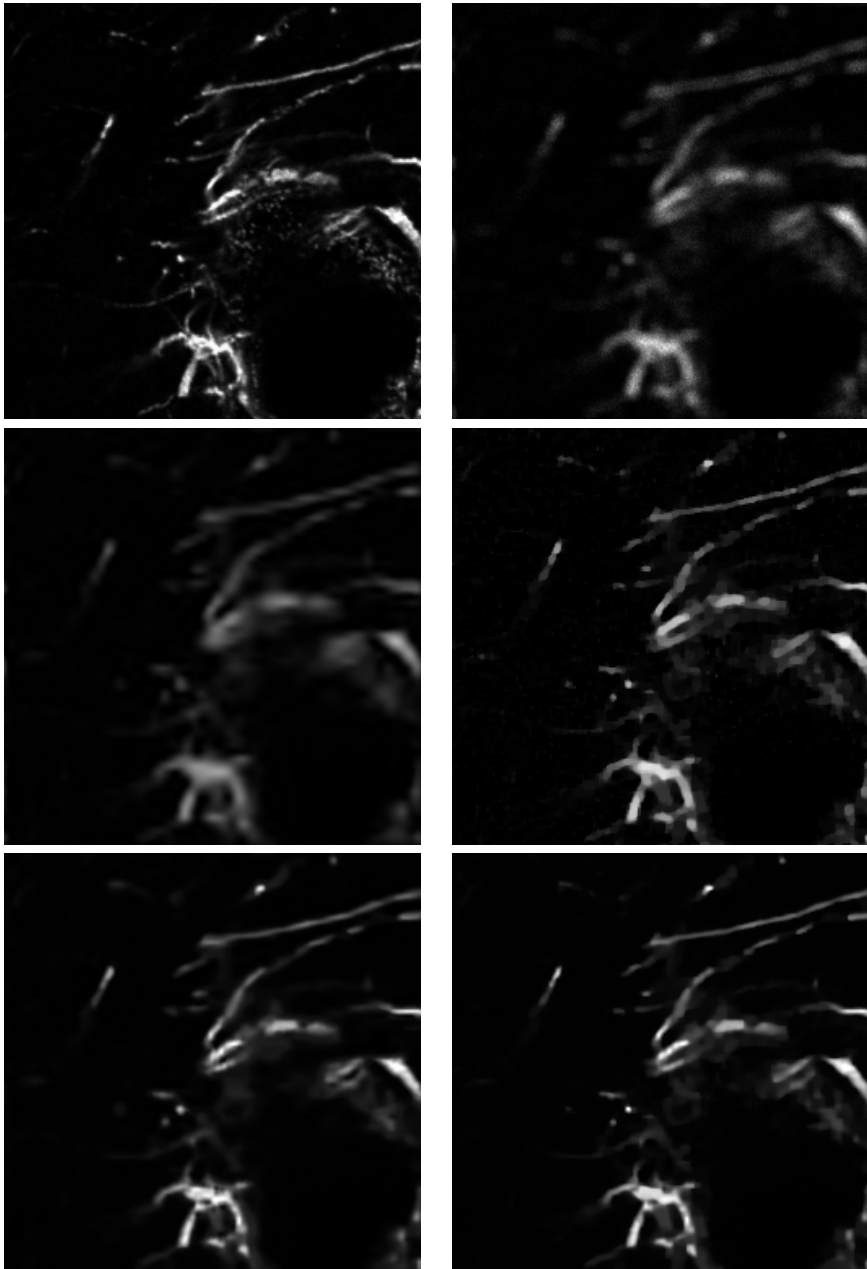


Figure 3.7: Deblurring of the *Cell* image, scenario 4. From left to right and from top to bottom are presented the following images: original, blurred noisy, reconstructed by DFS[DFS09] (PSNR 24.39, MAE 5.73), PIVAL-TV[FBD10] (PSNR 29.14, MAE 3.93), PIDD-BM3D with constructing frames using the result of DFS (PSNR 29.49, MAE 3.73), PIDD-BM3D with constructing frames using the result of PIVAL-TV (PSNR 30.55, MAE 3.20).

Chapter 4

Super-resolution

Image *upsampling* or *zooming*, can be defined as the process of resampling a *single* low-resolution (LR) image on a high-resolution (HR) grid. Different resampling methods [LOct], [NLD07], [LLP06], [DFKE08d] can be used to obtain zoomed images with specific desired properties, such as edge preservation, degree of smoothness, etc. However, fine details missing or distorted in the low-resolution image cannot be reconstructed in the upsampled one. Roughly speaking, there is no sufficient information in the low-resolution image to do this. The situation changes when a number of LR images portraying slightly different views of the same scene are available. The reconstruction algorithm now can try to improve the spatial resolution by incorporating into the final HR result the additional new details revealed in each LR image. The process of combining a sequence of undersampled and degraded low-resolution images in order to produce a single high-resolution image is commonly referred to as *super-resolution (SR) image reconstruction*, or, simply, *image super-resolution*.

The classical formulation of the image SR problem [TH84], [EHO01], [FREM04b] assumes that the relation between the set of observed LR images and the unknown HR image is given by the model

$$z^i = \mathcal{D}(\mathcal{A}(\mathcal{F}^i(\bar{y}))), \quad i = 1, \dots, k, \quad (4.1)$$

where z^i are the $\sqrt{N} \times \sqrt{N}$ size LR images, \mathcal{F}_i , \mathcal{A} , \mathcal{D} are the linear operators representing respectively warp, blur and decimation, and \bar{y} is a $d\sqrt{N} \times d\sqrt{N}$ size HR image of the scene subject to reconstruction. The blur operator \mathcal{A} models the PSF of the imaging system, which is typically considered to be known. The decimation factor d of the operator \mathcal{D} is given by the ratio between the intended size of the HR image and the actual size of LR images. Warp operators are always considered being unknown.

Assuming that the warp operators \mathcal{F}^i can be estimated precisely, problem (4.1) represents linear inverse problem with kN linear equation and d^2N unknowns. When the number of available LR images k is less than d^2 it is obviously an underdetermined, ill-posed problem. However, having $k \geq d^2$ LR images does not guarantee well-posedness, since first, information in the LR images may be redundant and second, the blur operator may not be invertible. Hence, in the

general case we are dealing with an ill-posed inverse problem where additionally we need to estimate the k warp operators.

The observation model (4.1) can be extended to the more general form

$$z^i = \mathcal{D}(\mathcal{A}(\bar{y}^i)), \quad i = 1, \dots, k, \quad (4.2)$$

where the warp operators on a single HR image are replaced by a sequence of HR images $\{\bar{y}^i\}_{i=1}^k$. It is assumed that the sequence $\{\bar{y}^i\}_{i=1}^n$ depicts essentially the same scene. Nevertheless, the scene itself may not be static. The deformation and changes between the frames¹ may be complex, caused, for example, by the relative motion between different objects in the scene or by occlusions. The reconstruction of the HR sequence $\{\bar{\mathbf{y}}^i\}_{i=1}^k$ from the LR sequence $\{\mathbf{z}^i\}_{i=1}^k$ is termed *video super-resolution*.

The classical SR approach is loosely based on the following three steps: 1) registration of the LR images to a HR coordinate grid, 2) warping of the LR images onto that grid by interpolation, and 3) fusion of the warped images into the final HR image. An additional deblurring step is sometimes considered to compensate the blur. A good review of the algorithms based on such classical approach can be found, e.g., in [FREM04a].

In practice, estimation of the warp operators constitutes the main difficulty of the SR problem. For successful reconstruction it is crucial to perform registration between the features represented across different frames with sub-pixel (with respect to the LR grid) accuracy. Note, that the LR images are aliased² which additionally hardens the registration task. Most of the classical SR methods rely either on a parametric global motion estimation, or on a computationally intensive optical flow calculation. However, an explicit registration of the LR frames is often not feasible: on the one hand, if the registration map has few degrees of freedom, it is too rigid to model the geometrical distortions caused by the lens system; on the other hand, when many degrees of freedoms are available (e.g., a dense optical flow), reliable estimates of the registration parameters cannot be obtained. In either case, registration artefacts are likely to appear in the fusion, requiring heavy regularization (smoothing) for their concealment [FREM04a]. The situation becomes even more difficult when non-global motion is present in images, something that is typical of video SR. The modern SR algorithms [DFKE08c], [EV08], [PETM09], overcome this problem by defining correspondence between small blocks of different frames. The correspondence is determined based on the similarities of the blocks. To make the estimation more robust, a given block may be put in a correspondence to several blocks in another frame.

Although the mentioned algorithms share a similar registration approach, their data-fusion strategies are very different. Algorithms [EV08], [PETM09] use Non-local Means, [TMPE09] is based on the Kernel Regression technique, while our algorithm from [DFKE08c], uses the BM3D modeling.

¹We use the term "frame" referring both to individual images of a video sequence and to the frame of a vector space. The actual meaning of the term should be clear from the context.

²The case when LR images are not aliased is not interesting from SR point of view. Being not aliased means that images are sampled above the Nyquist rate and the decimation process does not introduce loss of information. Hence, in the non-aliased case all LR images contain the same information and the reconstruction of the HR image reduces to the deblurring of one of the LR image followed by resampling with *sinc* kernel.

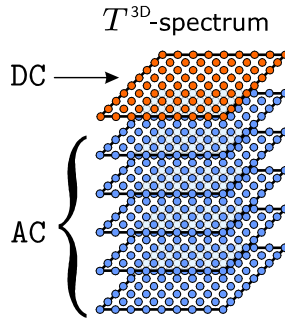


Figure 4.1: The DC and AC planes of the 3-D transform of a group. The DC coefficients are marked with red and the AC coefficients are blue.

In this chapter we discuss the role of BM3D-modeling in SR from a perspective which differs from the one of [DFKE08c] and [DFKE10]. We show that 3-D transform domain modeling can be considered not only as a regularization tool, but also as an efficient data fusion approach. This important observation will help us to explain the efficacy of the SR algorithm based on iterative BM3D filtering [DFKE08c] (later in the text we use abbreviation IFSR for referring to this algorithm) and also to suggest more efficient ways of implementing it. Finally, we develop a new SR algorithm based on the BM3D-modeling, deriving it as a solution of a GNE problem. Its key difference from IFSR is the way how different 3-D spectral component are treated. Further, in contrast to IFSR, the new algorithm is directly designed to handle noisy data.

4.1 Fusion via collaborative filtering

Consider a 3-D decorrelating transform constructed as a separable combination of a 2-D orthogonal transform applied to each block and a 1-D orthogonal transform applied along the third dimension (across the corresponding coefficients of the blocks' 2-D spectra). Suppose that the 1-D transform has a DC element (constant basis function), then the coefficients of the 3-D spectrum can be divided in two parts according to their role: the *DC plane* coefficients, which represent the 2-D spectrum of the average of the grouped blocks, and coefficients at the *AC planes* representing the differences between the grouped blocks (see illustration in Figure 4.1). In what follows we will refer to the coefficients in these two groups simply as the DC and AC coefficients omitting the word 'plane'. Given a group of blocks, we can perform inter-block smoothing (fusion) by computing 3-D transform, thresholding the AC coefficients and then applying inverse transform. Smoothing level can be controlled by changing the threshold value; higher threshold will result in stronger smoothing. In the extreme case, if we zero-out all AC coefficients, after inversion of the 3-D transform all block estimates will be equal to the pixelwise average of the original blocks.

Fused data often requires additional spatial regularization. In the 3-D transform domain spatial regularization can be implemented by thresholding DC coef-

ficients. Thresholding produces sparser representation of the average block with respect to the basis elements of the 2-D transform. Since DC coefficients constitute the main component in the representation of each of the blocks, their thresholding will affect all block estimates.

Smoothing via thresholding in transform domain is inherently resistant (to some extent) against registrations errors. Indeed, if due to erroneous block-matching an unsimilar block appears in the group, the magnitudes of the AC coefficients coding the block's differences from the other blocks will be high. As a result, these coefficients will survive after thresholding, preventing unsimilar block from being fused with the rest of data. Hence, 3-D transform domain collaborative filtering can be used as a robust and flexible data fusion strategy.

In our earlier SR algorithm based on BM3D-modeling, IFSR [DFKE08c], all coefficients were treated equally and the same threshold was applied to all of them. Such approach can be considered suboptimal, since typically SR reconstruction requires much stronger inter-block than intra-block smoothing, and the thresholding levels for AC and DC coefficients should be adjusted accordingly. This specific issue is addressed in the SR algorithm which we present below. We use the framework of BM3D-frames to formulate SR as a GNE problem and derive an algorithm for solving it. Obtained algorithm is similar to IFSR, but unlike the later one, which was introduced as an empirical procedure, for the new algorithm we can point out the corresponding optimization problem.

4.2 SR algorithm with BM3D-modeling

We consider a general form of the video super-resolution problem, where a low-resolution image sequence $\{\mathbf{z}^i\}_{i=1}^k$, $\mathbf{z}^i \in \mathbb{R}^N$, is obtained through observations given by the model

$$\mathbf{z}^i = \mathbf{D}\mathbf{A}\bar{\mathbf{y}}^i + \boldsymbol{\eta}, \quad i = 1, \dots, k, \quad (4.3)$$

$\{\bar{\mathbf{y}}^i\}_{i=1}^k$, $\bar{\mathbf{y}}^i \in \mathbb{R}^{Nd^2}$, $d \in \mathbb{N}$ being the unknown high-resolution (HR) image sequence. The vectors \mathbf{z}^i and $\bar{\mathbf{y}}^i$ represent the pixel values of the corresponding images arranged columnwise. The linear operators (matrices) \mathbf{A} and \mathbf{D} are assumed to be known, representing, respectively, blur and decimation with the factor d . The additive noise $\boldsymbol{\eta}$ is assumed to be i.i.d. Gaussian with mean zero and known standard deviation σ . The problem is to reconstruct the HR sequence $\{\bar{\mathbf{y}}^i\}_{i=1}^k$ from the LR observations $\{\mathbf{z}^i\}_{i=1}^k$.

For finding similar blocks in video sequences we use the same simple block-matching procedure which we used for images with the only difference that now similar blocks are searched not only in the current, but also in the adjacent frames within a certain temporal window. Let $\mathbf{y} \in \mathbb{R}^{kNd^2}$ be a column vector representing the image sequence $\{\mathbf{y}^i\}_{i=1}^k$, such that

$$\mathbf{y} = \left((\mathbf{y}^1)^T, (\mathbf{y}^2)^T, \dots, (\mathbf{y}^k)^T \right)^T.$$

Analogously to the case of *image* BM3D modeling (Section 1.4.1), it is possible to show that for a fixed grouping the *video* BM3D analysis/synthesis operations

can be given in the matrix form linking the image sequence \mathbf{y} and its groupwise spectrum vector $\boldsymbol{\omega} \in \mathbb{R}^M$ by the forward and backward transforms

$$\boldsymbol{\omega} = \boldsymbol{\Phi}\mathbf{y}, \mathbf{y} = \boldsymbol{\Psi}\boldsymbol{\omega}.$$

Statement of the Proposition 1 remains true also for the video BM3D-frames.

Without loss of generality, we can assume that components of $\boldsymbol{\omega}$ are arranged such that

$$\boldsymbol{\omega} = \begin{pmatrix} \boldsymbol{\omega}^{\text{dc}} \\ \boldsymbol{\omega}^{\text{ac}} \end{pmatrix},$$

where $\boldsymbol{\omega}^{\text{dc}}$ and $\boldsymbol{\omega}^{\text{ac}}$ are the column vectors of the DC and AC coefficients respectively. We denote by $\boldsymbol{\Phi}^{\text{dc}}$ and $\boldsymbol{\Phi}^{\text{ac}}$ the submatrices of the analysis matrix $\boldsymbol{\Phi}$ corresponding to calculation of the vectors $\boldsymbol{\omega}^{\text{dc}}$ and $\boldsymbol{\omega}^{\text{ac}}$. Also, we denote by $\boldsymbol{\Psi}_i, i = 1, \dots, k$, the $Nd^2 \times M$ submatrices of $\boldsymbol{\Psi}$ corresponding to the synthesis of the individual frames, such that $\mathbf{y}^i = \boldsymbol{\Psi}^i \boldsymbol{\omega}$.

We assume that the block-matching has been performed and the frame operators $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are already constructed. We formulate SR reconstruction corresponding to the observation model (4.3) as the GNE problem:

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}} L_1(\mathbf{y}, \boldsymbol{\omega}^*) \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}^*, \boldsymbol{\omega}) \end{cases}, \quad (4.4)$$

where

$$L_1(\mathbf{y}, \boldsymbol{\omega}) = \sum_{i=1}^k \left(\|\mathbf{z}^i - \mathbf{D}\mathbf{A}\mathbf{y}^i\|_2^2 + \frac{\gamma}{2} \|\mathbf{y}^i - \boldsymbol{\Psi}^i \boldsymbol{\omega}\|_2^2 \right), \quad (4.5)$$

$$\begin{aligned} L_2(\mathbf{y}, \boldsymbol{\omega}) &= \tau_{\text{dc}} \|\boldsymbol{\omega}^{\text{dc}}\|_p + \frac{1}{2} \|\boldsymbol{\omega}^{\text{dc}} - \boldsymbol{\Phi}^{\text{dc}} \mathbf{y}\|_2^2 + \\ &\tau_{\text{ac}} \|\boldsymbol{\omega}^{\text{ac}}\|_p + \frac{1}{2} \|\boldsymbol{\omega}^{\text{ac}} - \boldsymbol{\Phi}^{\text{ac}} \mathbf{y}\|_2^2. \end{aligned} \quad (4.6)$$

To highlight the algebraic similarity of the functional L_1 with its counterpart from the gaussian deblurring problem we can rewrite (4.5) in the form

$$L_1(\mathbf{y}, \boldsymbol{\omega}) = \left\| \mathbf{z} - \text{diag} \left(\overbrace{\mathbf{D}\mathbf{A}, \dots, \mathbf{D}\mathbf{A}}^{k\text{-times}} \right) \mathbf{y} \right\|_2^2 + \frac{\gamma}{2} \|\mathbf{y} - \boldsymbol{\Psi}\boldsymbol{\omega}\|_2^2, \quad (4.7)$$

where $\mathbf{z} = \left((\mathbf{z}^1)^T, (\mathbf{z}^2)^T, \dots, (\mathbf{z}^k)^T \right)^T$. The only difference is that unlike the deblurring problem, where all spectral coefficients were treated equally, we penalize complexity of the AC and DC spectral coefficients at different levels, using separate regularization parameters, which let us to control the intra- and inter-block smoothing independently.

It can be shown, that the alternating minimization procedure (3.4) can still be used to find a fix-point solution of the problem (4.4)-(4.6). The corresponding proof, with marginal changes, repeats the proof of Proposition 3 from [DKE12].

Derivation of the actual algorithm is based on the observation that minimization of L_1 is separable with respect to the variables $\mathbf{y}^i, i = 1, \dots, k$, and the minimization of L_2 is separable with respect to the vectors $\boldsymbol{\omega}^{\text{dc}}$ and $\boldsymbol{\omega}^{\text{ac}}$. Not to repeat the same type of calculations as we did in the previous chapters, we skip the step-by-step derivation and directly present the resulting Algorithm 5.

Algorithm 5 Solution of the problem (4.4)-(4.6)

```

1: input:  $\{\mathbf{z}^i\}_{i=1}^k, \mathbf{A}, \mathbf{D}, \Phi, \Psi, \{\mathbf{y}_{\text{init}}^i\}_{i=1}^k, \tau_{\text{dc}}, \tau_{\text{ac}}, \gamma$ 
2: initialization:
3: set:  $t = 0, \mathbf{y}_0 = \mathbf{y}_{\text{init}}, \boldsymbol{\omega}_0 = \Phi \mathbf{y}_{\text{init}}$ 
4: repeat
5:    $t = t + 1$ 
6:    $\mathbf{y}_t^i = (\mathbf{A}^T \mathbf{D}^T \mathbf{D} \mathbf{A} + \gamma \mathbf{I})^{-1} (\mathbf{A}^T \mathbf{D}^T \mathbf{z}^i + \gamma \Psi^i \boldsymbol{\omega}_{t-1}), i = 1, \dots, k$ 
7:    $\boldsymbol{\omega}_t^{\text{dc}} = Th_{\tau_{\text{dc}}}(\Phi^{\text{dc}} \mathbf{y}_t)$ 
8:    $\boldsymbol{\omega}_t^{\text{ac}} = Th_{\tau_{\text{ac}}}(\Phi^{\text{ac}} \mathbf{y}_t)$ 
9:    $\boldsymbol{\omega}_t = \begin{pmatrix} \boldsymbol{\omega}_t^{\text{dc}} \\ \boldsymbol{\omega}_t^{\text{ac}} \end{pmatrix}$ 
10: until convergence
11: output:  $\hat{\mathbf{y}} = \Psi \boldsymbol{\omega}_t$ 

```

The above derived algorithm can achieve resolution enhancement only if the provided block-matching information accurately captures motion in the frames. To obtain such accuracy, block-matching needs to be applied to some good estimates of the HR frames, which are obviously not available at the beginning of the reconstruction. We solve this "chicken-egg" problem iteratively. We start with an initial estimate of the HR sequence \mathbf{y}_0 (it can be obtained through bilinear or bicubic interpolation or by calculating pseudoinverse) and do block-matching on \mathbf{y}_0 . Obtained registration, although not very accurate, can be used to improve estimates of the HR images. So we construct operators Φ and Ψ and run Algorithm 5. After the new estimates of the HR images are obtained, we repeat the block-matching procedure applying it to the newly obtained estimates. The improved block-matching information is then used to update the frame operators and formulate new GNE problem, which is again solved with Algorithm 5. This process can be repeated several (typically 3-10) times, until we reach steady-state when neither HR estimates nor the block-matching can be further improved. Described steps are summarized as Algorithm 6.

4.3 Implementation

4.3.1 Initialization

For its initialization, Algorithm 6 requires rough estimates of the high-resolution (HR) images which are used for performing the first block-matching. In other words, these rough image estimates are necessary to obtain the initial location estimates of the mutually similar blocks with respect to the HR coordinates. The SR algorithm puts very mild requirements on the quality of the initialization images:

Algorithm 6 The SR-BM3D algorithm

- 1: **input:** $\{\mathbf{z}^i\}_{i=1}^k, \mathbf{A}, \mathbf{D}, \tau_{dc}, \tau_{ac}, \gamma$
 - 2: **initialization:**
 - 3: *set:* $m = 0$
 - 4: *set:* $\mathbf{y}_0^i = \text{Bicubic_interpolation}(\mathbf{z}^i), i = 1, \dots, k$
 - 5: **repeat**
 - 6: $m = m + 1$
 - 7: *perform block-matching:*
 - if** $m == 1$
 - do it on a smoothed copy of \mathbf{y}_0 ,*
 - otherwise**
 - do it on the current estimate \mathbf{y}_{m-1} .*
 - 8: *update operators Φ and Ψ*
 - 9: *formulate new GNE problem corresponding to the updated frames and solve it with Algorithm 5 using \mathbf{y}_{m-1} as the initialization and $\tau_{dc}, \tau_{ac}, \gamma$ as regularization parameters; assign the found solution to \mathbf{y}_m*
 - 10: **until** convergence
 - 11: **output:** $\hat{\mathbf{y}} = \mathbf{y}_m$
-

in practice, it suffices that the initialization images feature all coarsest structures in the scene while being free of structured artefacts, such as aliasing, ringing, stair-casing, or blockiness. Presence of structured artefacts affects the accuracy of the block-matching, leading to the erroneous motion estimation and reproduction of artefacts in the reconstructed images. Suitable images for initializing the SR algorithm can be obtained, for example, by first upsampling the input low-resolution (LR) images using bicubic interpolation and then blurring the upsampled images with a low-pass Gaussian filter to suppress structured artefacts.

4.3.2 Computing the inverse

Implementation of Algorithm 5 possesses a technical problem, namely, in step 6 of the algorithm it is required to solve large system of linear equations:

$$\mathbf{y}_t^i = (\mathbf{A}^T \mathbf{D}^T \mathbf{D} \mathbf{A} + \gamma \mathbf{I})^{-1} (\mathbf{A}^T \mathbf{D}^T \mathbf{z}^i + \gamma \Psi^i \boldsymbol{\omega}_{t-1}). \quad (4.8)$$

In the general case this system can be solved iteratively using one of the standard methods from numerical algebra, for example, the conjugate gradient (CG) method. However, due to the high dimensionality of the problem, convergence of the iterative process can be very slow, making implementation impractical.

It should be mentioned that problem of computing inverse or conjugate of operator $\mathbf{D} \mathbf{A}$ is common to all SR algorithms. There are two approaches, which allow to obtain relatively fast implementations though producing approximate and often suboptimal solutions.

The first approach is based on substituting the true degradation model with an approximate one for which it is possible to find an efficient algorithm for computing (4.8) exploiting the particular structures of the matrices \mathbf{A} and \mathbf{D} . We mention few such models:

- a) no blur ($\mathbf{A} = \mathbf{I}$);
- b) blurring $d \times d$ size uniform PSF followed by decimation with factor d ;
- c) LR images coincide (up to the scaling factor) with the LL band of the wavelet decomposition of the corresponding HR images; here the low-pass wavelet filter plays the role of the blurring operator;
- d) LR images are obtained by ideal low-pass filtering HR images with *sinc* function, followed by decimation; in this case the DFT spectrum of the LR image coincides, up to the scaling factor, with the low-frequency part of the spectrum of the corresponding HR image.

More models together with detailed discussion of this approach can be found in [DFKE08c] and [DFKE10]. In Figures 4.2-4.3 we show examples of HR sequences reconstructed with Algorithm 5 from the data with unknown true PSF using degradation models a) and b) respectively. We can see, that thanks to the robustness of the algorithm, we are able to reveal new details in the reconstructed image, even using imprecise and very approximate degradation model.

The second approach, is based on the idea of splitting SR in two problems: estimation of the blurry HR image $\mathbf{A}\bar{\mathbf{y}}$ and followed by deblurring in order to obtain an estimate of $\bar{\mathbf{y}}$. In other words, instead of inverting degradation operator \mathbf{AD} at once, we first invert \mathbf{D} and then \mathbf{A} . The split and joint (original) SR problems generally are not equivalent. One particular example when such equivalence exist is considered in [EHO01]. Nevertheless, many algorithms [FREM04b], [PETM09], [TMPE09] ignore the equivalence issues and use the splitting to obtain approximate solutions.

4.4 Discussion

4.4.1 Reconstruction from noise-free data

The SR algorithm for the noise-free data can be obtained by taking the regularization parameter γ to 0, in which case formula (4.8) reduces to the projection on the subspace defined by the observation \mathbf{z}^i :

$$\mathbf{y}_t^i = P_O(\Psi^i \boldsymbol{\omega}_{t-1}), \quad O = \{\mathbf{x} \in \mathbb{R}^{Nd^2} : \mathbf{DA}\mathbf{x} = \mathbf{z}^i\}. \quad (4.9)$$

Here P is the orthogonal projection operator. Since arbitrary element \mathbf{x} from O can be represented as the sum of its projection on the kernel of operator \mathbf{DA} and the vector $(\mathbf{A}^T \mathbf{D}^T \mathbf{DA})^{-1} (\mathbf{DA})^T \mathbf{z}^i$, we can write (4.9) as

$$\begin{aligned} \mathbf{y}_t^i &= (\mathbf{A}^T \mathbf{D}^T \mathbf{DA})^{-1} (\mathbf{DA})^T \mathbf{z}^i + P_{\ker(\mathbf{DA})}(\Psi^i \boldsymbol{\omega}_{t-1}), \\ \ker(\mathbf{DA}) &= \left\{ \mathbf{x} \in \mathbb{R}^{Nd^2} : \mathbf{DA}\mathbf{x} = \mathbf{0} \right\}. \end{aligned}$$

Finally, using the identity $P_{\ker(\mathbf{DA})}(\mathbf{x}) = \mathbf{x} - (\mathbf{A}^T \mathbf{D}^T \mathbf{DA})^{-1} (\mathbf{DA})^T \mathbf{DA}\mathbf{x}$ we obtain the equivalent of formula (4.8) for noise-free data

$$\mathbf{y}_t^i = P_O(\Psi^i \boldsymbol{\omega}_{t-1}) = \Psi^i \boldsymbol{\omega}_{t-1} + (\mathbf{A}^T \mathbf{D}^T \mathbf{DA})^{-1} (\mathbf{DA})^T (\mathbf{z} - \mathbf{DA}\Psi^i \boldsymbol{\omega}_{t-1}).$$



Figure 4.2: SR reconstruction from low-resolution compressed video taken with webcam. The sequence contains in total 53 frames from which number 10, 42 and 47 are shown. From top to down: nearest-neighbour interpolation, bicubic interpolation, SR reconstruction with Algorithm 6. The images are 3 times larger the size of the LR images. The true degradation model is unknown; the model assumed by the reconstruction algorithm is decimation with factor 3, no blur ($\mathbf{A} = \mathbf{I}$).

Used low-resolution test sequence is a part of "MDSP Super-Resolution And Demosaicing Dataset" which is publicly available at <http://www.soe.ucsc.edu/milanfar/software/sr-datasets.html>



Figure 4.3: SR reconstruction from low-resolution video taken with infrared camera. The sequence contains in total 64 frames from which number 5, 30 and 60 are shown. From top to down: nearest-neighbour interpolation, bicubic interpolation, SR reconstruction with Algorithm 6. The images are 3 times larger the size of the LR images. The true degradation model is unknown; the model assumed by the reconstruction algorithm is 3×3 uniform blur followed by decimation with factor 3.

Used low-resolution test sequence is a part of "MDSP Super-Resolution And Demosaicing Dataset" which is publicly available at <http://www.soe.ucsc.edu/milanfar/software/sr-datasets.html>

Note, that even in the absence of noise, the two other regularization parameters: τ_{dc} and τ_{ac} , need to be non-zero in order to enable fusion of the data.

4.4.2 Reconstruction from noisy data

The noisy data can be handled either directly, by adjusting the regularization parameters in Algorithm 5, or by first prefiltering LR sequence with some denoising algorithm and then applying Algorithm 5 to the noise-free data. Which of these two approaches is more efficient depends on the level of the noise. If noise is very low ($\sigma \leq 3$, assuming $0 \dots 255$ image intensity range) the direct approach typically demonstrates slightly better results than those obtained with prefiltering, since prefiltering inevitably removes some of the high-frequency information which otherwise is used in the direct approach. For higher noise levels situation changes, results of the direct approach become inferior to the prefiltering approach. The reason is that the noise component of the estimate \mathbf{y}_t^i is not AWGN, it is given by the expression

$$(\mathbf{A}^T \mathbf{D}^T \mathbf{D} \mathbf{A} + \gamma \mathbf{I})^{-1} \mathbf{A}^T \mathbf{D}^T \boldsymbol{\eta}^i.$$

The linear transformation introduces strong spatial correlation and the noise becomes structured which hardens its separation from signal. Suppression of such noise requires stronger regularization, which drops the efficacy of the SR reconstruction.

In Fig 4.4 we show comparison of the results obtained with both reconstruction approaches for synthetic LR sequence corrupted by AWGN with $\sigma = 5$. The prefiltering approach uses VBM3D for denoising LR sequence. We can see that prefiltering approach demonstrates slightly better results both numerically and visually.

4.4.3 Relation to the IFSR algorithm

Ignoring minor differences, the IFSR algorithm [DFKE08c] can be viewed as a variation of Algorithm 6 for noise-free data, where regularization parameters $\tau_{dc} = \tau_{ac} = \tau(m)$ depend on m , so that $\tau(m) > \tau(m+1) > 0$. Such scheme with gradually decreasing regularization renders coarse-to-fine reconstruction: coarse scale image features are reconstructed first using strong regularization, and finer features are reconstructed in the subsequent iterations using weaker regularization.

The necessity of lowering regularization parameter through iterations in IFSR was never questioned. In fact, in the above experiments with Algorithm 6 we saw that successful reconstruction can be obtained without changing regularization parameters. However, taking into account our discussion in Section 4.1 about the role of AC and DC coefficients in data fusion, we can now explain why it is compulsory to reduce the regularization in IFSR.

Recall, that data fusion is performed by thresholding AC coefficients, while thresholding DC coefficients serves only for suppression of possible fusion artefacts. The thresholding level required to enable fusion is typically much higher than the one needed for concealment of artefacts. For example in our experiments with Algorithm 6, the threshold used for AC coefficients is 3-10 times higher than the threshold for DC coefficients. As we already mentioned, IFSR do not differentiate

between AC and DC coefficients; at the fusion step they all are thresholded using the same threshold. To ensure fusion IFSR needs to start with high thresholding level, but since the same strong thresholding is also applied to the DC coefficients the resulting images are getting significantly smoothed. To recover the fine details, in subsequent iterations the threshold needs to be gradually lowered to the level at which it does not introduce unnecessary spatial smoothing.

Despite mentioned differences, our experiments show that both IFSR and Algorithm 6 produce results of very similar quality and it is difficult to give preference to one of them.

For practical applications, where complexity of the algorithm is critical, a hybrid approach should be considered where the AC and DC threshold parameters are controlled independently and both are gradually reduced through iterations. Such scheme allows to speed up the convergence (and hence reduce complexity of the algorithm) by applying stronger regularization at early stages of the reconstruction.



Figure 4.4: Comparison of two reconstruction approaches: directly from noisy and from prefiltered data. The synthetic LR sequence is obtained from the original HR sequence *Suzie* by blurring frames with 3×3 uniform PSF, decimating with factor 3 and adding AWGN with $\sigma = 5$. The test sequence contains 30 frames; frame number 21 is shown. Top row: nearest-neighbour interpolation of noisy frame, nearest-neighbour interpolation of prefiltered frame. Middle row: reconstruction with Algorithm 6 directly from noisy data ($\gamma = 0.3, \tau_{dc} = 91.1, \tau_{ac} = 1053$, mean PSNR over 30 frames 32.4 dB), reconstruction with Algorithm 6 from prefiltered data ($\tau_{dc} = 14.6, \tau_{ac} = 1053$, mean PSNR over 30 frames 33.1 dB). Last row: differences between the reconstructed and ground truth images.

Conclusions to the thesis

C.1 Overview

The contribution of this thesis concerns three main aspects of image and video reconstruction: image modeling, variational problem formulation, and algorithm design.

First, we formalize the BM3D-modeling in terms of the overcomplete sparse frame representation. We construct analysis and synthesis BM3D-frames and study their properties, making BM3D-modeling available for use in variational formulations of image reconstruction problems.

Second, we demonstrate that standard problem formulations based on single objective optimization, such as Basis Pursuit Denoising and its various extensions, cannot be used with the imaging models generating non-tight frames such as BM3D. We propose an alternative sparsity promoting formulation based on generalized Nash equilibrium (GNE).

Third, using BM3D-frames we develop practical algorithms for image deblurring and super-resolution problems. To the best of our knowledge, these algorithms provide results which are the state of the art in the field.

C.2 Future research

Through this thesis we were dealing only with the standard BM3D-modeling. Using more elaborated versions of the 3-D transform domain modeling, such as BM3D with shape-adaptive supports [DFKE08b] and shape-adaptive BM3D with principle component analysis [DFKE09], could potentially improve the performance of the developed reconstruction algorithms.

Speaking about possible directions to improve the 3-D transform modeling itself, the most promising seems to be utilizing it in a multi-scale processing setup. Advantages of such approach has been demonstrated in [BH11], where authors developed mutli-scale meta-procedure, which applies existing denoising algorithms across different scales and combines the resulting images into a single denoised image. Operating at different scales allows to provide regularization not only to fine image details, but also to large-scale features, addressing one of the weak points of the current BM3D-modeling - poor performance at reconstructing low frequency content is the image.

Our closest targets are the development of blind image deblurring algorithm

based on the proposed non-blind deblurring method, and considering applications of the developed techniques in the inverse problems arising in the field of digital holography.

Bibliography

- [ABDF10] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, “Fast image recovery using variable splitting and constrained optimization,” *Image Processing, IEEE Transactions on*, vol. 19, no. 9, pp. 2345–2356, sept. 2010.
- [ABDF11] —, “An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems,” *Image Processing, IEEE Transactions on*, vol. 20, no. 3, pp. 681–695, march 2011.
- [AEB06] M. Aharon, M. Elad, and A. Bruckstein, “K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, nov. 2006.
- [Ans48] F. J. Anscombe, “The transformation of poisson, binomial and negative-binomial data,” *Biometrika*, vol. 35, no. 3/4, pp. pp. 246–254, 1948. [Online]. Available: <http://www.jstor.org/stable/2332343>
- [BB98] P. Boccacci and M. Bertero, *Introduction to Inverse Problems in Imaging*. Taylor & Francis, 1998.
- [BCM05] A. Buades, B. Coll, and J. M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [BD08] T. Blumensath and M. Davies, “Iterative thresholding for sparse approximations,” *Journal of Fourier Analysis and Applications*, vol. 14, pp. 629–654, 2008.
- [BD09] —, “Iterative hard thresholding for compressed sensing,” *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.
- [Ber96] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

- [BH11] H. Burger and S. Harmeling, “Improving denoising algorithms via a multi-scale meta-procedure,” in *Pattern Recognition*, ser. Lecture Notes in Computer Science, R. Mester and M. Felsberg, Eds. Springer Berlin Heidelberg, 2011, vol. 6835, pp. 206–215. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23123-0_21
- [CBFZ07] C. Chaux, L. Blanc-Féraud, and J. Zerubia, “Wavelet-based restoration methods: application to 3D confocal microscopy images,” in *Wavelets XII*, vol. 6701. San Diego, CA, USA: SPIE, sep. 2007, pp. 67 010E–11.
- [CDS01] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.
- [CE07] P. Campisi and K. Egiazarian, *Blind image deconvolution: theory and applications*. CRC, 2007.
- [CGMK10] G. Chantas, N. Galatsanos, R. Molina, and A. Katsaggelos, “Variational bayesian image restoration with a product of spatially weighted total variation image priors,” *Image Processing, IEEE Transactions on*, vol. 19, no. 2, pp. 351–362, feb. 2010.
- [Chr03] O. Christensen, *An introduction to frames and Riesz bases*. Birkhäuser, 2003.
- [CW05] P. L. Combettes and V. R. Wajs, “Signal recovery by proximal forward-backward splitting,” *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.
- [Dab10] K. Dabov, “Image and video restoration with nonlocal transform-domain filtering,” Ph.D. dissertation, Tampere University of Technology, September 2010.
- [DBFZ⁺06] N. Dey, L. Blanc-Féraud, C. Zimmer, P. Roux, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, “Richardson-lucy algorithm with total variation regularization for 3d confocal microscope deconvolution,” *Microscopy Research and Technique*, vol. 69, no. 4, pp. 260–266, 2006. [Online]. Available: <http://dx.doi.org/10.1002/jemt.20294>
- [DDDM04] I. Daubechies, M. Defrise, and C. De Mol, “An iterative thresholding algorithm for linear inverse problems with a sparsity constraint,” *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004. [Online]. Available: <http://dx.doi.org/10.1002/cpa.20042>
- [DFKE07] 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, pp. 2080–2095, Aug. 2007.

- [DFKE08a] —, “Image restoration by sparse 3d transform-domain collaborative filtering,” in *SPIE Electronic Imaging '08*, vol. 6812, San Jose, California, USA, Jan. 2008.
- [DFKE08b] —, “A nonlocal and shape-adaptive transform-domain collaborative filtering,” in *Proc. Local and Nonlocal Approx. in Image Process.*, Lausanne, Switzerland, Sep. 2008.
- [DFKE08c] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, “Image and video super-resolution via spatially adaptive block-matching filtering,” in *Proc. Int. Workshop Local Non-Local Approx. Image Process.*, Lausanne, Switzerland, Aug. 2008, pp. 53–60.
- [DFKE08d] —, “Image upsampling via spatially adaptive block-matching filtering,” in *Proc. European Signal Process. Conf.*, Lausanne, Switzerland, Aug. 2008.
- [DFKE09] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Bm3d image denoising with shape-adaptive principal component analysis,” in *Workshop on Sig. Process. with Adaptive Sparse Structured Representations, SPARS 09*, Saint-Malo, France, Apr. 2009.
- [DFKE10] A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian, “Spatially adaptive filtering as regularization in inverse imaging: compressive sensing, super-resolution, and upsampling,” in *Super-Resolution Imaging*, P. Milanfar, Ed. CRC Press, 2010, pp. 123–153.
- [DFKE11] —, “Deblurring of poissonian images using BM3D frames,” in *Wavelets and Sparsity XIV*, vol. 8138, no. 1. SPIE, Sep. 2011, pp. 813 812–813 812–7.
- [DFS09] F.-X. Dupe, J. Fadili, and J.-L. Starck, “A proximal iteration for deconvolving poisson noisy images using sparse representations,” *Image Processing, IEEE Transactions on*, vol. 18, no. 2, pp. 310–321, feb. 2009.
- [DKE12] A. Danielyan, V. Katkovnik, and K. Egiazarian, “BM3D frames and variational image deblurring,” *Image Processing, IEEE Transactions on*, vol. 21, no. 4, pp. 1715–1728, april 2012.
- [DTDS12] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, “Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit,” *Information Theory, IEEE Transactions on*, vol. 58, no. 2, pp. 1094–1121, feb. 2012.
- [EAH00] K. Engan, S. O. Aase, and J. H. HusÅy, “Multi-frame compression: theory and design,” *Signal Processing*, vol. 80, no. 10, pp. 2121–2140, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168400000724>

- [EB92] J. Eckstein and D. Bertsekas, “On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Mathematical Programming*, vol. 55, pp. 293–318, June 1992.
- [EFK07] K. Egiazarian, A. Foi, and V. Katkovnik, “Compressed sensing image reconstruction via recursive spatially adaptive filtering,” in *Proc. IEEE Int. Conf. on Image Process., ICIP 2007*, Sept. 2007, pp. 549–552.
- [EHO01] M. Elad and Y. Hel-Or, “A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur,” *Image Processing, IEEE Transactions on*, vol. 10, no. 8, pp. 1187–1193, aug 2001.
- [EMR07] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors,” *Inverse Problems*, vol. 23, no. 3, p. 947, 2007.
- [EV08] M. Ebrahimi and E. R. Vrscay, “Multi-frame super-resolution with no explicit motion estimation,” in *Proc. Computer Vision and Pattern Recognition*, Las Vegas, Nevada, USA, July 2008, pp. 455–459.
- [FAT⁺05] A. Foi, S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, “A spatially adaptive Poissonian image deblurring,” in *Proc. IEEE Int. Conf. Image Process.*, Genova, Italy, Sep. 2005, pp. 925–928.
- [FATK06] A. Foi, S. Alenius, M. Trimeche, and V. Katkovnik, “Adaptive-size block transforms for poissonian image deblurring,” in *Proc. Int. TICSP Workshop Spectral Methods Multirate Signal Process.*, Florence, Italy, Sep. 2006.
- [FBD10] M. Figueiredo and J. Bioucas-Dias, “Restoration of poissonian images using alternating direction optimization,” *Image Processing, IEEE Transactions on*, vol. 19, no. 12, pp. 3133–3145, dec. 2010.
- [FBDN07] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, “Majorization-minimization algorithms for wavelet-based image restoration,” *Image Processing, IEEE Transactions on*, vol. 16, no. 12, pp. 2980–2991, dec. 2007.
- [FDKE06] A. Foi, K. Dabov, V. Katkovnik, and K. Egiazarian, “Shape-Adaptive DCT for denoising and image reconstruction,” in *Proc. SPIE Electronic Imaging: Algorithms and Systems V*, vol. 6064A-18, San Jose, CA, USA, Jan. 2006.
- [FK10] F. Facchinei and C. Kanzow, “Generalized nash equilibrium problems,” *Annals of Operations Research*, vol. 175, pp. 177–211, 2010. [Online]. Available: <http://dx.doi.org/10.1007/s10479-009-0653-x>

- [FKE07] 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, pp. 1395–1411, May 2007.
- [FN03] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.
- [FN05] M. A. T. Figueiredo and R. Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Genova, Italy, Sep. 2005, pp. 782–785.
- [Foi05] A. Foi, "Anisotropic nonparametric image processing: theory, algorithms and applications," Ph.D. dissertation, Politecnico di Milano, Apr. 2005.
- [Foi09] —, "Clipped noisy images: Heteroskedastic modeling and practical denoising," *Signal Processing*, vol. 89, no. 12, pp. 2609–2629, December 2009,
<http://dx.doi.org/10.1016/j.sigpro.2009.04.035> .
- [FREM04a] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, pp. 47–57, 2004.
- [FREM04b] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *Image Processing, IEEE Transactions on*, vol. 13, no. 10, pp. 1327–1344, oct. 2004.
- [GCMP08] J. Guerrero-Colon, L. Mancera, and J. Portilla, "Image restoration using space-variant Gaussian scale mixtures in overcomplete pyramids," *Image Processing, IEEE Transactions on*, vol. 17, no. 1, pp. 27–41, jan. 2008.
- [GRPMSM11] E. Gil-Rodrigo, J. Portilla, D. Miraut, and R. Suarez-Mesa, "Efficient joint poisson-gauss restoration using multi-frame l2-relaxed-l0 analysis-based sparsity," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 1385–1388.
- [KEA05] V. Katkovnik, K. Egiazarian, and J. Astola, "A spatially adaptive nonparametric regression image deblurring," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1469–1478, Oct. 2005.
- [KFEA10] 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, Jan. 2010.
- [Kin01] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234–253, 2001.

- [LBS08] K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory: A Concise Multidisciplinary Introduction*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2008, vol. 2, no. 1.
- [LLP06] H. Q. Luong, A. Ledda, and W. Philips, “An image interpolation scheme for repetitive structures,” in *Int. Conf. Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, vol. 4141. Springer, Sep. 2006, pp. 104–115.
- [LOct] X. Li and M. Orchard, “New edge-directed interpolation,” *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1521–1527, Oct.
- [MF11] M. Makitalo and A. Foi, “Optimal inversion of the anscombe transformation in low-count poisson image denoising,” *Image Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 99–109, jan. 2011.
- [MS06] F. Murtagh and J. Starck, *Astronomical image and data analysis*. Springer, 2006.
- [MZ93] S. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, dec 1993.
- [NCB04] R. Neelamani, H. Choi, and R. Baraniuk, “Forward: Fourier-wavelet regularized deconvolution for ill-conditioned systems,” *Signal Processing, IEEE Transactions on*, vol. 52, no. 2, pp. 418–433, feb. 2004.
- [NLD07] M. N., Y. Lu, and M. Do, “Image interpolation using multiscale geometric representations,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6498, Feb. 2007.
- [NW06] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.
- [OBDF09] J. Oliveira, J. M. Bioucas-Dias, and M. A. Figueiredo, “Adaptive total variation image deblurring: A majorization-minimization approach,” *Signal Processing*, vol. 89, no. 9, pp. 1683–1693, 2009.
- [PETM09] M. Protter, M. Elad, H. Takeda, and P. Milanfar, “Generalizing the nonlocal-means to super-resolution reconstruction,” *Image Processing, IEEE Transactions on*, vol. 18, no. 1, pp. 36–51, Jan. 2009.
- [Por09] J. Portilla, “Image restoration through l0 analysis-based sparse optimization in tight frames,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*, Cairo, Egypt, Nov. 2009, pp. 3909–3912.

- [PRRK93] Y. C. Pati, R. Rezaifar, Y. C. P. R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, nov 1993, pp. 40–44.
- [PSWS03] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *Image Processing, IEEE Transactions on*, vol. 12, no. 11, pp. 1338 – 1351, nov. 2003.
- [ROF92] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, November 1992.
- [TA77] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*. Wiley, 1977.
- [TFM07] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 2, February 2007.
- [TH84] R. Y. Tsai and T. S. Huang, "Multi-frame image restoration and registration," in *Advances in Computer Vision and Image Processing*, P. Milanfar, Ed. Greenwich, CT: JAI Press Inc., 1984, p. 317–339.
- [Tib96] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B*, vol. 58, pp. 267–288, 1996.
- [TMPE09] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *Image Processing, IEEE Transactions on*, vol. 18, no. 9, pp. 1958–1975, Sept. 2009.
- [Tro06] J. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," *Information Theory, IEEE Transactions on*, vol. 52, no. 3, pp. 1030–1051, 2006.
- [TW09] X.-C. Tai and C. Wu, "Augmented Lagrangian method, dual methods and split Bregman iteration for rof model," in *Scale Space and Variational Methods in Computer Vision*, ser. Lecture Notes in Computer Science, X.-C. Tai, K. Mikolajczyk, M. Lysaker, and K.-A. Lie, Eds. Springer, 2009, vol. 5567, pp. 502–513.

Publications

Publication[DKE12]

A. Danielyan, V. Katkovnik, and K. Egiazarian,
“BM3D frames and variational image deblurring,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1715–1728, April 2012.

© 2012 IEEE. Reprinted with permission.

Publication[DFKE11]

A. Danielyan, V. Katkovnik, and K. Egiazarian,
“Deblurring of Poissonian images using BM3D frames,”
Proc. Wavelets and Sparsity XIV, SPIE, vol. 8138, San
Diego, September 2011.

© 2011 SPIE. Reprinted with permission.

Deblurring of Poissonian images using BM3D frames

Aram Danielyan, Vladimir Katkovnik and Karen Egiazarian

Department of Signal Processing, Tampere University of Technology, Tampere, Finland

ABSTRACT

We propose a novel deblurring algorithm for Poissonian images. The algorithm uses data adaptive BM3D-frames for sparse image modeling. Reconstruction is formulated as a generalized Nash equilibrium problem, seeking a balance between the data fit and the complexity of the solution. Simulated experiments demonstrate numerical and visual superiority of the proposed algorithm over the current state-of-the-art methods.

Keywords: image reconstruction, sparsity, generalized Nash equilibrium, non-tight frames

1. INTRODUCTION

During the last years variational image restoration with sparsity promoting regularizers attracted much attention. Various sparse image modelling techniques were proposed: from the Total Variation (TV) and wavelets to the different overcomplete frames and learned dictionaries. In our recent paper¹ we demonstrated that the highly efficient sparse image modeling underlying the BM3D denoising filter² can be formalized in terms of frames. This formalization allows to use the BM3D modeling within the variational reconstruction framework. Nevertheless, there is a peculiarity in way how the regularization based on BM3D-frames should be implied. Due to their structure the BM3D-frames are non-tight. As it was demonstrated,¹ the classical formulation of the reconstruction problem as an optimization of a single objective function with the complexity penalty may lead to the degradation of the restoration quality due to the influence of the ill-conditioned matrices reflecting the inner structure of the non-tight frame. To avoid such an influence, it has been suggested¹ to formulate the reconstruction problem as a generalized Nash equilibrium (GNE) problem. The equilibrium of the GNE problem provides a balance between the data fit and the complexity of the solution.

The deblurring algorithm in¹ was developed for the additive white Gaussian noise model. In this paper we demonstrate how the technique from¹ can be extended to the deblurring of Poissonian data.

There are two main approaches to deal with the Poisson noise. First one is the variance stabilization approach, where observed data undergoes through a nonlinear transformation resulting in a homoscedastic data, which then can be treated with restoration algorithms designed for the homoscedastic noise.³ The second approach is to take into account the Poissonianity of the noise distribution directly in the reconstruction,^{4, 5, 6} We follow the later approach and formulate the reconstruction as a generalized Nash equilibrium problem, where fidelity criterion is devised from the Poissonian likelihood function.

Simulated experiments demonstrate a numerical and visual superiority of the proposed algorithm over the current state-of-the-art methods confirming the advantage of the BM3D-frames as an image modeling tool.

2. BM3D FRAMES

BM3D is a nonlocal image modelling technique based on adaptive high order groupwise models. This technique is well known for its ability to provide highly sparse and redundant image representations. The detailed discussion of the BM3D modelling can be found in.⁷ Here we briefly recall the concepts of the BM3D modeling. It involves two steps: analysis and synthesis.

Analysis. Similar image blocks are collected in groups. Blocks in each group are stacked together to form 3-D data arrays, which are decorrelated using an invertible 3D transform.

The blocking imposes a localization of the image on small pieces where simpler models may fit the observations. It has been demonstrated that a higher sparsity of the signal representation and a lower complexity of the

Corresponding author: Aram Danielyan (e-mail: aram.danielyan(at)tut.fi).

model can be achieved using joint 3D groupwise instead of 2D blockwise transforms. This joint 3D transform dramatically improves the sparsity and effectiveness of image spectrum approximation. The total number of groupwise spectrum elements is much larger than the image size. Thus we arrive to an *overcomplete* or *redundant* data representation.

Synthesis. The groupwise spectra are inverted, providing estimates for each block in the group. These blockwise estimates are returned to their original positions, and the final image reconstruction is calculated as a weighted average of all the obtained blockwise estimates.

Due to the overcompleteness of the spectral representation, at the synthesis step for each pixel we typically obtain multiple estimates. This redundancy is important for effectiveness of the BM3D modeling since it significantly improves the of the final estimate.

For a fixed grouping the BM3D analysis/synthesis operations can be given in the matrix form linking the image $\mathbf{y} \in R^N$ and its groupwise spectrum vector $\boldsymbol{\omega} \in R^M$ by the forward and backward transforms¹

$$\boldsymbol{\omega} = \boldsymbol{\Phi} \cdot \mathbf{y}, \mathbf{y} = \boldsymbol{\Psi} \cdot \boldsymbol{\omega}. \quad (1)$$

The actual form of the analysis and synthesis matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ can found in.¹ Here we present only their main properties.

Proposition 1. The matrices $\boldsymbol{\Phi}^T \boldsymbol{\Phi}$ and $\boldsymbol{\Psi} \boldsymbol{\Psi}^T$ are diagonal with positive items and $\boldsymbol{\Psi} \boldsymbol{\Phi} = \mathbf{I}_{N \times N}$.

The last formula ensures perfect reconstruction of the image \mathbf{y} from the groupwise spectrum $\boldsymbol{\omega}$. It follows from the proposition that $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}^T$ are full column rank ($M \times N$) matrices. Further, the rows of $\boldsymbol{\Phi}$ constitute a frame $\{\phi_n\}$ in R^N , and the columns of $\boldsymbol{\Psi}$ constitute a dual frame $\{\psi_n\}$. These frames are not tight, i.e. $\boldsymbol{\Phi}^T \cdot \boldsymbol{\Phi} \neq \alpha \mathbf{I}_{N \times N}$ and $\boldsymbol{\Psi}^T \cdot \boldsymbol{\Psi} \neq \alpha \mathbf{I}_{N \times N}, \forall \alpha > 0$. In general, $\boldsymbol{\Psi} \neq (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T$ and $\{\psi_n\}$ is an *alternative dual* frame due to the presence of the group weights. The equality $\boldsymbol{\Psi} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T$ takes place only when all group weights are equal to each other. $\{\psi_n\}$ then becomes the *canonical dual* frame.

Since matrices $\boldsymbol{\Phi}, \boldsymbol{\Phi}^T, \boldsymbol{\Psi}$ and $\boldsymbol{\Psi}^T$ are based on groupwise separable 3-D transforms (possibly with some averaging) the multiplications against these matrices can be implemented efficiently. The find the similar blocks and build the groups the block matching (grouping) procedure from² is used.

We want to emphasize that the BM3D-frames are data adaptive, which make them quite different from the other popular frames used for image modeling.

3. PROBLEM STATEMENT

Let $\mathbf{y}, \mathbf{z} \in \mathbb{N}_0^N$ ($\mathbb{N}_0 = \mathbb{N} \cup \{0\}$) represent, respectively, the true and observed images arranged in the columnwise order. It is assumed that \mathbf{z} is a sample of a random vector $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_N\}$ of N independent Poissonian variables with the joint probability distribution

$$P(\mathbf{Z} = \mathbf{z} | \boldsymbol{\lambda}) = \prod_{i=1}^N \frac{\lambda_i^{z_i} e^{-\lambda_i}}{z_i!}, \quad (2)$$

where \mathbf{A} is an $N \times N$ matrix representing blur operator and $\boldsymbol{\lambda} = \mathbf{A} \mathbf{y}$, $\boldsymbol{\lambda} \in \mathbb{R}_+^N$, ($\mathbb{R}_+ = \{x : x \in \mathbb{R}, x > 0\}$) is the blurred true image. We use subindex i to denote the i -th component of the vector. The deblurring problem is to reconstruct \mathbf{y} from observations \mathbf{z} .

The negative log-likelihood function corresponding to the model (2) takes the form

$$L(\mathbf{y}) = \sum_{i=1}^N [(\mathbf{A} \mathbf{y})_i - z_i \log (\mathbf{A} \mathbf{y})_i], \quad (3)$$

where the term $\log(z_i!)$ independent of \mathbf{y} is omitted.

The maximum likelihood solution $\hat{\mathbf{y}} = \max_{\mathbf{y}} L(\mathbf{y})$ of the Poisson deblurring problem can be obtained using the recursive Richardson-Lucy (RL) algorithm.⁴ Nevertheless, since the problem is ill-conditioned the RL solution is unacceptably noisy and a regularization is typically required. Below we develop an algorithm where regularization is implied through penalizing the complexity of the solution in the BM3D-frame domain.

4. THE ALGORITHM

We formulate the deblurring as a *generalized Nash equilibrium* problem:

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}} L(\mathbf{y}), \text{ subject to } \|\mathbf{y} - \Psi \boldsymbol{\omega}^*\|_2^2 \leq \varepsilon_1, \mathbf{y}_i \geq 0, i = 1 \dots N \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\|_p, \text{ subject to } \|\boldsymbol{\omega} - \Phi \mathbf{y}^*\|_2^2 \leq \varepsilon_2, \end{cases} \quad (4)$$

where $\varepsilon_1, \varepsilon_2 > 0$ and $\|\cdot\|_p$ is the standard notation of the l_p -norm. We consider two cases $p = 0$ and $p = 1$.

In terms of the game theory (4) can be interpreted as a game of two players identified, respectively, with two variables \mathbf{y} and $\boldsymbol{\omega}$. Each player tries to minimize its own cost function. The interaction between players is noncooperative; minimizing its own cost function player in general increases the cost function of the other player. The equilibrium of this game called *Nash equilibrium* defines the fixed-point $(\mathbf{y}^*, \boldsymbol{\omega}^*)$ for (4).

Problem (4) can be replaced by the equivalent one:

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}} L(\mathbf{y}) + \frac{1}{2\gamma} \|\mathbf{y} - \Psi \boldsymbol{\omega}\|_2^2, \text{ subject to } \mathbf{y}_i \geq 0, i = 1 \dots N \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} \|\boldsymbol{\omega}\|_p + \frac{1}{2\tau} \|\boldsymbol{\omega} - \Phi \mathbf{y}\|_2^2. \end{cases} \quad (5)$$

In order to simplify the optimization in the first subproblem in (5) we introduce an additional splitting variable \mathbf{v} and consider the criterion

$$L_1(\mathbf{y}, \boldsymbol{\omega}, \mathbf{v}) = \tilde{L}(\mathbf{v}) + \frac{1}{2\xi} \|\mathbf{v} - \mathbf{A}\mathbf{y}\|_2^2 + \frac{1}{2\gamma} \|\mathbf{y} - \Psi \boldsymbol{\omega}\|_2^2,$$

where $\tilde{L}(\mathbf{v}) = \sum_{i=1}^N [\mathbf{v}_i - \mathbf{z}_i \log \mathbf{v}_i]$ and $\mathbf{v}_i \geq 0$. Denoting $L_2(\mathbf{y}, \boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_p + \frac{1}{2\tau} \|\boldsymbol{\omega} - \Phi \mathbf{y}\|_2^2$, the fixed-point problem is then reformulated as follows:

$$\begin{cases} \mathbf{y}^* = \arg \min_{\mathbf{y}, \mathbf{v}} L_1(\mathbf{y}, \boldsymbol{\omega}^*, \mathbf{v}), \text{ subject to } \mathbf{y}_i \geq 0, \mathbf{v}_i \geq 0, i = 1 \dots N \\ \boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}^*, \boldsymbol{\omega}). \end{cases}$$

To solve this reformulated problem we use the alternating minimization procedure:

$$\begin{cases} \mathbf{v}_{t+1} = \arg \min_{\mathbf{v}} L_1(\mathbf{y}_t, \boldsymbol{\omega}_t, \mathbf{v}) \\ \mathbf{y}_{t+1} = \arg \min_{\mathbf{y}} L_1(\mathbf{y}, \boldsymbol{\omega}_t, \mathbf{v}_{t+1}), \quad t = 0, 1, \dots \\ \boldsymbol{\omega}_{t+1} = \arg \min_{\boldsymbol{\omega}} L_2(\mathbf{y}_{t+1}, \boldsymbol{\omega}) \end{cases} \quad (6)$$

Minimization of L_1 is separable with respect to the components of \mathbf{v} . For each component it takes form

$$(\mathbf{v}_{t+1})_i = \arg \min_{\mathbf{v}_i} \{\mathbf{v}_i - \mathbf{z}_i \log \mathbf{v}_i + (\mathbf{v}_i - (\mathbf{A}\mathbf{y}_t)_i)\},$$

and the non-negative solution is given by the formula

$$(\mathbf{v}_{t+1})_i = \frac{1}{2} \left((\mathbf{A}\mathbf{y}_t)_i - \xi + \sqrt{((\mathbf{A}\mathbf{y}_t)_i - \xi)^2 + 4\xi \mathbf{z}_i} \right).$$

Since L_1 is quadratic its minimization with respect to \mathbf{y} results in a linear solution:

$$\mathbf{y}_{t+1} = \left[\mathbf{A}^T \mathbf{A} + \frac{1}{\gamma} \mathbf{I} \right]^{-1} \times \left[\mathbf{A}^T \mathbf{v}_{t+1} + \frac{1}{\gamma} \max \{ \Psi \boldsymbol{\omega}_t, 0 \} \right].$$

The elementwise maximum operation is applied to the $\Psi \boldsymbol{\omega}_t$ to ensure the nonnegativity of the estimate of \mathbf{y}_{t+1} .

The minimization of L_2 with respect to ω is obtained by the soft or hard thresholding of $\Phi \mathbf{y}_t$, depending whether l_0 or l_1 norm used as a penalty:

$$\omega_{t+1} = \mathfrak{H}_\tau (\Phi \mathbf{y}_{t+1}) = \begin{cases} \text{sign}(\Phi \mathbf{y}_{t+1}) \circ \max(|\Phi \mathbf{y}_{t+1}| - \tau, 0), & p = 1, \\ \Phi \mathbf{y}_{t+1} \circ 1(|\Phi \mathbf{y}_{t+1}| \geq \sqrt{2\tau}), & p = 0. \end{cases}$$

Here all vector operations are elementwise, ' \circ ' stands for the elementwise product of two vectors and $1(\cdot)$ is the indicator function.

Summarizing all steps we arrive to the algorithm presented below which we call Poissonian Iterative Decoupled Deblurring BM3D (PIDD-BM3D).

Algorithm 1 PIDD-BM3D - Poissonian Iterative Decoupled Deblurring BM3D algorithm

- 1: **input:** $\mathbf{z}, \mathbf{A}, \mathbf{y}_{\text{init}}$
 - 2: **initialization:**
 - 3: *using* \mathbf{y}_{init} *construct operators* Φ *and* Ψ
 - 4: *set:* $\mathbf{y}_0, \omega_0 = \Phi \mathbf{y}_0$
 - 5: $t = 0$
 - 6: **repeat**
 - 7: $(\mathbf{v}_{t+1})_i = \frac{1}{2} \left((\mathbf{A}\mathbf{y})_i - \xi + \sqrt{((\mathbf{A}\mathbf{y})_i - \xi)^2 + 4\xi\mathbf{z}_i} \right)$
 - 8: $\mathbf{y}_{t+1} = \left[\mathbf{A}^T \mathbf{A} + \frac{1}{\gamma} \mathbf{I} \right]^{-1} \times \left[\mathbf{A}^T \mathbf{v}_{t+1} + \frac{1}{\gamma} \max\{\Psi \omega_t, 0\} \right]$
 - 9: $\omega_{t+1} = \mathfrak{H}_\tau (\Phi \mathbf{y}_{t+1})$
 - 10: $t = t + 1$
 - 11: **until** convergence.
-

In each iteration the algorithm first updates the blurred image estimate \mathbf{v}_{t+1} , then applies the regularized deblurring to obtain the estimate of the true image \mathbf{y}_{t+1} . Finally, this estimate is filtered by thresholding.

The algorithm is iterated until some convergence criteria is satisfied. Particularly, the iterations can be stopped as soon as the difference between consecutive estimates becomes small enough.

5. IMPLEMENTATION

Grouping and frame operators. The initial image estimate \mathbf{y}_{init} is obtained by the LPA-ICI deblurring algorithm.⁵ The grouping and the adaptive group weights are calculated only once, using \mathbf{y}_{init} and remain unchanged through the subsequent iterations. To build the groups, we apply the block-matching procedure from² to the initial image estimate. In our implementations we use weights inversely proportional to the number of significant spectrum coefficients of the groups. The significant coefficients are found by the hard thresholding of the group spectra of \mathbf{y}_{init} using a small threshold.

The 3-D transform is performed by first applying the 2-D discrete sine transform (DST) to each block in the group, followed by the 1-D Haar transform applied along the third dimension of the group. The image block size is 4×4 , and the number of blocks in the group is 8.

Choice of the regularization parameters. The parameters τ, γ, ξ are roughly optimized using ground truth image to provide best reconstruction quality. Optimization has been performed separately for each image and each deblurring scenario.

Initialization. We experimentally confirmed the convergence to an asymptotic solution that is independent of the initialization of \mathbf{y}_0 and ω_0 . Nevertheless, initialization with a better estimate results in a much faster convergence.

Practical considerations. For a circular shift-invariant blur operator, the solution of the step 8 of the PIDD-BM3D algorithm can be calculated in the Fourier domain using FFT. More, steps 8 and 9 can be merged into a

Scenario	PSF	M
1	7×7 uniform	5
2	7×7 uniform	30
3	7×7 uniform	100
4	7×7 uniform	255
5	9×9 uniform	17400

Table 1. Blur PSF and maximum true image intensity M used in each scenario.

single step

$$\mathbf{y}_{t+1} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}^* (\mathbf{h}) \circ \mathcal{F} (\mathbf{v}_{t+1}) + \frac{1}{\gamma} \mathcal{F} (\max \{ \Psi \mathfrak{I} \mathfrak{h}_\tau (\Phi \mathbf{y}_t), 0 \})}{|\mathcal{F} (\mathbf{h})|^2 + \frac{1}{\gamma}} \right),$$

where the analysis-thresholding-synthesis operation $\Psi \mathfrak{I} \mathfrak{h}_\tau (\Phi \mathbf{y}_t)$ can be calculated groupwise without storing the whole spectrum ω_t in the memory. Here \mathbf{h} denotes the vectorized blurring kernel corresponding to the blur operator \mathbf{A} , and 'o' stands for the elementwise product of two vectors. The operator $\mathcal{F}(\cdot)$ reshapes the input vector into a 2-D array, performs 2-D FFT and vectorizes the obtained result. $\mathcal{F}^{-1}(\cdot)$ works analogously, performing inverse FFT.

Complexity. Application of the frame operators is the most computationally expensive part of the proposed algorithms. However, due to their specific structure, the complexity of the frame operators Φ and Ψ is growing only linearly with respect to the number of the pixels in the image. To give an estimate of the complexity of the PIDD-BM3D algorithm, we mention that, on a 256×256 image, one iteration takes about 0.35 seconds, and about 50 iterations are typically sufficient. This timing has been done on dual core 2.6 GHz processor for an implementation where the computationally most intensive parts have been written in C++.

6. EXPERIMENTS

In this section we compare the PIDD-BM3D algorithm against the state-of-the-art Poissonian deblurring methods and with the IDD-BM3D¹ algorithm. The later algorithm is similar to PIDD-BM3D but designed for the Gaussian data. We consider five standard scenarios used as a benchmarks in the recent publications^{6,3}. The blur PSF $h(x_1, x_2)$ and maximal true image intensity M are summarized in Table 1. These PSFs are normalized such that $\sum h = 1$. The hard thresholding has been used both in the PIDD-BM3D and IDD-BM3D algorithms.

Table 2 presents some of the obtained numerical results. The Mean Absolute Error (MAE) values for the DFS³ and PIDAL-FA⁶ algorithms are taken from the Table II of.⁶ The results of the LPA-ICI,⁵ IDD-BM3D¹ and proposed PIDD-BM3D algorithms were obtained by simulations. For all methods the regularization parameters have been roughly tuned using ground truth images to provide best results.

The data in the table demonstrates that in the most cases the proposed PIDD-BM3D algorithm enables the best results with the significant advantage over competitors. Interesting results demonstrates IDD-BM3D. Despite the fact that it is designed for the different noise model, results of the IDD-BM3D algorithm are close to those by the best Poissonian deblurring methods for all noise levels. This fact speaks in the favour of the BM3D modeling which allows the algorithm to obtain good results even with imprecise noise model.

The visual quality of some of the restored images can be evaluated in Figure 1. One can see that the proposed algorithm is able to suppress the ringing artifacts better than LPA-ICI and provides images containing less restoration artefacts.

7. CONCLUSION

We presented a novel deblurring algorithm for Poissonian images. The proposed algorithm is fast and demonstrates state-of-the-art reconstruction quality.

Though we discussed only deblurring problem, the algorithm can be extended to the other inverse problems as well. As a primary target for the future work we mark two problems: theoretically proving convergence of the algorithm and finding a strategy for selection the optimal parameters for the algorithm.



Figure 1. Deblurring of the *Cameraman* image, scenario 5. From left to right and from top to bottom are presented zoomed fragments of the following images: original, blurred noisy, reconstructed by LPA-ICI⁵ (PSNR 27.2, MAE 432) and by the proposed PIDD-BM3D method (PSNR 28.56, MAE 352).

Scenario	1	2	3	4	5
Image	Cameraman (256x256)				
LPA-ICI ⁵	0.27	1.27	3.77	8.65	432
DFS ³	0.35	1.47	4.31	10.26	-
PIDAL-FA ⁶	0.26	1.22	3.63	8.45	415
IDD-BM3D ¹	0.26	1.19	3.48	7.91	361
PIDD-BM3D (proposed)	0.26	1.18	3.44	7.61	352

Table 2. Comparison of the several deblurring algorithms. Mean Absolute Errors of the outputs.

Acknowledgments

This work is supported by Academy of Finland: project no. 213462, 2006-2011 (Finnish Programme for Centres of Excellence in Research) and project no. 138207, 2011-2014, and by Tampere Doctoral Programme in Information Science and Engineering (TISE).

REFERENCES

- [1] Danielyan, A., Katkovnik, V., and Egiazarian, K., "BM3D frames and variational image deblurring," *accepted Image Processing, IEEE Transactions on*.
- [2] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K., "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.* **16**, 2080-2095 (Aug. 2007).
- [3] Dupe, F.-X., Fadili, J., and Starck, J.-L., "A proximal iteration for deconvolving poisson noisy images using sparse representations," *Image Processing, IEEE Transactions on* **18**, 310–321 (feb. 2009).
- [4] Boccacci, P. and Bertero, M., [*Introduction to Inverse Problems in Imaging*], Taylor & Francis (1998).
- [5] Foi, A., Alenius, S., Trimeche, M., Katkovnik, V., and Egiazarian, K., "A spatially adaptive Poissonian image deblurring," in [*Proc. IEEE Int. Conf. Image Process.*], 925–928 (Sep. 2005).
- [6] Figueiredo, M. and Bioucas-Dias, J., "Restoration of poissonian images using alternating direction optimization," *Image Processing, IEEE Transactions on* **19**, 3133–3145 (dec. 2010).
- [7] Katkovnik, V., Foi, A., Egiazarian, K., and Astola, J., "From local kernel to nonlocal multiple-model image denoising," *International Journal of Computer Vision* **86**, 1–32 (Jan. 2010).

Publication[DFKE10]

A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian,
“Spatially adaptive filtering as regularization in inverse
imaging: compressive sensing, super-resolution, and up-
sampling,” in *Super-Resolution Imaging*, P. Milanfar, Ed.,
pp. 123–153. CRC Press, 2010.

© 2010 CRC Press. Reprinted with permission.

Publication[DFKE08c]

A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian,

“Image and video super-resolution via spatially adaptive block-matching filtering,” in *Proc. International Workshop on Local and Non-Local Approximations in Image Processing*, Lausanne, Switzerland, August 2008, pp. 53–60.

Publication[DFKE08d]

A. Danielyan, A. Foi, V. Katkovnik, and K. Egiazarian,
“Image upsampling via spatially adaptive block-matching
filtering,” *in Proc. European Signal Processing Confer-
ence*, Lausanne, Switzerland, August 2008.

© 2008 EURASIP. Reprinted with permission.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3101-9
ISSN 1459-2045