First-order Convex Optimization Methods for Signal and Image Processing

Jensen, Tobias Lindstrøm

Publication date:
2012

Document Version
Accepted author manuscript, peer reviewed version

First-order Convex Optimization Methods for Signal and Image Processing

Ph.D. Thesis

Tobias Lindstrøm Jensen

Multimedia Information and Signal Processing
Department of Electronic Systems
Aalborg University
Niels Jernes Vej 12, 9220 Aalborg Ø, Denmark

First-order Convex Optimization Methods for Signal and Image Processing
Ph.D. Thesis

ISBN 978-87-92328-76-2
December 2011

Abstract

In this thesis we investigate the use of first-order convex optimization methods applied to problems in signal and image processing. First we make a general introduction to convex optimization, first-order methods and their iteration complexity. Then we look at different techniques, which can be used with first-order methods such as smoothing, Lagrange multipliers and proximal gradient methods. We continue by presenting different applications of convex optimization and notable convex formulations with an emphasis on inverse problems and sparse signal processing. We also describe the multiple-description problem. We finally present the contributions of the thesis.

The remaining parts of the thesis consist of five research papers. The first paper addresses non-smooth first-order convex optimization and the trade-off between accuracy and smoothness of the approximating smooth function. The second and third papers concern discrete linear inverse problems and reliable numerical reconstruction software. The last two papers present a convex optimization formulation of the multiple-description problem and a method to solve it in the case of large-scale instances.

Resumé

I denne afhandling undersøger vi brugen af førsteordens konvekse optimeringsmetoder, anvendt på problemer indenfor signal- og billedbehandling. Først giver vi en general introduktion til konveksoptimering, førsteordensmetoder og deres iterationskompleksitet. Herefter ser vi på forskellige teknikker, som kan benyttes i sammenspil med førsteordensmetoder, f.eks. udglatning, Lagrangemultiplikator- og proksimale gradientmetoder. I de efterfølgende afsnit præsenteres forskellige applikationer af konveksoptimering, samt vigtige formuleringer og algoritmer med hovedvægt på inverse problemer og sparse signalbehandling. Desuden præsenteres flerbeskrivelses problemet. Til sidst i introduktionen præsenteres bidragene af denne afhandling.

Efter introduktionen følger fem artikler. Den første artikel adresserer brugen af førsteordens konveksoptimering for ikke glatte funktioner samt forholdet mellem nøjagtighed og glathed af den tilnærmede glatte funktion. Den anden og tredje artikel omhandler diskrete lineære inverse problemer samt numerisk rekonstruktionssoftware. De sidste to artikler omhandler en konveksoptimerings formulering af flerbeskrivelses problemet, hvor vi diskuterer formuleringen og en metode til at løse storskala problemer.

List of Papers

The main body of this thesis consists of the following papers:

[A] T. L. Jensen, J. Østergaard, and S. H. Jensen, "Iterated smoothing for accelerated gradient convex minimization in signal processing", *in Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Dallas, Texas, Mar., 2010, 774–777.

[B] J. Dahl, P. C. Hansen, S. H. Jensen and T. L. Jensen, (alphabetical order), "Algorithms and software for total variation image reconstruction via first-order methods", *Numerical Algorithms*, 53, 67–92, 2010.

[C] T. L. Jensen, J. H. Jørgensen, P. C. Hansen and S. H. Jensen, "Implementation of an optimal first-order method for strongly convex total variation regularization", *accepted for publication in BIT Numerical Mathematics*, 2011.

[D] T. L. Jensen, J. Østergaard, J. Dahl and S. H. Jensen, "Multiple descriptions using sparse representation", *in Proc. the European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, Aug., 2010, 110–114.

[E] T. L. Jensen, J. Østergaard, J. Dahl and S. H. Jensen, "Multiple-description l_1-compression", *IEEE Transactions on Signal Processing*, 59, 8, 3699–3711, 2011.

Besides the above papers, the author has participated in the development of the numerical software mxTV (see paper B) and TVReg (see paper C) for discrete linear inverse problems.

Other relevant publications which are omitted in this thesis:

[1] J. Dahl, J. Østergaard, T. L. Jensen and S. H. Jensen, "An efficient first-order method for ℓ_1 compression of images", *in Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipai, Taiwan, Apr., 2009, 1009–1012.

[2] J. Dahl, J. Østergaard, T. L. Jensen and S. H. Jensen, "ℓ_1 compression of image sequences using the structural similarity index measure", *in Proc. IEEE Data Compression Conference (DCC)*, Snowbird, Utah, 2009, 133–142.

[3] T. L. Jensen, J. Dahl, J. Østergaard and S. H. Jensen, "A first-order method for the multiple-description l1-compression problem", *Signal Processing with Adaptive Sparse Structured Representations (SPARS'09)*, Saint-Malo, France, Apr., 2009.

Preface

This thesis is written in partial fulfilment of the requirements for a Ph.D. degree from Aalborg University. The majority of the work was carried out from Sep. 2007 to Oct. 2010 when I was a Ph.D. student at Aalborg University. My work was supported by the CSI: Computational Science in Imaging project from the Danish Technical Research Council, grant no. 274-07-0065.

First I would like to thank my supervisor Søren Holdt Jensen for giving me a chance in the world of research and letting me pursue the topics I have found interesting. I am grateful that my co-supervisor Joachim Dahl has introduced me to convex optimization and all its possibilities and limitations. A big thank you goes to my other co-supervisor Jan Østergaard for his commitment and effort on almost all aspects of a Ph.D. project.

I would also like to thank my colleagues at DTU, who have significantly contributed to the results in this thesis. I appreciate the discussions with Per Christian Hansen on inverse problems and the often tricky numerical aspects to consider when designing software. My pseudo-twin of the CSI project, Jakob Heide Jørgensen has been a great collaborator in all details from problem formulation, numerical aspects and tests, to paper writing. A special thank goes to Lieven Vandenberghe for hosting my stay at UCLA - his great course and lecture notes on first-order methods and large scale convex optimization have helped to form my understanding of the field. Finally, I am grateful for the detailed comments and suggestions of Prof. Dr. Moritz Diehl, K.U. Leuven.

My colleagues at Aalborg also deserve thanks. The office buddies Thomas Arildsen and Daniele Giacobello for discussions and every day social life. The general research related discussion with Mads Græsbøll Christensen have also shaped my view on what research really is. I would also like to thank the "Danish lunch time"-group for offering a great daily break with various discussion topics.

Last but not least, to Wilfred for being a lovely manageable child and to Stine for love and support.

Tobias Lindstrøm Jensen

Contents

Abstract **i**

Resumé **iii**

List of Papers **v**

Preface **vii**

1 Introduction **1**
1 Convex Optimization 1
2 First-Order Methods 3
3 Techniques 9
4 When are First-Order Methods Efficient? 12

2 Applications **15**
1 Inverse Problems 15
2 Sparse Signal Processing 16
3 Multiple Descriptions 18

3 Contributions **21**

References **22**

Paper A: Iterated Smoothing for Accelerated Gradient Convex Minimization in Signal Processing **37**
1 Introduction 39
2 A Smoothing Method 40
3 Restart 42
4 Iterated Smoothing 42
5 Simulations 44
6 Conclusions 48

References 48

Paper B: Algorithms and Software for Total Variation Image Reconstruction via First-Order Methods 51
1 Introduction 53
2 Notation 54
3 Denoising 56
4 Inpainting 60
5 Deblurring for Reflexive Boundary Conditions 62
6 Numerical Examples 66
7 Performance Studies 68
8 Conclusion 74
Appendix A: The Matlab Functions 76
Appendix B: The Norm of the Derivative Matrix 78
References 78

Paper C: Implementation of an Optimal First-Order Method for Strongly Convex Total Variation Regularization 83
1 Introduction 85
2 The Discrete Total Variation Reconstruction Problem 87
3 Smooth and Strongly Convex Functions 88
4 Some Basic First-Order Methods 90
5 First-Order Inequalities for the Gradient Map 93
6 Nesterov's Method With Parameter Estimation 94
7 Numerical Experiments 98
8 Conclusion 105
Appendix A: The Optimal Convergence Rate 106
Appendix B: Complexity Analysis 109
References 113

Paper D: Multiple Descriptions using Sparse Decompositions 119
1 Introduction 121
2 Convex Relaxation 123
3 A First-Order Method 124
4 Simulations 128
5 Discussion 132
References 132

Paper E: Multiple-Description l_1-Compression 137
1 Introduction 139
2 Problem Formulation 141
Analysis of the Multiple-description l_1-Compression Problem 143

3 Analysis of the Multiple-description l_1-Compression Problem . . 143
4 Solving the MD l_1-Compression Problem 145
5 Analyzing the Sparse Descriptions 153
6 Simulation and Encoding of Sparse Descriptions 154
7 Conclusion 163

Chapter 1

Introduction

There are two main fields in convex optimization. First, understanding and formulating convex optimization problems in various applications such as in estimation and inverse problems, modelling, signal and image processing, automatic control, statistics and finance. Second, solving convex optimization problems. In this thesis we will address both fields. In the remaining part of Chapter 1 we describe methods for solving convex optimization problems with a strong emphasis on first-order methods. In Chapter 2 we describe different applications of convex optimization.

1 Convex Optimization

We shortly review basic results in convex optimization, following the notation in [1]. A constrained minimization problem can be written as

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad \forall\, i = 1, \cdots, m \\ & h_i(x) = 0, \quad \forall\, i = 1, \cdots, p\,, \end{array} \tag{1.1}$$

where $f_0(x) : \mathbf{R}^n \mapsto \mathbf{R}$ is the *objective function*, $f_i(x) : \mathbf{R}^n \mapsto \mathbf{R}$ are the *inequality constraints* and $h_i(x) : \mathbf{R}^n \mapsto \mathbf{R}$ are the *equality constraints*. We call a problem a convex problem if $f_i, \forall\, i = 0, \cdots, m$, are convex and $h_i, \forall\, i = 1, \cdots, p$, are affine. Convex problems are a class of optimization problems which can be solved using efficient algorithms [2].

An important function in constrained optimization is the *Lagrangian*

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x), \tag{1.2}$$

where $(\lambda, \nu) \in \mathbf{R}^m \times \mathbf{R}^p$ are called the Lagrange multipliers or the *dual variables*. Define the dual function by

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu)\,. \tag{1.3}$$

The (Lagrange) *dual problem* is then given as

$$\begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda_i \geq 0, \quad \forall\, i = 1, \cdots, m\,. \end{array} \tag{1.4}$$

For a convex problem, necessary and sufficient conditions for primal and dual optimality for differentiable objective and constraint functions are given by the Karush-Kuhn-Tucker (KKT) conditions

$$\left\{ \begin{array}{rl} f_i(x^\star) \leq 0, & \forall\, i = 1, \cdots, m \\ h_i(x^\star) = 0, & \forall\, i = 1, \cdots, p \\ \lambda_i^\star \geq 0, & \forall\, i = 1, \cdots, m \\ \lambda_i^\star f_i(x^\star) = 0, & \forall\, i = 1, \cdots, m \\ \nabla f_0(x^\star) + \sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) + \sum_{i=1}^p \nu_i^\star \nabla h_i(x^\star) = 0. & \end{array} \right. \tag{1.5}$$

1.1 Methods

The above formulation is a popular approach, but for convenience we will write this as

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathcal{Q} \end{array} \tag{1.6}$$

where $f(x) = f_0(x)$ and

$$\mathcal{Q} = \{x \,|\, f_i(x) \leq 0,\, \forall\, i = 1, \cdots, m;\, h_i(x) = 0,\, \forall\, i = 1, \cdots, p\,\}. \tag{1.7}$$

Methods for solving the problem (1.6) are characterized by the type of structural information, which can be evaluated.

- *Zero-order oracle*: evaluate $f(x)$.
- *First-order oracle*: evaluate $f(x)$ and $\nabla f(x)$.
- *Second-order oracle*: evaluate $f(x)$, $\nabla f(x)$ and $\nabla^2 f(x)$.

To exemplify, an exhaustive search in combinatorial optimization employs a zero-order oracle. The classic gradient method or steepest descent [3], conjugate gradient [4], quasi-Newton [5–10], heavy ball [11] employ a first-order oracle. Newtons method and interior-point methods (where f is a modified function) [2, 12–14] employ a second-order oracle.

Since only a minor set of problems can be solved using closed-form solutions with an accuracy given by the machine precision, it is common to say that to solve a problem is to find an approximate solution with a certain accuracy ϵ [15]. We can now define the two complexity measures:

- *Analytical complexity* The number of calls of the oracle which is required to solve the problem up to accuracy ϵ.
- *Arithmetical complexity* The total number of arithmetic operations to solve the problem up to accuracy ϵ

For an iterative method, if an algorithm only calls the oracle a constant number of times in each iteration, the analytical complexity is also referred to as the iteration complexity.

Since zero-order methods have the least available structural information of a problem, we would expect zero-order methods to have higher analytical complexity than first- and second-order methods. Equivalently, first-order methods are expected to have higher analytical complexity than second-order methods. However, the per-iteration arithmetic complexity is expected to operate in an opposite manner, second-order methods have higher per-iteration complexity than first-order methods an so forth. So, if we are interested in which method is the most "efficient" or "fastest", we are interested in the arithmetical complexity which is a product of the iteration (analytical) complexity and the per-iteration arithmetical complexity. We will discuss this trade-off between higher and lower iteration complexity and per-iteration arithmetical complexity for specific first- and second-order methods in Sec. 4.

2 First-Order Methods

We will review first-order methods in the following subsections. To this end, we present some important definitions involving first-order inequalities [15].

Definition 2.1. *A function f is called convex if for any $x, y \in \mathbf{dom}\, f$ and $\alpha \in [0,1]$ the following inequality holds*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)\,. \tag{1.8}$$

For convenience, we will denote the set of general convex functions $\mathcal{C}$, and closed convex functions $\bar{\mathcal{C}}$.

Definition 2.2. *Let f be a convex function. A vector $g(y)$ is called a subgradient of function f at point $y \in \mathbf{dom}\, f$ if for any $x \in \mathbf{dom}\, f$ we have*

$$f(x) \geq f(y) + g(y)^T (x - y)\,. \tag{1.9}$$

The set of all subgradients of f at y, $\partial f(y)$ is called the subdifferential of function f at point y.

For convenience, we will denote the set of subdifferentiable functions $\mathcal{G}$, *i.e.*, functions for which (1.9) hold for all $y \in \mathbf{dom}\, f$.

Definition 2.3. *The continuously differentiable function f is said to be* μ-strongly convex *if there exists a $\mu \geq 0$ such that*

$$f(x) \geq f(y) + \nabla f(y)^T (x - y) + \tfrac{1}{2}\mu \|x - y\|_2^2, \qquad \forall\, x, y \in \mathbf{R}^n. \tag{1.10}$$

The function f is said to be L-smooth *if there exists an $L \geq \mu$ such that*

$$f(x) \leq f(y) + \nabla f(y)^T (x - y) + \tfrac{1}{2} L \|x - y\|_2^2, \qquad \forall\, x, y \in \mathbf{R}^n. \tag{1.11}$$

The set of functions that satisfy (1.10) and (1.11) is denoted $\mathcal{F}_{\mu,L}$. The ratio $Q = L/\mu$, is referred to as the "modulus of strong convexity" [16] or the "condition number for f" [15] and is an upper bound on the *condition number* of the Hessian matrix. For twice differentiable functions

$$Q \geq \frac{\max_x \lambda_{\max}(\nabla^2 f(x))}{\min_x \lambda_{\min}(\nabla^2 f(x))}. \tag{1.12}$$

In particular, if f is a quadratic function, $f(x) = \frac{1}{2} x^T A\, x + b^T x$, then $Q = \kappa(A)$.

2.1 Complexity of First-Order Black-Box Methods

Important theoretical work on the complexity of first-order methods was done in [16]. However, optimal first-order methods for smooth problems were not known at the time [16] was written. An optimal first-order methord was first given in [17]. In this light, the material [16] is not up to date. On the other hand, the material in [15] includes both the complexity analysis based on [16] and optimal first-order methods for smooth problems and it is therefore possible to give a better overview of the field. This is the reason we will make most use of the material presented in [15].

In a historical perspective, it is interesting to note that complexity analyses of first-order methods [16] as well as the optimal methods [17] were forgotten in many years, although the same authors continued publishing in the field [11, 18, 19]. As noted in [15], research in polynomial-time algorithms was soon to set off based on [12]. The advent of polynomial-time algorithms might have left no interest in optimal first-order methods or knowledge of their existence (many of the publications above first appeared in Russian).

A first-order method is called optimal for a certain class if there exist a single problem in the class for which the complexity of solving this problem

coincides with the first-order methods worst-case complexity for all problems in the class. Note that this means that there may be problems in the class for which a first-order method has lower complexity. The definition of optimality also involves two important assumptions. First, for smooth problems, optimal complexity is only valid under the assumption that the number of iterations is not too large compared to the dimensionality of the problem [15]. Specifically, it is required that $k \leq \frac{1}{2}(n-1)$, where k is the number of iterations and n is the dimension of the problem. This is not a significant issue when dealing with large-scale problems since even for $n = 10000$, the optimality definition holds up to $k \leq 4999$. There is a special case, where it has been shown that the Barzilai-Borwein strategy exhibits superlinear convergence for strictly quadratic problems in the case of $n = 2$ [20], *i.e.*, the optimality condition does not hold for $k \geq 1$. For larger dimensions, the strongest results for strictly quadratic problems show that the Barzilai-Borwein strategy has the same complexity as the gradient method [21][1]. Second, we need to restrict ourselves to "black-box schemes", where the designer is not allowed to manipulate the structure of the problem; indeed, it has been shown that for subdifferentiable problems, non black-box schemes can obtain better complexity than the optimal black-box complexity [23–25]. The same idea underlines polynomial time path-following interior-point methods [2, 12–14].

One question naturally arises – why do we consider optimal first-order methods? Why not use all the specialized algorithms developed for special problems? First, since we are considering a range of problems occurring in image and signal processing, it is interesting to investigate a relatively small set of algorithms which are provably optimal for all the problems. Besides, optimal first-order methods should not be compared to specialized algorithms but with the twin method: the gradient-/steepest descent method. As discussed in Sec. 3 on different techniques to construct complete algorithms, many techniques can be used with both the gradient method and the optimal/accelerated first-order method.

Following [15], we will define a first-order black-box method as follows:

Definition 2.4. *A first-order method is any iterative algorithm that selects*

$$x^{(k)} \in x^{(0)} + span\left\{\nabla f(x^{(0)}), \nabla f(x^{(1)}), \cdots, \nabla f(x^{(k-1)})\right\} \tag{1.13}$$

for differentiable problems or

$$x^{(k)} \in x^{(0)} + span\left\{g(x^{(0)}), g(x^{(1)}), \cdots, g(x^{(k-1)})\right\} \tag{1.14}$$

for subdifferentiable problems.

[1] In [22] it was argued that "[i]n practice, this is the best that could be expected from the Barzilai and Borwein method."

ID	Function	Analytical complexity
A	$f \in \mathcal{G}$	$\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$
B	$f(x) = h(x) + \Psi(x)$, $h \in \mathcal{F}_{0,L}$, $\Psi \in \mathcal{G}$	$\mathcal{O}\left(\sqrt{\frac{L}{\epsilon}}\right) + \mathcal{O}\left(\frac{1}{\epsilon^2}\right)$
C	$f \in \mathcal{F}_{0,L}$	$\mathcal{O}\left(\sqrt{\frac{L}{\epsilon}}\right)$
D	$f \in \mathcal{F}_{\mu,L}$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right)$

Table 1.1: Worst-case and optimal complexity for first-order black-box methods.

ad **A**: An optimal method for $f \in \mathcal{G}$ is the subgradient method.

ad **B**: An optimal method was given in [26].

ad **C, D**: An optimal method for the last two function classes was first given in [17], other variants exists [15, 19, 23], see the overview in [27].

Consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \mathbf{R}^n\,, \end{array} \tag{1.15}$$

which we will solve to an accuracy of $f(x^{(k)}) - f^\star \leq \epsilon$, *i.e.*, $x^{(k)}$ is an ϵ-suboptimal solution. The optimal complexity for different classes are reported in Table 1.1.

For quadratic problems with $f \in \mathcal{F}_{\mu,L}$, the conjugate gradient method achieves the same iteration complexity [16] as optimal first-order methods achieve for all $f \in \mathcal{F}_{\mu,L}$.

2.2 Complexity for First-Order Non Black-Box Methods

In the case of optimal methods, we restricted ourself to black-box schemes. However, "[i]n practice, we never meet a pure black box model. We always know something about the structure of the underlying objects" [23]. In the case we dismiss the black-box model, it is possible to obtain more information of the problem at hand and as a consequence decrease the complexity. Complexity for non black-box first-order methods are reported in Table 1.2.

ID	Function	Analytical complexity
E	$f(x) = \max_{u \in U} u^T Ax, f \in \mathcal{C}$	$\mathcal{O}\left(\frac{1}{\epsilon}\right)$
F	$f(x) = h(x) + \Psi(x),\ h \in \mathcal{F}_{0,L},\ \Psi \in \bar{\mathcal{C}}$	$\mathcal{O}\left(\sqrt{\frac{L}{\epsilon}}\right)$
G	$f(x) = h(x) + \Psi(x),\ h \in \mathcal{F}_{\mu,L},\ \Psi \in \bar{\mathcal{C}}$	$\mathcal{O}\left(\sqrt{\frac{L}{\mu}} \log \frac{1}{\epsilon}\right)$

Table 1.2: Worst-case complexity for certain first-order non black-box methods.

ad **E**: A method for non-smooth minimization was given in [23], which can be applied to more general models. The *extra gradient* method [28] was shown to have the same complexity [29] and the latter applies to the more general problem of variational inequalities. Some $f \in \mathcal{G}$ can be modelled as this max-type function. This method does not apply to all functions $f \in \mathcal{C}$ but only those that can me modelled as a max-type function.

ad **F**: A method for this composite objective function was given in [24, 25].

ad **G** See [24].

If we compare Tables 1.1 and 1.2, we note similarities between **C**-**F** and **D**-**G**. This comes from a modified first-order method, which handles the more general convex function using the so-called proximal map, see Sec. 3.2. This essentially removes the complexity term which give the worst complexity.

2.3 Algorithms

We will now review two important first-order methods: the classic gradient method and an optimal method. The gradient method is given below.

Gradient method

Given $x^{(0)} \in \mathbf{R}^n$

for $k = 0, \cdots$

$x^{(k+1)} = x^{(k)} - t_k \nabla f(x^{(k)})$

Traditional approaches for making the gradient method applicable and efficient are based on selecting the stepsize t_k appropriate.

Stepsize selection techniques include [15, 30]:

- Constant stepsize.
- Minimization rule/exact line search/full relaxation.
- Backtracking line search with Armijo rule.
- Goldstein-Armijo rule.
- Diminishing stepsize.
- Barzilai-Borwein strategy [20].

The line searches can also be limited and/or include non-monotonicity. Despite the efforts it has not been possible to obtain better theoretical convergence rate than that of the constant stepsize $t_k = \frac{1}{L}$ [15, 21].

An optimal method is given below [15].

Optimal first-order method

Given $x^{(0)} = y^{(0)} \in \mathbf{R}^n$ and $1 > \theta_0 \geq \sqrt{\frac{\mu}{L}}$

for $k = 0, \cdots$

$x^{(k+1)} = y^{(k)} - \frac{1}{L} \nabla f(y^{(k)})$

θ_{k+1} positive root of $\theta^2 = (1-\theta)\theta_k^2 + \frac{\mu}{L}\theta$

$\beta_k = \frac{\theta_k(1-\theta_k)}{\theta_k^2 + \theta_{k+1}}$

$y^{(k+1)} = x^{(k+1)} + \beta_k \left(x^{(k+1)} - x^{(k)} \right)$

For the case $\mu = 0$, it is possible to select $\beta_k = \frac{k}{k+3}$ and for $\mu > 0$ it is possible to select $\beta_k = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ [15, 27]. The optimal gradient method has a close resemblance to the heavy ball method [31]. In fact, the heavy ball method [11, Sec. 3.2.1] and the two step method [32] are both optimal for unconstrained optimization for $f \in \mathcal{F}_{\mu,L}$ (compare with the analysis in [15]).

3 Techniques

In this section we will give an overview of techniques used to solve different optimization problems.

3.1 Dual Decomposition

An approach to handle problems with intersecting constraints/intersections, sometimes referred to as complicating or coupling constraints, is the dual decomposition [30, 33]. This approach is only applicable when the primal objective is separable. The dual problem is then solved using a sub-/gradient method. This idea is exploited in Chambolle's algorithm for total variation denoising [34], see also [35] for total variation deblurring, routing and resource allocation [36, 37], distributed model predictive control [38] and stochastic optimization [39].

3.2 Proximal Map

The gradient map [16] was defined to generalize the well-known gradient from unconstrained to constrained problems. This idea can be extended to other more complicated functions using the proximal map. If the objective has the form $f(x) = h(x) + \Psi(x)$, $h \in \mathcal{F}_{\mu,L}$, $\Psi \in \mathcal{C}$, then we can use the proximal map defined by Moreau [40] to handle the general convex function Ψ in an elegant way. The proximal map is defined as

$$\mathbf{prox}_{\Psi}(x) = \underset{u}{\operatorname{argmin}} \left(\Psi(u) + \frac{1}{2}\|u - x\|_2^2 \right) .$$

The proximal map generalizes the projection operator in the case Ψ is the indicator function of the constrained set $\mathcal{Q}$. In the case $\Psi(x) = \|x\|_1$, it was shown how to use the proximal map to form proximal gradient algorithms for linear inverse problems [41–44] in which case the proximal map becomes the *soft-threshold* or *shrinkage* operator [45, 46]. In the case of the Nuclear norm $\Psi(X) = \|X\|_*$ the proximal map is the *singular value threshold* [47]. These algorithms can be seen in the view of forward-backward iterative schemes [48, 49] with the forward model being a gradient or Landweber iteration. The proximal map may have a closed form solution [25] or require iterative methods [50]. The mirror descent algorithm [16] is also closely related to the proximal map framework [51]. The proximal map can be combined with optimal methods for smooth problems to obtain *accelerated* proximal gradient methods [24, 25, 52], see [53–55] for nuclear norm minimization or the overview work in [27, 56]. The extra gradient method of Nemirovskii also relies on the proximal map [29], see [57] for applications in image processing.

3.3 Smoothing

The convergence rate for non-smooth problems may be too low for certain problems in which case a solution is to make a smooth approximation of the considered non-smooth function. The motivation is that a smooth problem has better convergence properties as reported in Sec. 2.1. This is, *e.g.*, used for smoothing total-variation problems [58, 59].

Smoothing can also be used to obtain one order faster convergence for non-smooth problems compared to the sub-gradient method [23]. The idea is to form a smooth approximation with known bounds and subsequently apply an optimal first-order method to the smooth function. Following the outline in [31], consider a possible non-smooth convex function on the form[2]

$$f(x) = \max_{u \in \mathcal{U}} u^T A x \,.$$

We can then form the approximation

$$f_\mu(x) = \max_{u \in \mathcal{U}} u^T A x - \mu \tfrac{1}{2} \|u - \bar{u}\|_2^2$$

with

$$\Delta = \max_{u \in \mathcal{U}} \tfrac{1}{2} \|u - \bar{u}\|_2^2 \,.$$

Then the approximation $f_\mu(x)$ bounds $f(x)$ as

$$f_\mu(x) \leq f(x) \leq f_\mu(x) + \mu\Delta$$

and $f_\mu(x)$ is Lipschitz continuous with constant $L_\mu = \frac{\|A\|_2^2}{\mu}$. If we select $\mu = \frac{\epsilon}{2\Delta}$ and solve the smoothed problem to accuracy $\epsilon/2$ we have

$$f(x) - f^\star \leq f_\mu(x) - f_\mu^\star + \mu\Delta \leq \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$$

so we can achieve an ϵ-suboptimal solution for the original problem in

$$\mathcal{O}\left(\sqrt{\frac{L_\mu}{\epsilon/2}}\right) = \mathcal{O}\left(\sqrt{\frac{2\|A\|_2^2}{\mu\epsilon}}\right) = \mathcal{O}\left(\sqrt{\frac{4\Delta\|A\|_2^2}{\epsilon^2}}\right) = \mathcal{O}\left(\frac{2\sqrt{\Delta}\|A\|_2}{\epsilon}\right)$$

iterations if we use an optimal first-order method to solve the smoothed problem. This approach has motivated a variety works [35, 60–65]. Another way to handle the non-smoothness of, *e.g.*, the $\|x\|_1$-norm, is to make an equivalent smooth and constrained version of the same problem [66, 67], using standard reformulation techniques.

[2] More general models of the function f is given in [23].

3.4 Lagrange Multiplier Method

This method is sometimes also known as the augmented Lagrangian method or the method of multipliers. Initially suggested in [68, 69], see [70] for a more complete consideration of the subject. The idea is to augment a quadratic function to the Lagrangian to penalize infeasible points and then solve a series of these augmented Lagrangian problems – updating the dual variables after each approximate solution. It is important that the augmented Lagrange problem is sufficiently easy to solve and that we can benefit from warm start to make the Lagrange multiplier method efficient. The Lagrange multiplier method is equivalent to applying the gradient method to a Moreau-Yoshida regularized version of the dual function [71, 72]. To see this, consider for simplicity the following convex problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array} \tag{1.16}$$

with the Lagrange dual function $g(v) = -f^*(-A^T v) - v^T b$, where $f^*(y) = \sup_{x \in \mathbf{dom}\, f} x^T y - f(x)$ is the conjugate function. The dual problem is then

$$\text{maximize} \quad g(v)$$

which is equivalent to the Moreau-Yoshida regularized version

$$\text{maximize}_v \quad g_\mu(v), \qquad g_\mu(v) = \sup_z \left(g(z) - \tfrac{1}{2\mu}\|z - v\|_2^2 \right) \tag{1.17}$$

for $\mu > 0$. To solve the original problem (1.16), we will solve the above problem (1.17) by maximizing $g_\mu(v)$. Note that $g_\mu(v)$ is smooth with constant $L = \frac{1}{\mu}$. To evaluate $g_\mu(v)$ and $\nabla g_\mu(v)$, we note that the dual problem of

$$\text{maximize}_z \quad g(z) - \tfrac{1}{2\mu}\|z - v\|_2^2$$

is

$$\begin{array}{ll} \text{minimize}_{x,y} & f(x) + v^T y + \frac{\mu}{2}\|y\|_2^2 \\ \text{subject to} & Ax - b = y \end{array}$$

and $\nabla g_\mu(v) = A\hat{x}(v) - b$ [31], where $\hat{x}(v)$ is the minimizer in the above expression for a given v. If we then apply the gradient method for maximizing $g_\mu(v)$ with constant stepsize $t = \frac{1}{L} = \mu$, we obtain the algorithm

$$\begin{aligned} &\text{for } k = 1, 2, \cdots \\ &\qquad x^{(k)} = \operatorname*{argmin}_x f(x) + v^{(k-1)T}(Ax - b) + \frac{\mu}{2}\|Ax - b\|_2^2 \\ &\qquad v^{(k)} = v^{(k-1)} + \mu(Ax^{(k)} - b)\,. \end{aligned}$$

The above algorithm is the Lagrange multiplier method [70] for the problem (1.16). Further, the Lagrange multiplier method is the same as the Bregman iterative method in certain cases [73], *e.g.*, for (1.16) with $f(x) = \|x\|_1$ (see also [74] for total variation problems). The Lagrange multiplier method is applied in many algorithms, *e.g.*, [54, 75, 76].

A related approach in the case of coupled variables in the objective is to apply alternating minimization [77]. Alternatively, we can first apply variable splitting [75, 76, 78–80] and then apply alternating minimization and/or Lagrange multiplier method on the new problem. Splitting and alternating minimization can also be combined with accelerated methods and smoothing [81].

3.5 Continuation and Restart

For a specific parameter setting of an algorithm, the convergence may be slow. The idea in continuation/restart/re-initialization is then to adapt the parameter settings, and instead run a series of stages for which we approach the requested parameter setting – utilizing warm-start at each stage. Even though such a scheme requires several stages, the overall efficiency may be better if the warm-start procedure provides a benefit. A well-known example is the barrier method where the barrier parameter is the setting under adaptation [1, 82].

One application is in case of regularized problems in unconstrained form, where it is noted that the regularization parameter determines the efficiency of the method, in which case continuation has successfully been applied in [67, 83]. The observation that the regularization parameter determines the efficiency of a first-order method can also be motivated theoretically since the Lipschitz constant of the gradient function is a function of the regularization parameter [84]. Continuation has also motivated the approach in [62] and the use of accelerated continuation in [85].

There are limited theoretical results on the continuation scheme presented above. However, in the case of strongly convex functions it is possible to obtain guarantees. For smooth problems, it can be used to obtain optimal complexity [16, 17]. In some cases we have functions which are not strongly convex but show a similar behaviour [61]. A permutation using a strongly convex functions can also be used to obtain theoretical results in case the original problem is not strongly convex [86].

4 When are First-Order Methods Efficient?

It is interesting to discuss when first-order methods are efficient compared to second-order methods such as interior-point methods. A good measure of efficiency is the arithmetical complexity of those methods. In the following we

will discuss three important parameters which determine the *favourable* choice of first-order versus second-order methods.

4.1 When the dimension of the problem is sufficiently large

Second-order methods employing direct methods scale as $\mathcal{O}(n^3)$ in the dimensions to obtain the step direction[3]. However, additional structure in the problem can be utilized to reduce the arithmetic complexity. The most expensive operation for first-order methods is often the multiplication with a matrix $A \in \mathbf{R}^{q\times n}$ to obtain the step direction, which scale as $\mathcal{O}(qn)$ in the dimensions. Second-order methods employing iterative methods scales as $\mathcal{O}(qn)$, but efficient use usually requires preconditioning. Empirical analysis of arithmetic complexity for the well structured basis pursuit denoising problem using an interior-point method and preconditioned conjugate gradient with $q = 0.1n$ and approximately $3n$ nonzero elements in A, show no better complexity than $\mathcal{O}(n^{1.2})$ when solved to moderate accuracy [67, 88]. Many of the other second-order methods show no better than $\mathcal{O}(n^2)$ complexity. But even at small problems $n = 10^4$, a first-order method based on the Barzilai-Borwein strategy is more efficient than the second-order method. Results on total variation denoising in constrained form using an interior-point method showed empirical $\mathcal{O}(n^2 \log n)$ to $\mathcal{O}(n^3)$ complexity with nested dissection [89] and a standard second-second order cone solver [90]. Analysis in [91] also shows that a second-order method [92] using direct methods (the conjugate gradient method was not efficient on the investigated examples) scaled worse in the dimensions than the investigated first-order methods. In [93] it is shown that for low accuracy solutions, a first-order method scales better than an interior-point method using preconditioned conjugate gradient, and is more efficient for the investigated dimensions. To conclude, both theory and empirical data shows that first-order methods scale better in the dimension, *i.e.*, first-order methods are the favourable choice for sufficiently large problems.

4.2 When the proximal map is simple

In the case of constrained problems, the proximal map becomes the projection operator. If the projection can be calculated efficiently then the arithmetical complexity is not significantly larger than that of an unconstrained problem. This includes constraints such as box, simplex, Euclidean ball, 1-norm ball, small affine sets or invertible transforms and simple second-order cones [15, 31]. The same holds for the proximal map when based on, *e.g.*, the Euclidean norm, 1-norm, ∞-norm, logarithmic barrier, or the conjugate of any of the previous functions, see [56] for an extensive list. On the other hand, if we need to rely

[3]The worst-case analytical complexity scales as $\mathcal{O}(\sqrt{m})$ for path-following interior-point methods, but in practice it is much smaller or almost constant [87].

on more expensive operations to solve the proximal map, the arithmetical complexity may be significantly larger. This can occur with affine sets involving a large/dense/unstructured matrix, large/unstructured positive semidefinite cones or nuclear norms such that the singular value decomposition is arithmetically expensive to compute. For the case of inaccurate calculation of the proximal map, different behaviour for the gradient and optimal/accelerated first-order method can be expected. It is shown that the gradient method only yields a constant error offset [94]. The case for optimal/accelerated methods is more complicated depending on the proximal map error definition. Under the most restrictive proximal map error definition [94], the optimal/accelerated first-order method only shows a constant error offset [95, 96], but under less restrictive definitions the optimal/accelerated method must suffer from accumulation of errors [94]. An example of this is in total variation regularization using the proximal map where it is required to solve a total variation denoising problem in each iteration to obtain an approximate proximal map. For an insufficient accurate proximal map, the overall algorithm can diverge and suffer from accumulation of errors as shown in [50]. If the proximal map error can be chosen, which happens in most practical cases, it is possible to obtain ϵ accuracy with the same analytical complexity as with exact proximal map calculations [94]. Whether to choose the gradient or an optimal/accelerated first-order method in the case of inaccurate proximal map depends on the arithmetical complexity of solving the proximal map to a certain accuracy [94]. Such two level methods with gradient updates are used in [50, 97]. The use of inaccurate proximal maps combined with other techniques is also presented and discussed in [79, 98].

4.3 When the problem class is favourable compared to the requested accuracy

First-order methods are sensitive to the problem class. Compare, *e.g.*, non-smooth problems with iteration complexity $\mathcal{O}(1/\epsilon)$ using smoothing techniques with strongly-convex smooth problems with iteration complexity $\mathcal{O}\left(\sqrt{L/\mu}\log 1/\epsilon\right)$. Clearly, if we request high accuracy (small ϵ) the latter problem class has a much more favourable complexity. On the other hand, for low accuracy (large ϵ) the difference between the two problem classes might not be that significant. Second-order methods scale better than first-order methods in the accuracy, *e.g.*, quadratic convergence of the Newton method in a neighbourhood of the solution. This means that for sufficiently high accuracy, second-order methods are the favourable choice, see [88, 91].

Chapter 2

Applications

The application of convex optimization has expanded greatly in recent years and we will therefore not try to make a complete coverage of the field but only address a few applications and only focus on areas closely related to this thesis. To be specific, we will discuss inverse problems, sparse signal processing and multiple descriptions (MDs) in the following sections. For more overview literature on convex optimization for signal processing, we refer to [99–103] and for image processing (imaging) to [59, 104].

1 Inverse Problems

In signal and image processing, an inverse problem is the problem of determining or estimating the input signal which produced the observed output. Inverse problems arise frequently in engineering and physics, where we are interested in the internal state (input) of a system which gave cause to the measurements (output). Applications are, *e.g.*, geophysics, radar, optics, tomography and medical imaging.

An important concept in inverse problems is whether the specific problem is well-posed or ill-posed – a concept defined by Hadamard. For our purpose, it is adequate to state that a problem is ill-posed if,

a) the solution is not unique, or

b) the solution is not a continuous function of the data.

Even in the case of a solution being a continuous function of the data, the sensitivity can be high, *i.e.*, almost discontinuous, in which case it is said to be ill-conditioned. An engineering interpretation of the latter statement is that

small permutations in the measurement data can lead to large permutations of the solution [105].

A popular estimation method is *maximum likelihood (ML) estimation*. However, for ill-conditioned problems this approach is not suitable [106] in which case one can use *maximum a posteriori (MAP) estimation* where additional information of the solution is included to stabilize the solution [105]. The additional information introduces the regularizer. The problem can also be seen from a statistical point of view where the regularizer is introduced to prevent overfitting.

Classic approaches to obtain a faithful reconstruction of the unknown variables are Tikhonov regularization [107], truncated singular value decomposition [108, 109] and iterative regularization methods using semiconvergence, using *e.g.*, the conjugate gradient method [110, 111]. Tikhonov regularization is a MAP estimator if the unknown signal and noise have a Gaussian distribution. Tikhonov regularization applies Euclidean norm regularization/l_2-norm regularization. Another important type of regularization (or corresponding prior) is the l_1-norm [43, 112–114]. Total variation regularization [59, 115] can be seen as a l_1-norm regularization of the gradient magnitude field. These approaches can be combined or generalized to form other regularization methods, such as the general-form Tikhonov regularization.

To balance the fit and the regularization and obtain a meaningful reconstruction it is necessary to make a proper selection of the regularization parameter. This can be complicated and there exist several approaches. In the case of (approximately) known noise norm, the discrepancy principle [116] tries to select the regularization parameter such that the norm of the evaluated fit is of the order of the noise norm. There also exist methods for unknown noise norm. The generalized cross-validation [117, 118] tries to minimize the (statistical) expected fit to the noise free observation (the prediction error). The L-curve method [119] is based on a log-log plot of the fit and solution norm for a range of regularization parameters and it is advocated to select the regularization parameter corresponding to the point of maximum curvature. Note that in some cases the regularization parameter is implicitly given, such as for multiple-input multiple-output detection using the minimum mean squared error measure. In this case the resulting problem is a Tikhonov regularized problem with the inverse signal-to-noise-ratio as the regularization parameter.

2 Sparse Signal Processing

Sparse methods are important in modern signal processing and have been applied to different applications such as transform coding [120, 121], estimation [122], linear prediction of speech [123], blind source separation [124] and denoising

[45, 46]. Compressed/compressive sensing [125, 126] is a method which can be used in a variety of applications. See [127] for an overview on sparse signal processing. The use of sparse methods in signal processing requires that the signal of interest can be represented, in *e.g.*, a basis, where the representation is sparse or almost sparse. Further, maintaining the most *significant* components in the representation yields accurate approximations of the original signal. Notable methods include the Fourier, cosine, wavelet and Gabor representations.

An important problem in sparse signal processing is how to incorporate the use of a sparse representation into an appropriate algorithm and/or problem formulation. Many attempts have been proposed, including greedy algorithms, convex relaxation, non-convex optimization and exhaustive search/brute force – we will shortly review the two first attempts since they are the most common.

Greedy algorithms are iterative methods which in each iteration perform a locally optimal choice. The basic greedy algorithm for sparse estimation is the matching pursuit (MP) algorithm [128, 129]. Attempts to offset the suboptimal performance of MP are algorithms such as orthogonal matching pursuit (OMP) [130–132] which includes a least-squares minimization over the support in each iteration and for compressive sensing the compressive sampling matching pursuit (CoSaMP) [133] which can extend and prune the support by more than one index in each iteration.

Convex relaxation is an approach to model difficult or high worst-case complexity optimization problems to a more convenient convex optimization problem. A minimum cardinality problem can be relaxed to a minimum l_1-norm problem. In fact, this is the closest convex approximation in case of equally bounded coefficients, or in a statistical framework, equally distributed coefficients. Minimum l_1-norm problems can also result from Bayesian estimation with Laplacian prior. Notable approaches in l_1-norm minimization are formulations such as least absolute shrinkage and selection operator (LASSO) [134], basis pursuit (denoising) (BPDN) [135] and the Dantzig selector [136].

Many of the above convex problem formulations refer to both the constrained and unconstrained Lagrange formulations of the same problems, since the problems are equivalent [137, Thm. 27.4]. For comparison among constrained and unconstrained problems, the relation between the regularization in constrained and unconstrained form can be found to obtain a reliable comparison, see [64, 89] for some total variation problems. However, in this case we assume that the regularization parameter in constrained and unconstrained form are equivalently easy to obtain – but it is argued that the regularization parameter in constrained form is more convenient, see *e.g.*, the discussions in [62, 76].

3 Multiple Descriptions

Multiple descriptions (MDs) is a method to enable a more stable transmission scheme by exploiting channel diversity. The idea is to divide a description of a signal into multiple descriptions and send these over separate erasure channels [138]. Only a subset of the transmitted descriptions are received which then can be decoded. The problem is then to construct the descriptions such that they individually provide an acceptable approximation of the source and furthermore are able to refine each other. Notice the contradicting requirements associated with the MD problem; in order for the descriptions to be individually good, they must all be similar to the source and therefore, to some extent, the descriptions are also similar to each other. However, if the descriptions are the same, they cannot refine each other. This is the fundamental trade-off of the MD problem. This is different from successive refinement, where one of the descriptions forms a base layer, and the other description forms a refinement layer, which is no good on its own.

A simple MD setup with two channels is given in Fig. 2.1 with the descriptions z_1, z_2, rate function $R(\cdot)$ and the decoding functions $g_1(\cdot), g_2(\cdot), g_{\{1,2\}}(\cdot)$.

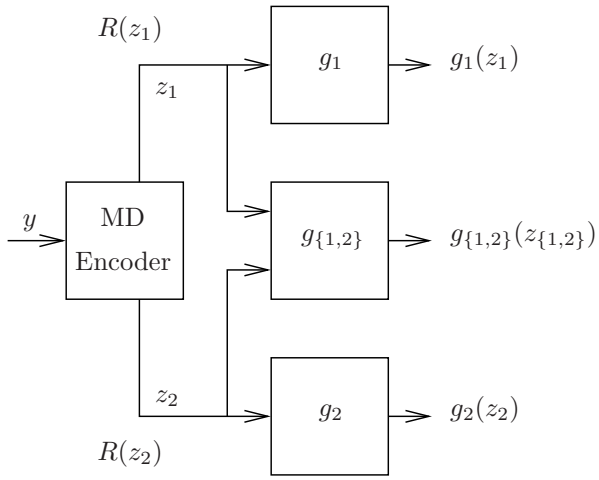


Fig. 2.1: The MD problem for two channels.

The application of MD is in the area of communication over unreliable channels where the erasure statistics of the channels are sufficiently independent such that we can explore channel diversity [139, 140]. We must also accept various quality levels and the different quality levels are distinguishable such that central

decoding is more valuable than side decoding, which is more valuable than no decoding [139]. Finally, we must have close to real-time play-out requirement such that retransmission and excess receiver side buffering is impossible [140]. These conditions can occur for real-time/two-way speech, audio and video applications.

Traditionally MD approaches try to characterize the rate-distortion region in a statistical setup [138]. The MD region in the case of two descriptions, Euclidean fidelity criterion and Gaussian sources is known and the bound is tight [141]. For the general J-channel case, $J \geq 2$, a MD achievable region is known but it is not known if these bounds are tight [142, 143].

Deterministic MD encoders are also known for speech [144, 145], audio [146, 147] and video [148–151] transmission. The MD image coding problem is also studied [152–155]. These approaches are specific for the certain application, but more general MD encoders exist. The MD scalar quantizer [156] for $J = 2$ channels which uses overlapping quantization regions. There are different approaches to MD vector quantization such as MD lattice vector quantization [157–159]. The MD l_1-compression formulation presented in paper D and E of this thesis can also be applied to a range of applications.

Chapter 3

Contributions

The applications presented in this thesis are broad, as also indicated in the title of the thesis and in the previous chapters. The unifying concept is the use of (optimal) first-order methods for convex optimization problems. Paper A is on a technique for addressing the trade-off between smoothness and accuracy using a smoothing technique for non-smooth functions. In paper B and C we design algorithms and software for solving known total variation problems. Paper D and E is on a new formulation of the MD problem.

Paper A: In this paper, we consider the problem of minimizing a non-smooth, non-strongly convex function using first-order methods. We discuss restart methods and the connection between restart methods and continuation. We propose a method based on applying a smooth approximation to an optimal first-order method for smooth problems and show how to reduce the smoothing parameter in each iteration. The numerical comparison show that the proposed method requires fewer iterations and an empirical lower complexity than reference methods.

Paper B: This paper describes software implementations for total variation image reconstruction in constrained form. The software is based on applying a smooth approximation of the non-smooth total variation function to an optimal first-order method for smooth problems. We use rank-reduction for ill-conditioned image deblurring to improve speed and numerical stability. The software scales well in the dimensions of the problem and only the regularization parameter needs to be specified.

Paper C: In this paper we discuss the implementation of total variation tomography regularized least-squares reconstruction in Lagrangian form with box-constraints. The underlying optimal first-order method for smooth

and strongly convex objective requires the knowledge of two parameters. The Lipschitz constant of the gradient is handled by the commonly used backtracking procedure and we give a method to handle an unknown strong convexity parameter and provide worst-case complexity bounds. The proposed algorithm is competitive with state-of-the-art algorithms over a broad class of problems and superior for difficulty problems, *i.e.*, ill-conditioned problems solved to high accuracy. These observations also follow from the theory. Simulations also shows, that in the case of problems that are not strongly convex, in practice the proposed algorithm still achieves the favourable convergence rate associated with strong convexity.

Paper D: In this paper, we formulate a general multiple-description framework based on sparse decompositions. The convex formulation is flexible and allows for non-symmetric distortions, non-symmetric rates, different decoding dictionaries and an arbitrary number of descriptions. We focus on the generated sparse signals, and conclude that the obtained descriptions are non-trivial with respect to both the cardinality and the refinement.

Paper E: We extend the work in D, by elaborating more on the issue of partially overlapping information corresponding to enforcing coupled constraints, and discuss the use of non-symmetric decoding functions. We show how to numerical solve large-scale problems and describe the solution set in terms of (non)-trivial instances. The sparse signals are encoded using a set partitioning in hierarchical trees (SPIHT) encoder and compared with state-of-the-art MD encoders. We also give examples for both video and images using respectively two and three channels.

References

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[2] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Methods in Convex Programming*. SIAM, 1994.

[3] A. Cauchy, "Méthode générale pour la résolution des systemés d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, pp. 536–538, 1847.

[4] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Stand.*, vol. 49, no. 6, pp. 409–436, Dec. 1952.

[5] W. C. Davidson, "Variable metric method for minimization," AEC Res. and Dev. Report ANL-5990 (revised), Tech. Rep., 1959.

[6] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization," *Comput. J.*, vol. 6, no. 2, pp. 163–168, August 1963.

[7] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms: 2. the new algorithm," *IMA J. Appl. Math.*, vol. 6, no. 3, pp. 222–231, Sep. 1970.

[8] R. Fletcher, "A new approach to variable metric algorithms," *Comput. J.*, vol. 13, no. 3, pp. 317–322, Mar. 1970.

[9] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Math. Comput.*, vol. 24, no. 109, pp. 23–26, Jan. 1970.

[10] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Math. Comput.*, vol. 24, no. 111, pp. 647–656, Jul. 1970.

[11] B. T. Polyak, *Introduction to Optimization*. Optimization Software, New York, 1987.

[12] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, pp. 373–395, 1984.

[13] Y. Nesterov and A. Nemirovskii, "A general approach to polynomial-time algorithms design for convex programming," Centr. Econ. and Math. Inst., USSR Acad. Sci., Moscow, USSR, Tech. Rep., 1988.

[14] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM J. Optim.*, vol. 2, pp. 575–601, 1992.

[15] Y. Nesterov, *Introductory Lectures on Convex Optimization, A Basic Course*. Kluwer Academic Publishers, 2004.

[16] A. S. Nemirovskii and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, Ltd., 1983.

[17] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Dokl. AN SSSR (translated as Soviet Math. Docl.)*, vol. 269, pp. 543–547, 1983.

[18] A. Nemirovskii and Y. Nesterov, "Optimal methods of smooth convex minimization," *USSR Comput.Maths.Math.Phys.*, vol. 25, pp. 21–30, 1985.

[19] Y. Nesterov, "On an approach to the construction of optimal methods of minimization of smooth convex functions," *Ekonom. i. Mat. Mettody*, vol. 24, pp. 509–517, 1988.

[20] J. Barzilai and J. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, pp. 141–148, 1988.

[21] Y.-H. Dai and L.-Z. Liao, "R-linear convergence of the Barzilai and Borwein gradient method," *IMA J. Numer. Anal.*, vol. 22, no. 1, pp. 1–10, 2002.

[22] R. Fletcher, "Low storage methods for unconstrained optimization," in *Computational Solution of Non-linear Systems of Equations*, E. L. Ellgower and K. Georg, Eds. Amer. Math. Soc., Providence, 1990, pp. 165–179.

[23] Y. Nesterov, "Smooth minimization of nonsmooth functions," *Math. Program. Series A*, vol. 103, pp. 127–152, 2005.

[24] ——, "Gradient methods for minimizing composite objective function," Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007, no 2007076, CORE Discussion Papers.

[25] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, 2009.

[26] G. Lan, "An optimal method for stochastic composite optimization," 2010, *Math. Program., Ser. A*, DOI:10.1007/s10107-010-0434-y.

[27] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," submitted to *SIAM J. Optim.*, 2008.

[28] G. Korpelevich, "The extragradient method for finding saddle points and other problems," *Ekonom. i. Mat. Mettody (In russian, english translation in Matekon)*, vol. 12, pp. 747–756, 1976.

[29] A. Nemirovskii, "Prox-method with rate of convergence $O(1/T)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems," *SIAM J. Optim.*, vol. 25, pp. 229–251, 2005.

[30] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.

[31] L. Vandenberghe, "Optimization methods for large-scale systems," Lecture notes, Electrical Engineering, University of California, Los Angeles (UCLA), available at http://www.ee.ucla.edu/~vandenbe/ee236c.html, 2009.

[32] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, 2007.

[33] G. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Oper. Res.*, vol. 8, no. 1, pp. 101–111, Jan.-Feb. 1960.

[34] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imaging Vision*, vol. 20, pp. 89–97, 2004.

[35] P. Weiss, G. Aubert, and L. Blanc-Féraud, "Efficient schemes for total variation minimization under constraints in image processing," *SIAM J. Sci. Comput.*, vol. 31, pp. 2047–2080, 2009.

[36] X. Lin, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, pp. 1136–1144, 2004.

[37] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2254–2268, 2007.

[38] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1192–1206, 2008.

[39] R. T. Rockafellar and R. J. B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Math. Oper. Res.*, vol. 16, no. 1, pp. 119–147, 1991.

[40] J. J. Moreau, "Proximitéet dualité dans un espace hilbertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[41] A. Chambolle, R. A. DeVore, N.-Y. Lee, and B. J. Lucier, "Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Trans. Image Process.*, vol. 7, pp. 319–335, 1998.

[42] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2005.

[43] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, pp. 906–916, 2003.

[44] P. Combetttes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, pp. 1168–1200, 2005.

[45] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[46] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[47] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, pp. 1956–1982, 2010.

[48] R. J. Bruck, "On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in Hilbert space," *J. Math. Anal. Appl.*, vol. 61, pp. 159–164, 1977.

[49] G. B. Passty, "Ergodic convergence to a zero of the sum of monotone operators in hilbert space," *J. Math. Anal. Appl.*, vol. 72, pp. 383–390, 1979.

[50] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, pp. 2419–2434, 2009.

[51] ——, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Oper. Res. Lett.*, vol. 31, no. 3, pp. 167–175, May 2003.

[52] A. Auslender and M. Teboulle, "Interior gradient and proximal methods for convex and conic optimization," *SIAM J. Optim.*, vol. 16, no. 3, pp. 697–725, 2006.

[53] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares," *Pac. J. Optim.*, vol. 6, pp. 615–640, 2010.

[54] Y.-J. Liu, D. Sun, and K.-C. Toh, "An implementable proximal point algorithmic framework for nuclear norm minimization," Tech report, Department of Mathematics, National University of Singapore, 2009.

[55] S. Ji and J. Ye, "An accelerated gradient method for trace norm minimization," in *Proc. Annual Int. Conf. on Machine Learning (ICML)*, 2009, pp. 457–464.

[56] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. Bauschke, R. Burachnik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, Eds. Springer-Verlag, 2010.

[57] P. Weiss and L. Blanc-Féraud, "A proximal method for inverse problems in image processing," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2009, pp. 1374–1378.

[58] C. R. Vogel, *Computational Methods for Inverse Problems*. SIAM, 2002.

[59] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia, 2005.

[60] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, "First-order methods for sparse covariance selection," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 1, pp. 56–66, 2008.

[61] A. Gilpin, J. Peña, and T. Sandholm, "First-order algorithm with $O(\ln(1/\epsilon))$ convergence for ϵ-equilibrium in two-person zero-sum games," 2008, 23rd National Conference on Artificial Intelligence (AAAI'08), Chicago, IL.

[62] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imag. Sci.*, vol. 4, no. 1, pp. 1–39, 2011.

[63] S. Hoda, A. Gilpin, J. Peña, and T. Sandholm, "Smoothing techniques for computing Nash equilibria of sequential games," *Math. Oper. Res.*, vol. 35, pp. 494–512, 2010.

[64] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algo.*, vol. 53, pp. 67–92, 2010.

[65] G. Lan, Z. Lu, and R. D. C. Monteiro, "Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming," *Math. Prog. Ser. A*, vol. 126, no. 1, pp. 1–29, 2011.

[66] J. J. Fuchs, "Multipath time-delay detection and estimation," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 237–243, Jan. 1999.

[67] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Top. Sign. Proces.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[68] M. J. D. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, R. Fletcher, Ed. Academic Press, 1969, pp. 283–298.

[69] M. R. Hestenes, "Multiplier and gradient methods," *J. Optim. Theory Appl.*, vol. 4, pp. 303 – 320, 1969.

[70] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[71] R. T. Rockafellar, "A dual approach to solving nonlinear programming problems by unconstrained optimization," *Math. Program.*, vol. 5, no. 1, pp. 354–373, 1973.

[72] A. N. Iusem, "Augmented Lagrangian methods and proximal point methods for convex optimization," *Investigación Operativa*, vol. 8, pp. 11–50, 1999.

[73] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with application to compressed sensing," *SIAM J. Imag. Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

[74] X.-C. Tai and C. Wu, "Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model," in *Proc. of Second International Conference of Scale Space and Variational Methods in Computer Vision, SSVM 2009, Lecture Notes in Computer Science*, vol. 5567, 2009, pp. 502–513.

[75] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 323–343, 2009.

[76] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, 2011.

[77] T. F. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 370–375, 1998.

[78] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, pp. 248–272, 2008.

[79] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, 2010.

[80] J. M. Bioucas-Dias and M. A. T. Figueiredo, "Multiplicative noise removal using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1720–1730, 2010.

[81] D. Goldfarb and S. Ma, "Fast multiple splitting algorithms for convex optimization," 2009, submitted to *SIAM J. Optim.*

[82] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM, 1990.

[83] E. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1107–1130, 2008.

[84] T. L. Jensen, J. Østergaard, and S. H. Jensen, "Iterated smoothing for accelerated gradient convex minimization in signal processing," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2010, pp. 774–777.

[85] S. Becker, E. J. Candès, and M. Grant, "Templates for convex cone problems with applications to sparse signal recovery," 2011, to appear in Mathematical Programming Computation.

[86] G. Lan and R. D. C. Monteiro, "Iteration-complexity of first-order augmented Lagrangian methods for convex programming," 2009, submitted for publication.

[87] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, 1996.

[88] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE J. Sel. Top. Sign. Proces.*, vol. 1, no. 4, pp. 606–617, 2007.

[89] D. Goldfarb and W. Yin, "Second-order cone programming methods for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 27, pp. 622 – 645, 2005.

[90] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Math. Program. Series B*, pp. 249–277, 2003.

[91] M. Zhu, S. J. Wright, and T. F. Chan, "Duality-based algorithms for total-variation regularized image restoration," *Comput. Optim. Appl.*, vol. 47, no. 3, pp. 377–400, 2010.

[92] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 20, pp. 1964–1977, 1999.

[93] Z. Lu, "Primal-dual first-order methods for a class of cone programming," 2010, submitted.

[94] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2010, available at www.optimization-online.org.

[95] M. Baes, "Estimate sequence methods: Extensions and approximations," Institute for Operations Research, ETH, Zürich, Switzerland, 2009, available at www.optimization-online.org.

[96] A. d'Aspremont, "Smooth optimization with approximate gradient," *SIAM J. Opt.*, vol. 19, pp. 1171–1183, 2008.

[97] J. Fadili and G. Peyré, "Total variation projection with first order schemes," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 657–669, 2011.

[98] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, pp. 3133–3145, 2010.

[99] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, pp. 193–228, 1998.

[100] Z.-Q. Lou, "Applications of convex optimization in signal processing and digital communication," *Math. Prog.*, vol. 97, pp. 177 – 207, 2003.

[101] J. Dahl, "Convex problems in signal processing and communications," Ph.D. dissertation, Department of Communication Technology, Aalborg University, Denmark, 2003.

[102] Eds. D. P. Palomar and Y. C. Eldar, *Convex Optimization in Signal Processing and Communications.* Cambridge University Press, 2010.

[103] Eds. Y. C. Eldar, Z.-Q. Lou, W.-K. Ma, D. P. Palomar, and N. D. Sidiropoulos, "Convex optimization in signal processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 19 – 127, May 2010.

[104] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering.* SIAM, Philadelphia, 2006.

[105] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems - Numerical Aspects of Linear Inversion.* SIAM, 1998.

[106] E. Thiébaut, "Introduction to image reconstruction and inverse problems," in *Optics in Astrophysics*, R. Foy and F.-C. Foy, Eds. Springer, 2005.

[107] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Math. Dokl. (English translation of Dokl. Akad. Nauk. SSSR.)*, vol. 4, pp. 1035–1038, 1963.

[108] R. J. Hanson, "A numerical method for solving Fredholm integral equations of the first kind using singular values," *SIAM J. Numer. Anal.*, vol. 8, pp. 616–622, 1971.

[109] J. M. Varah, "On the numerical solution off ill-conditioned linear systems with applications to ill-posed problems," *SIAM J. Numer. Anal.*, vol. 10, pp. 257–267, 1973.

[110] Å. Björck and L. Eldén, "Methods in numerical algebra and ill-posed problems," 1979, report, LiTH-MAT-R33-1979, Dept. of Mathematics, Linköping University, Sweden.

[111] J. A. Scales and A. Gersztenkorn, "Robust methods in inverse theory," *Inverse Prob.*, vol. 4, pp. 1071 – 1091, 1988.

[112] J. Claerbout and F. Muir, "Robust modelling of erratic data," *Geophysics*, vol. 38, pp. 826–844, 1973.

[113] H. Taylor, S. Banks, and F. McCoy, "Deconvolution with the ℓ_1 norm," *Geophysics*, vol. 44, pp. 39–52, 1979.

[114] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[115] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1993.

[116] V. A. Morozov, "On the solution of functional equations by the method of regularization," *Soviet Math. Dokl.*, vol. 7, pp. 414 – 417, 1966.

[117] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[118] G. Wahba, "Practical approximate solutions to linear operator equations when the data are noisy," *SIAM J. Numer. Anal.*, vol. 14, pp. 651–667, 1979.

[119] P. C. Hansen and D. P. O'Leary, "The use of the L-curve in the regularization of discrete ill-posed problems," *SIAM J. Sci. Comput.*, vol. 14, no. 6, pp. 1487–1503, 1993.

[120] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec 1993.

[121] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, 1996.

[122] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, 2003.

[123] D. Giacobello, M. G. Christensen, J. Dahl, S. H. Jensen, and M. Moonen, "Sparse linear predictors for speech processing," in *Proc. Ann. Conf. Int. Speech Commun. Ass. (INTERSPEECH)*, Brisbane, Australia, Sep. 2008, pp. 1353–1356.

[124] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Comp.*, vol. 13, no. 4, pp. 863–882, 2001.

[125] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[126] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[127] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way.* Academic Press, 2009.

[128] S. Mallat and S. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. Signal Process.*, vol. 41, pp. 3397–3415, 1993.

[129] S. Qian and D. Chen, "Signal representation using adaptive normalized Gaussian functions," *Signal Process.*, vol. 36, pp. 329–355, 1994.

[130] S. Chen, S. A. Billings, and W. Lou, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[131] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. on Signal, Systems and Computers*, Nov. 1993, pp. 40–44.

[132] G. Davis, S. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *Opt. Eng.*, vol. 33, no. 7, pp. 2183–2191, 1994.

[133] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, May. 2009.

[134] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Stat. Soc., Ser. B*, vol. 58, pp. 267–288, 1994.

[135] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, Aug. 1998.

[136] E. Candès and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Ann. Stat.*, vol. 35, no. 6, pp. 2313–2351, December 2007.

[137] R. T. Rockafellar, *Convex Analysis.* Princeton Univ. Press, 1970.

[138] A. A. E. Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 28, no. 6, pp. 851 – 857, Nov. 1982.

[139] V. K. Goyal, "Multiple description coding: Compression meets the network," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 74–93, Sep. 2001.

[140] M. H. Larsen, "Multiple description coding and applications," Ph.D. dissertation, Multimedia Information and Signal Processing, Aalborg University, 2007.

[141] L. Ozarow, "On a source-coding problem with two channels and three receivers," *Bell Syst. Tech. J.*, vol. 59, pp. 1909 – 1921, Dec. 1980.

[142] R. Venkataramani, G. Kramer, and V. K. Goyal, "Multiple description coding with many channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2106–2114, 2003.

[143] R. Purit, S. S. Pradhan, and K. Ranchandram, "n-channel symmetric multiple descriptions-part ii: An achievable rate-distortion region," *IEEE Trans. Inf. Theory*, vol. 51, no. 4, pp. 1377–1392, 2005.

[144] A. Ingle and V. Vaishampayan, "DPCM system design for diversity systems with applications to packetized speech," *IEEE Trans. Speech Audio Processing*, vol. 3, no. 1, pp. 48–58, Jan. 95.

[145] W. Jiang and A. Ortega, "Multiple description speech coding for robust communication over lossy packet networks," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2000, pp. 444–447.

[146] R. Arean, J. Kovacevic, and V. K. Goyal, "Multiple description perceptual audio coding with correlating transforms," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 2, pp. 140–145, Mar. 2000.

[147] G. Schuller, J. Kovacevic, F. Masson, and V. K. Goyal, "Robust low-delay audio coding using multiple descriptions," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1014–1024, Sep. 2005.

[148] T. Nguyen and A. Zakhor, "Matching pursuits based multiple description video coding for lossy environments," in *Proc. Int. Conf. Image Process. (ICIP)*, Barcelona, Spain, Sep. 2003, pp. 57–60.

[149] H. Chan and C. Huang, "Multiple description and matching pursuit coding for video transmission over the internet," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Hong Kong, Apr. 2003, pp. 425–428.

[150] Y. Wang, A. R. Reibman, and S. Lin, "Multiple description coding for video delivery," in *Proc. IEEE*, vol. 93, 2005, pp. 57 – 70.

[151] Y. Zhang, S. Mei, Q. Chen, and Z. Chen, "A multiple description image/video coding method by compressed sensing theory," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, Seattle, Washington, May. 2008, pp. 1830–1833.

[152] P. A. Chou, S. Mehrotra, and A. Wang, "Multiple description decoding of overcomplete expansions using projections onto convex sets," in *Proc. IEEE Data Comp. Conf. (DCC)*, Snowbird, Utah, Mar. 1999, pp. 72 – 81.

[153] T. Petrisor, B. Pesquet-Popescu, and J.-C. Pesquet, "A compressed sensing approach to frame-based multiple description coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Honolulu, Hawaii, Apr. 2007, pp. 709–712.

[154] T. Tillo, M. Grangetto, and G. Olmo, "Multiple description image coding based on Lagrangian rate allocation," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 673–683, 2007.

[155] U.-S. G. Sun, J. Liang, C. Tian, C. Tu, and T.-D. Tran, "Multiple description coding with prediction compensation," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1037–1047, 2009.

[156] V. A. Vaishampayan, "Design of multiple description scalar quantizers," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 821–834, 1993.

[157] S. D. Servetto, V. A. Vaishampayan, and N. J. A. Sloane, "Multiple description lattice vector quantization," in *Proc. IEEE Data Comp. Conf. (DCC)*, 1999, pp. 13–22.

[158] V. A. Vaishampayan, N. J. A. Sloane, and S. D. Servetto, "Multiple-description vector quantization with lattice codebooks: design and analysis," *IEEE Trans. Inf. Theory*, vol. 47, no. 5, pp. 1718–1734, 2001.

[159] J. Østergaard, R. Heusdens, and J. Jensen, "n-channel asymmetric entropy-constrained multiple-description lattice vector quantization," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6354–6375, Dec. 2010.

Paper A

Iterated Smoothing for Accelerated Gradient Convex Minimization in Signal Processing

T. L. Jensen, J. Østergaard and S. H. Jensen

This paper is published in
Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP),
Dallas, Texas, pp. 774–777, May 2010.

The layout is revised.

Minor spelling, gramma and notation errors have been corrected.

Abstract

In this paper, we consider the problem of minimizing a non-smooth convex problem using first-order methods. The number of iterations required to guarantee a certain accuracy for such problems is often excessive and several methods, e.g., restart methods, have been proposed to speed-up the convergence. In the restart method a smoothness parameter is adjusted such that smoother approximations of the original non-smooth problem are solved in a sequence before the original, and the previous estimate is used as the starting point each time. Instead of adjusting the smoothness parameter after each restart, we propose a method where we modify the smoothness parameter in each iteration. We prove convergence and provide simulation examples for two typical signal processing applications, namely total variation denoising and ℓ_1-norm minimization. The simulations demonstrate that the proposed method require fewer iterations and show lower complexity compared to the restart method.

1 Introduction

Recently there has been a renewed interest in optimal first-order methods even though these methods have been known for some time [1, 2], see also [3] for a unified framework. The inspiration for the current interest in first-order methods appears to come from a recent method that guarantee linear complexity for non-smooth problems with certain structures [4].

The motivation for using first-order methods is usually in the case of large scale problems, where second-order methods might scale poorly or problems where moderate accuracy of the solution is sufficient. Such problems occur in image processing [5, 6], but also compressed sensing recovery applies first-order methods [7–9]. These methods have also been used for robust numerical software packages [9, 10].

One method to minimize a non-smooth function is by minimizing a smooth approximation of the original non-smooth function. The effectiveness of such an approach is dependent upon the choice of a smoothness parameter, which also determines the accuracy of the smooth approximation. A large smoothness parameter yields a very smooth problem and in the early iterations of the algorithm, the function value will quickly decrease. However, the algorithm might not converge because the smooth approximation is not accurate enough. On the other hand, a sufficiently small smoothness parameter, gives a less smooth but more accurate approximation. In this case the function value will slowly decrease but convergence within the required accuracy is guaranteed. To decrease the number of iterations, and thereby speed up the algorithm, one may use restart methods. The idea is to combine the fast decreasing function value

in the early iterations for a very smooth problem with a sufficiently well approximated smooth function to ensure convergence in the final iterations. The algorithm starts by solving a much smoother problem than the original problem and then subsequently solve lesser smooth problems, using the previous estimate as the starting point at each restart, see [9, 11] and references therein. Such an approach is considered a heuristic except for the case of strongly convex functions where there are interesting theoretical results [11], or [12, §5.1.2] for composite objective functions.

In this paper, we will consider convex (but not strongly convex) non-smooth functions. For this case the results indicate that continuation or restart are practical efficient methods to decrease the number of iterations and yet reach the required accuracy [9]. We first review the restart method [9, 11] and relate this approach to the continuation method, see [7, 13] and references therein. We also demonstrate via simulations that restart methods reduce the complexity compared to an approach with a fixed smoothness parameter. Then, inspired by [7, 9, 11, 13] we propose a new method where we decrease the smoothness parameter in each iteration and prove that it converges. Our bound is, however, loose and the actual complexity is in practice much better than what the bound suggests. Simulation examples for two typical signal processing applications, namely total variation denoising and ℓ_1-norm minimization, show that the proposed method yield lower complexity compared to both the fixed smoothing approach and the restart approach.

2 A Smoothing Method

Let us consider the following optimization problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in Q_{\mathrm{p}} \end{array} \tag{A.1}$$

with the dual problem

$$\begin{array}{ll} \text{maximize} & g(u) \\ \text{subject to} & u \in Q_{\mathrm{d}} \end{array} \tag{A.2}$$

where f is a non-smooth, non-strongly, convex function and Q_{p}, Q_{d} are convex sets. Let $x^\star$ and $u^\star$ be solutions to the problems (A.1) and (A.2), respectively. The complexity estimate for problem (A.1) is $\mathcal{O}(1/\epsilon^2)$, where ϵ is the accuracy of the objective value

$$f(x) - f(x^\star) \leq f(x) - g(u) \leq \epsilon, \quad x \in Q_{\mathrm{p}},\, u \in Q_{\mathrm{d}}\,.$$

In [4] it was, however, shown that for problems with certain structures it is possible to obtain the complexity $\mathcal{O}(1/\epsilon)$, which is one order faster than the

sub-gradient method. The idea is to exploit the structure of the non-smooth problem. This is done by making a smooth approximation of the non-smooth function and then subsequently minimize the smooth approximation using an optimal first-order method for the class of smooth problems.

In the following we review the steps required for approximating a non-smooth function by a smooth function. A more general approach is given in [4], but this reduced form will be sufficient for our simulations in Sec. 5. Let the function f have the form

$$f(x) = \max_{u \in Q_{\mathrm{d}}} u^T A x \,, \tag{A.3}$$

where we now assume Q_{d} is a closed and bounded convex set. We then approximate f by f_μ where

$$f_\mu(x) = \max_{u \in Q_{\mathrm{d}}} u^T A x - \mu d_{\mathrm{d}}(u) \,,$$

with $\mu > 0$ called the smoothness parameter and $d_{\mathrm{d}}(u) \geq \frac{1}{2}\|u - \hat{u}\|_2^2$. The function f_μ satisfy

$$f_\mu(x) \leq f(x) \leq f_\mu(x) + \mu \Delta_{\mathrm{d}} \,, \quad \Delta_{\mathrm{d}} = \max_{u \in Q_{\mathrm{d}}} d_{\mathrm{d}}(u) \,. \tag{A.4}$$

The approximation function is also smooth, *i.e.*, it has Lipschitz continuous gradient

$$\|\nabla f_\mu(x) - \nabla f_\mu(\tilde{x})\|_2 \leq L_\mu \|x - \tilde{x}\|_2, \quad x, \tilde{x} \in Q_{\mathrm{p}}$$

with

$$L_\mu = \frac{\|A\|_2^2}{\mu} \,. \tag{A.5}$$

It was shown in [4] that the optimal selection of a fixed μ for achieving an ϵ-accuracy is

$$\mu = \frac{\epsilon}{2\Delta_{\mathrm{d}}} \,, \tag{A.6}$$

which results in an $\epsilon/2$ approximation of f, *i.e.*,

$$f_\mu(x) \leq f(x) \leq f_\mu(x) + \frac{\epsilon}{2} \,. \tag{A.7}$$

We now apply the smooth approximation to an optimal first-order method with complexity estimate $\mathcal{O}(\sqrt{L/\epsilon})$ [14] where L is the Lipschitz constant of the gradient of the objective function. Using (A.5) and (A.6) we obtain the complexity $\mathcal{O}(1/\epsilon)$ for non-smooth problems.

3 Restart

As indicated in (A.6), a fixed μ is selected so small that the approximation accuracy in (A.7) would be smaller than the required accuracy. Another approach is to select μ large in the early iterations because the smooth approximation converges like $\mathcal{O}(\sqrt{L_\mu/\epsilon})$ and then in the final iterations select μ small enough to ensure the smooth approximation comes within the required accuracy. This idea is used in [9, 11], where the main algorithm is restarted several times with first a large μ, and then subsequently a smaller and smaller μ. Note that in [7, 13], they solve composite problems of the form

$$\min_x \psi(x) + \frac{1}{\mu} h(x)\,,$$

where $h(x)$ is smooth and $\psi(x)$ is non-smooth. The smoothness, or the Lipschitz constant of $\frac{1}{\mu}\nabla h(x)$, is $L = \frac{1}{\mu}L(\nabla h(x))$ where $L(f)$ is the Lipschitz constant of the function f. For small μ we will then have a large Lipschitz constant of the gradient function. The continuation idea in [7, 13] is then similar to the restart approaches in [9, 11] because the sequence of problems solved in the continuation strategy becomes less and less smooth, as in the restart approach.

For strongly convex functions, it is possible to guarantee that the previous estimate is useful as a starting point, *i.e.*, warm start, and then show the advantage of applying a restart method. Let ϕ be a strongly convex max-type function with strong convexity parameter σ, but ϕ does not have a Lipschitz continuous gradient. We then have [14, Corollary 2.3.1]

$$\frac{\sigma}{2}\|y - y^\star\|_2^2 \leq \phi(y) - \phi(y^\star), \quad y \in Q$$

where $y^\star$ is the solution that minimize $\phi(y)$ for $y \in Q$. It was shown in [11] that the restart algorithm has the complexity $\mathcal{O}(1/\log(\epsilon))$ for strongly convex non-smooth functions. Warm start approaches for first-order methods are also studied in [15] and in [12, §5.1.2]. The restart algorithm from [11] is given below.

The function **NESTEROV** is not shown, but is the algorithm presented in [4, §3.11], which outputs a primal and dual $\bar{\epsilon}_j$-optimal solution after $\bar{k}^{(j+1)}$ iterations with the starting point $\bar{x}^{(j)}$ and using the smoothness parameter $\mu = \frac{\bar{\epsilon}_j}{2\bar{\Delta}_{\mathrm{d}}}$.

4 Iterated Smoothing

In the previous section we reviewed a restart algorithm where the smoothness parameter was decreased before a restart. The idea proposed in this section is to

Algorithm: **Restart** [11]

Given a $\bar{x}^{(0)}$,$\bar{u}^{(0)}$, $\gamma > 0$, $k = 0$ and ϵ

Repeat for $j = 0, 1, 2, \ldots$

$\bar{\epsilon}_j = \max\left(\frac{f(\bar{x}^{(j)}) - g(\bar{u}^{(j)})}{\gamma}, \epsilon\right)$

$\bar{x}^{(j+1)}, \bar{u}^{(j+1)}, \bar{k}^{(j+1)} = \mathbf{NESTEROV}(\bar{x}^{(j)}, \bar{\epsilon}_j)$

$k = k + \bar{k}^{(j+1)}$

if $f(\bar{x}^{(j+1)}) - g(\bar{u}^{(j+1)}) \leq \epsilon$ **then break**

decrease the smoothness parameter in each iteration instead of only after each restart, using an optimal first-order method as base.

We will study the convergence properties of such an algorithm. Let $\{(x^{(j)}, y^{(j)}, z^{(j)}, \theta_j)\}$ be generated by Algorithm 1 or Algorithm 2 from [3], and use the smooth approximation $f_{\mu_j}(x)$ (with a variable smoothness parameter μ_j). We then have

$$\begin{aligned} f_{\mu_j}(x^{(j+1)}) \leq\ & (1-\theta_j) f_{\mu_j}(x^{(j)}) + \theta_j f_{\mu_j}(x^\star) \\ & + \theta_j^2 L_{\mu_j} (\tfrac{1}{2}\|x^\star - z^{(j)}\|_2^2 - \tfrac{1}{2}\|x^\star - z^{(j+1)}\|_2^2) \end{aligned}$$

for the iterations $j = 0, 1, ...$ Using the approximation in (A.4), we obtain

$$\begin{aligned} f(x^{(j+1)}) - \mu_j \Delta_{\mathrm{d}} \leq & (1-\theta_j) f(x^{(j)}) + \theta_j f(x^\star) \\ & + \theta_j^2 L_{\mu_j} (\tfrac{1}{2}\|x^\star - z^{(j)}\|_2^2 - \tfrac{1}{2}\|x^\star - z^{(j+1)}\|_2^2). \end{aligned}$$

With $\theta_k = \frac{2}{k+2}$, we select $\mu_j = \alpha\theta_j^2$ as a quadratically decreasing function. This will ensure that the approximation error converges to a constant. We then obtain

$$\begin{aligned} & f(x^{(j+1)}) - f(x^\star) - (1-\theta_j)(f(x^{(j)}) - f(x^\star)) \\ & \quad \leq \frac{\|A\|_2^2}{\alpha} \left(\tfrac{1}{2}\|x^\star - z^{(j)}\|_2^2 - \tfrac{1}{2}\|x^\star - z^{(j+1)}\|_2^2 \right) + \theta_j^2 \alpha \Delta_{\mathrm{d}} \,. \end{aligned}$$

Adding the inequalities from $j = 0, 1, \ldots, k-1$, gives

$$\begin{aligned} & f(x^{(k)}) - f(x^\star) + \sum_{j=1}^{k-1} \theta_j (f(x^{(j)}) - f(x^\star)) \\ & \quad \leq \frac{\|A\|_2^2}{\alpha} (\tfrac{1}{2}\|x^\star - z^{(0)}\|_2^2 - \tfrac{1}{2}\|x^\star - z^{(k)}\|_2^2) + \sum_{j=0}^{k-1} \theta_j^2 \alpha \Delta_{\mathrm{d}} \,. \end{aligned}$$

We then obtain the lower bound

$$f(x^{(k)}) - f(x^\star) + \sum_{j=1}^{k-1} \theta_j (f(x^{(j)}) - f(x^\star)) \\ \geq \min_{i=1,\dots,k} \Big\{ f(x^{(i)}) - f(x^\star) \Big\} \Big(1 + \sum_{j=1}^{k-1} \theta_j \Big) .$$

For $\theta_k = \frac{2}{k+2}$, we have

$$\sum_{j=1}^{k-1} \theta_j \geq 2 \log_{\mathrm{e}}(k+1) - 3, \quad \sum_{j=0}^{k-1} \theta_j^2 \leq \sum_{j=0}^{\infty} \theta_j^2 = \frac{2}{3}\pi^2 - 4 .$$

It is important that the sum of the approximation errors is bounded by a constant. This is achieved for quadratically decreasing functions, which motivated our selection $\mu_j = \alpha \theta_j^2$. For $k \geq 2$,

$$\min_{i=1,\dots,k} \Big\{ f(x^{(i)}) - f(x^\star) \Big\} \\ \leq \frac{1}{2\log_{\mathrm{e}}(k+1) - 2} \left(\frac{\|A\|_2^2}{2\alpha} \|x^\star - x^{(0)}\|_2^2 + \alpha \Delta_{\mathrm{d}} \left(\frac{2}{3}\pi^2 - 4 \right) \right) .$$

The algorithm converges, although the upper bound decreases slowly. The parameter α works as a tradeoff between the two terms in the brackets. Since $\|x^\star - x^{(0)}\|_2^2$ is unknown in practice and the bound above is loose, we are instead inspired by (A.6) and set

$$\alpha = \frac{f(x^{(0)}) - g(u^{(0)})}{2\Delta_{\mathrm{d}} c} ,$$

where c is a scaling reflecting that $g(u^{(0)})$ might severely underestimate $f(x^\star)$. The algorithm **Smooth** implements the iteratively decreasing smoothness parameter studied in this section and is applied to Algorithm 1 in [3], with the smoothing technique presented in [4]. The function $P_Q(x)$ is the projection of x onto Q,

$$P_Q(x) = \operatorname*{argmin}_{y \in Q} \|x - y\|_2^2 .$$

5 Simulations

In this section, we compare the three algorithms, **Fixed** (as in [4, §3.11] with a fixed μ selection), **Restart** and **Smooth** for solving two different problems on

the form (A.1) and (A.3). For algorithms **Fixed** and **Smooth** we record the number of iterations k required to reach the duality gap

$$f(x^{(k)}) - g(u^{(k)}) \leq \epsilon, \quad x^{(k)} \in Q_{\mathrm{p}},\, u^{(k)} \in Q_{\mathrm{d}}$$

Algorithm: **Smooth**

Given a $x^{(0)}$, $u^{(0)}$ and ϵ, set $\alpha = \frac{f(x^{(0)}) - g(u^{(0)})}{2\Delta_{\mathrm{d}} c}$, $z^{(0)} = x^{(0)}$

Repeat for $k = 0, 1, 2, \ldots$

$$y^{(k)} = (1-\theta_k)x^{(k)} + \theta_k z^{(k)}, \quad \theta_k = \tfrac{2}{k+2}$$

$$\mu_k = \alpha\theta_k^2$$

$$\tilde{u}(y^{(k)}) = \operatorname*{argmax}_{u \in Q_{\mathrm{d}}} \left\{ u^T A y^{(k)} - \mu_k d_{\mathrm{d}}(u) \right\}$$

$$u^{(k)} = (1-\theta_k)u^{(k)} + \theta_k \tilde{u}(y^{(k)})$$

if $f(x^{(k)}) - g(u^{(k)}) \leq \epsilon$ **then break**

$$\nabla f_{\mu_k}(y^{(k)}) = A^T \tilde{u}(y^{(k)})$$

$$z^{(k+1)} = P_{Q_{\mathrm{p}}}\left(z_k - \tfrac{1}{\theta_k L_{\mu_k}} \nabla f_{\mu_k}(y^{(k)}) \right)$$

$$x^{(k+1)} = (1-\theta_k)x^{(k)} + \theta_k z^{(k+1)}$$

where $x^{(k)} \in \mathbb{R}^{N \times 1}$. For the algorithm **Restart**, we record the total number of inner iterations k. As primal and dual prox-function we use $d_{\mathrm{p}}(x) = \frac{1}{2}\|x - x^{(0)}\|_2^2$ and $d_{\mathrm{d}}(u) = \frac{1}{2}\|u\|_2^2$ as in [9] ($\hat{u} = 0$). By choosing the center of the primal prox-function as the starting point for the new iterations, we obtain a good initial/warm starting point in each restart [9]. For a fixed accuracy, it was suggested in [9] to use a fixed number of restarts. However, since we sweep over a large range of accuracies it is more appropriate to allow a variable number of restarts. We therefore select γ as suggested in [11].

5.1 Total Variation Denoising

Our first example is the total variation denoising problem [10, 16],

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} \|D_{ij}x\|_2 \\ \text{subject to} & \|x - b\|_2 \leq \delta \end{array}$$

where $D_{ij}x$ is an approximation of the gradient at pixel i, j, and m, n is the image dimensions with the number of variables $N = mn$. We observe $b = x_0 + e$

with x_0 the original image and e being i.i.d. Gaussian noise. As initialization we use $x^{(0)} = \bar{x}^{(0)} = b$ and

$$u^{(0)} = \bar{u}^{(0)} = \underset{u \in Q_{\mathrm{d}}}{\operatorname{argmax}}\, u^T A x^{(0)} - \frac{\epsilon}{2\Delta_{\mathrm{d}}} d_{\mathrm{d}}(u) \tag{A.8}$$

with $A = [D_{11}, D_{12}, \cdots, D_{mn}]$. For the total variation denoising problem we obtain the simulation results shown in Fig. A.1. We observe that the algorithm **Fixed** with fixed μ converges approximately linear $\mathcal{O}(1/\epsilon)$. If we, however, apply **Restart** then the algorithm is faster and the complexity is lower (the slope is closer to that of $\mathcal{O}(1/\sqrt{\epsilon})$ compared to $\mathcal{O}(1/\epsilon)$). The proposed approach **Smooth** with decreasing μ for each iteration converges faster and shows slightly better complexity, approximately $\mathcal{O}(1/\sqrt{\epsilon})$.

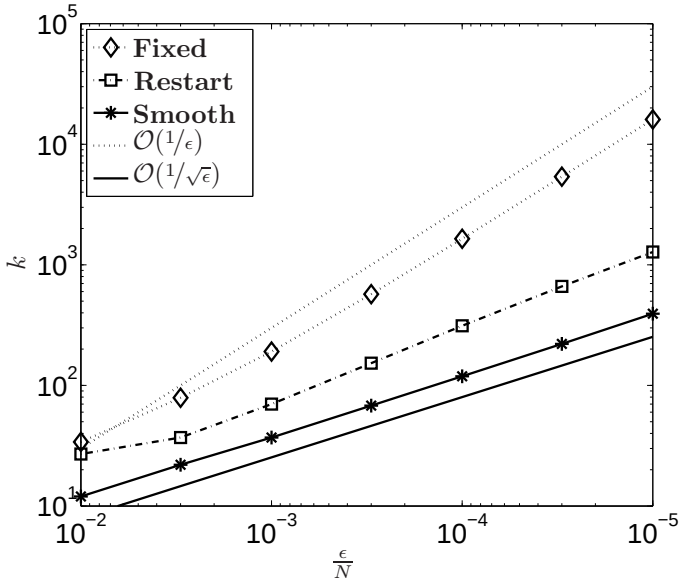


Fig. A.1: Simulation results for a total variation denoising example of a noisy image of Lenna (512×512). We report the number of iterations k required to reach the relative accuracy $\frac{\epsilon}{N}$. As a reference, we also show the complexity functions $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ and $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$.

5.2 ℓ_1-norm Minimization

For the second example, we will consider the problem of finding a sparse representation of an image b in an overcomplete dictionary B:

$$\begin{array}{ll} \text{minimize} & \|z\|_1 \\ \text{subject to} & \|Bz - b\|_2 \leq \delta \end{array} \tag{A.9}$$

where $B = [B_1; B_2]$ and B_1 is the 2-dimensional discrete cosine transform and B_2 is a Symlet16 wavelet transform with 3 levels. As shown in [17], the problem (A.9) can be posed as an equivalent problem with simpler projection constraints

$$\begin{array}{ll} \text{minimize} & \|Wx\|_1 \\ \text{subject to} & \|x_1 - b\|_2 \leq \delta \end{array}$$

$$\text{where} \qquad W = \begin{bmatrix} B_1 & -B_1 B_2^{-1} \\ 0 & I \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

We initialize the algorithms with $x^{(0)} = \bar{x}^{(0)} = [b; 0]$ and $u^{(0)} = \bar{u}^{(0)}$ as in (A.8) with $A = W$. For this problem we obtain the simulation results shown in Fig. A.2, where we again observe that the approach **Fixed** with fixed μ converges approximately linear $\mathcal{O}(1/\epsilon)$. For the **Restart** approach, the convergence rate is closer to $\mathcal{O}(1/\sqrt{\epsilon})$ for high accuracy (small ϵ) but approximately $\mathcal{O}(1/\epsilon)$ for low accuracy. The proposed algorithm **Smooth** with decreasing μ for each iteration converges faster and shows lower complexity than the other two methods. We also generated 100 problems with the vector $b \in \mathbb{R}^{128^2 \times 1}$ being i.i.d. Gaussian. For these simulations we observe similar results as reported in Fig. A.2 for the relative convergence speed and complexity.

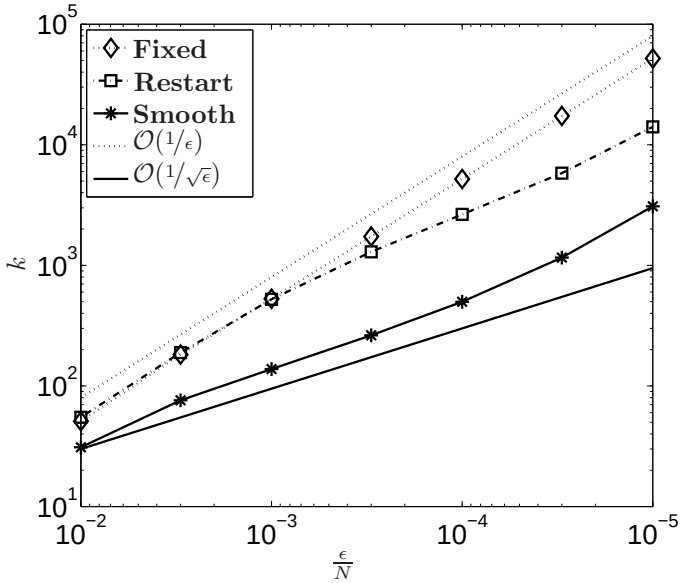


Fig. A.2: Simulation results for an ℓ_1-norm minimization example using the image of Lenna (512×512). We report the number of iterations k required to reach the relative accuracy $\frac{\epsilon}{N}$. As a reference, we also show the complexity functions $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ and $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$.

6 Conclusions

We presented a new method to speed up the convergence for non-smooth problems using accelerated gradient methods for convex minimization. We provided a proof of convergence, which resulted in a loose bound on the complexity. In fact, practical simulations revealed that the complexity is lower than what the bound suggests. For comparison, we studied and simulated existing methods. The simulations showed that the proposed method has both faster convergence and lower complexity compared to the existing methods.

References

[1] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Doklady AN SSSR (translated as Soviet Math. Docl.)*, vol. 269, pp. 543–547, 1983.

[2] ——, "On an approach to the construction of optimal methods of minimization of smooth convex functions," *Ékonom. i. Mat Metody.*, vol. 24, pp. 509–517, 1988.

[3] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," submitted to SIAM J. Optim., 2008.

[4] Y. Nesterov, "Smooth minimization of nonsmooth functions," *Math. Prog. Series A*, vol. 103, no. 127-152, 2005.

[5] P. Weiss, G. Aubert, and L. Blanc-Féraud, "Efficient schemes for total variation minimization under constraints in image processing," *SIAM J. Sci. Comput.*, vol. 31, pp. 2047–2080, 2009.

[6] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, 2009.

[7] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Top. Sign. Proces.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[8] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with application to compressed sensing," *SIAM J. Imag. Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

[9] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imag. Sci.*, vol. 4, no. 1, pp. 1–39, 2011.

[10] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algo.*, vol. 53, pp. 67–92, 2010.

[11] A. Gilpin, J. Peña, and T. Sandholm, "First-order algorithm with $O(\ln(1/\epsilon))$ convergence for ϵ-equilibrium in two-person zero-sum games," 2008, 23rd National Conference on Artificial Intelligence (AAAI'08), Chicago, IL.

[12] Y. Nesterov, "Gradient methods for minimizing composite objective functions," CORE Discussion Papers series, Université Catholique de Louvain, Center for Operations Research and Econometrics, Tech. Rep., 2007, available http://www.uclouvain.be/en-44660.html.

[13] W. Y. E. Hale and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularization with application to compressed sensing," 2007, CAAM Technical Report TR07-07, Rice University, Houston, Texas.

[14] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

[15] G. Lan and R. D. C. Monteiro, "Iteration-complexity of first-order augmented Lagrangian methods for convex programming," submitted, 2009.

[16] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, Philadelphia, 2005.

[17] J. Dahl, J. Østergaard, T. L. Jensen, and S. H. Jensen, "An efficient first-order method for ℓ_1 compression of images," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2009, pp. 1009–1012.

Paper B

Algorithms and Software for Total Variation Image Reconstruction via First-Order Methods

J. Dahl, P. C. Hansen, S. H. Jensen and T. L. Jensen

This paper is published in
Numerical Algorithms,
No. 53, pp. 67–92, 2010.

The layout is revised.

Minor spelling, gramma and notation errors have been corrected.

Abstract

This paper describes new algorithms and related software for total variation (TV) image reconstruction, more specifically: denoising, inpainting, and deblurring. The algorithms are based on one of Nesterov's first-order methods, tailored to the image processing applications in such a way that, except for the mandatory regularization parameter, the user needs not specify any parameters in the algorithms. The software is written in C with interface to Matlab (version 7.5 or later), and we demonstrate its performance and use with examples.

1 Introduction

Image reconstruction techniques have become important tools in computer vision systems and many other applications that require sharp images obtained from noisy and otherwise corrupted ones. At the same time the total variation (TV) formulation has proven to provide a good mathematical basis for several basic operations in image reconstruction [5], such as *denoising*, *inpainting*, and *deblurring*. The time is ripe to provide robust and easy-to-use public-domain software for these operations, and this paper describes such algorithms along with related Matlab and C software. To our knowledge, this is the first public-domain software that includes all three TV image reconstruction problems. The software is available from http://www.netlib.org/numeralgo in the file na28, the Matlab files have been tested on Matlab versions 7.5–7.8, and they require version 7.5 or later.

We note that some Matlab codes are already available in the public domain, see the overview in Table B.1. In §7 we compare the performance of our algorithms with those in Table B.1; such a comparison is not straightforward as these codes solve slightly different problems and do not use comparable stopping criteria. Our comparisons show that our algorithms indeed scale well for large-scale problems compared to the existing methods.

The optimization problems underlying the TV formulation of image restoration cannot easily be solved using standard optimization packages due to the large dimensions of the image problems and the non-smoothness of the objective function. Many customized algorithms have been suggested in the literature, such as subgradient methods [1, 7], dual formulations [4, 24], primal-dual methods [6, 16, 21], graph optimization [9], second-order cone programming [13], etc. However, the implementation of all these methods for large-scale problem is not straightforward.

Our algorithms are based on recently published first-order methods developed by Nesterov [17–20], but tailored specifically to the problems in image restoration that we consider. The new first-order methods have $O(1/\epsilon)$ complexity, where ϵ

is the accuracy of the solution. These methods show promising potential in large-scale optimization but have, so far, been used only scarcely for image processing algorithms – except for very recent work in [2] and [22].

Compared to [22], we provide practical complexity bounds and stopping criteria, we included inpainting into Nesterov's framework, and we use rank reduction to improve the speed and numerical stability of the deblurring algorithm. Our approach allows us to choose all necessary parameters in the algorithms in a suitable fashion, such that only the regularization parameter must be specified by the user. More experienced users can set additional parameters if needed. Our algorithms and implementations are robust, user friendly, and suited for large problems.

Our paper starts with a brief summary of the notation in §2. We then present our three methods for TV-based denoising, inpainting, and deblurring in §3–§5; the presentation follows that of Nesterov, but with a simplified notation tailored to our image processing applications. Next, in §6 we illustrate the use of our methods and software with three examples, and in §7 we demonstrate the performance and the computational complexity of our methods. Brief manual pages for the Matlab functions are given in the appendix.

2 Notation

In this package we consider $m \times n$ grayscale images, represented by the image arrays B (the noisy/corrupted image) and X (the reconstructed image). For our mathematical formulation it is convenient to represent the images by the two vectors x and b of length mn, given by

$$x = \mathrm{vec}(X), \qquad b = \mathrm{vec}(B),$$

where "vec" denotes column-wise stacking.

Associated with each pixel X_{ij} is a 2×1 gradient vector, and we approximate this gradient via finite differences. To set the notation, we first define two $m \times n$ arrays X'_{c} and X'_{r} with the finite-difference approximations to the partial derivatives in the directions of the columns and rows:

$$X'_{\mathrm{c}} = D_m\, X, \qquad X'_{\mathrm{r}} = X\, D_n^T,$$

where the two matrices D_m and D_n hold the discrete approximations to the derivative operators, including the chosen boundary conditions. Then we write the *gradient approximation* for pixel ij as the 2×1 vector

$$D_{(ij)}\, x = \begin{pmatrix} (X'_{\mathrm{c}})_{ij} \\ (X'_{\mathrm{r}})_{ij} \end{pmatrix} \in \mathbb{R}^{2\times 1}, \tag{B.1}$$

Table B.1: Freely available Matlab codes for TV reconstruction.

Code:	`tvdenoise` – denoising.
Author:	Pascal Getreuer, Dept. of Mathematics, UCLA, Los Angeles.
Comments:	Chambolle's algorithm [4] (dual formulation), stopping criterion, very fast, also treats color images.
Availability:	Matlab Central File Exchange: www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=16236
Code:	`perform_tv_denoising` – denoising.
Author:	Gabriel Peyré, CNRS, CEREMADE, Université Paris Dauphine.
Comments:	Chambolle's algorithm [4] (dual formulation), no stopping criterion, fast.
Availability:	Toolox – A Toolbox for General Purpose Image Processing: www.ceremade.dauphine.fr/∼peyre/matlab/image/content.html
Code:	`TVGP` – denoising.
Authors:	M. Zhu, Dept. of Mathematics, UCLA, Los Angeles. S. Wright, Dept. of Computer Sciences, Univ. of Wisconsin, Madison. T. F. Chan, Dept. of Mathematics, UCLA, Los Angeles.
Comments:	Gradient projection algorithm for the dual formulation, software, stopping criterion, very fast. Described in [24].
Availability:	TV-Regularized Image Denoising Software: www.cs.wisc.edu/∼swright/TVdenoising
Code:	`SplitBregmanROF` – denoising.
Authors:	Tom Goldstein and Stanley Osher, Dept. of Mathematics, UCLA, Los Angeles.
Comments:	Bregman iterations, C++ code with Matlab mex interface, stopping criterion, very fast. Described in [14].
Availability:	Split Bregman Denoising: www.math.ucla.edu/∼tagoldst/code.html
Code:	`tv_dode_2D` – inpainting.
Author:	Carola-Bibiane Schönlieb, Centre for Mathematical Sciences, Cambridge University, UK.
Comments:	Script with built-in stopping criterion, no interface, slow. Described in [11].
Availability:	Domain Decomposition for Total Variation Minimization: homepage.univie.ac.at/carola.schoenlieb/webpage_tvdode/tv_dode_numerics.htm
Code:	`Fixed_pt` and `Primal_dual` – deblurring.
Author:	Curtis R. Vogel, Dept. of Mathematical Sciences, Montana State University, Bozeman.
Comments:	Scripts with no stopping criteria or interface. Described in [21].
Availability:	Codes for the book *Computational Methods for Inverse Problems:* www.math.montana.edu/∼vogel/Book/Codes/Ch8/2d
Code:	`FTVdG` – deblurring.
Authors:	Junfeng Yang, Nanjing University, China. Yin Zhang, Wotao Yin, and Yilun Wang, Dept. of Computational and Applied Mathematics, Rice University, Houston.
Comments:	Script with stopping criteria, fast, treats color images. Described in [23].
Availability:	FTVd: A Fast Algorithm for Total Variation based Deconvolution. www.caam.rice.edu/∼optimization/L1/ftvd/v3.0

where the notation (ij) for the subscript denotes that we operate on the pixel with index ij, and $D_{(ij)}$ is a matrix of dimensions $2\times mn$. For one-sided difference approximations at the "inner pixels", we have

$$D_{(ij)} = \left[\; e_{i+1+(j-1)m} - e_{i+(j-1)m} \;,\; e_{i+jm} - e_{i+(j-1)m} \;\right]^T,$$

in which e_k denotes the kth canonical unit vector of length mn. We also define the matrix D (of dimensions $2mn \times mn$) obtained by stacking all the $D_{(ij)}$ matrices:

$$D = \begin{pmatrix} D_{(11)} \\ \vdots \\ D_{(mn)} \end{pmatrix}. \tag{B.2}$$

In Appendix B we show that the 2-norm of this matrix satisfies $\|D\|_2^2 \leq 8$. The approximation to the gradient norm satisfies $\|D_{(ij)}\, x\|_2^2 = (X'_{\mathrm{c}})_{ij}^2 + (X'_{\mathrm{r}})_{ij}^2$.

We also need to introduce the vector $u \in \mathbb{R}^{2mn}$ of *dual variables*, and similar to before we use the notation $u_{(ij)}$ for the 2-element sub-vector of u that conforms with Eq. (B.2) and corresponds to pixel ij.

The total variation (TV) of a function $f(s,t)$ in a domain Ω is defined as the 1-norm of the gradient magnitude, i.e., $\int_\Omega \|\nabla f\|_2 \, ds\, dt$ in which $\|\nabla f\|_2^2 = (\partial f/\partial s)^2 + (\partial f/\partial t)^2$. For our discrete problem, we define the analogous *discrete TV function* associated with the image X as

$$\mathcal{T}(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} \|D_{(ij)}\, x\|_2, \tag{B.3}$$

i.e., the sum of all the 2-norms of the gradient approximations.

In our algorithms we need to extract elements of a vector $x \in \mathbb{R}^N$ specified by an index-set $\mathcal{I} = \{i_1,\, i_2,\, \ldots,\, i_{|\mathcal{I}|}\}$ with indices i_k between 1 and N. Here, $|\mathcal{I}|$ denotes the number of elements in $\mathcal{I}$. If all the elements in $\mathcal{I}$ are distinct (i.e., $i_k \neq i_l$ when $k \neq l$), then the complementary set is $\mathcal{I}_c := \{1, \ldots, N\} \setminus \mathcal{I} = \{j_1,\, j_2,\, \ldots,\, j_{N-|\mathcal{I}|}\}$ again with indices j_k between 1 and N.

3 Denoising

Given a noisy image $B = X^{\text{exact}} + \text{noise}$, the discrete TV denoising problem amounts to minimizing $\mathcal{T}(x)$ subject to a constraint on the difference between the reconstruction x and the data b. This ensures that the reconstructed image is closely related to the noisy image, but "smoother" as measured by the TV function (B.3). The discrete TV denoising problem can thus be formulated as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m} \sum_{j=1}^{n} \|D_{(ij)}\, x\|_2 \\ \text{subject to} & \|x - b\|_2 \leq \delta, \end{array} \tag{B.4}$$

which is a *second-order cone programming problem* (SOCP) [3]. The dual problem is also a SOCP, given by

$$\begin{array}{ll} \text{maximize} & -\delta \, \|D^T u\|_2 + b^T D^T u \\ \text{subject to} & \|u_{(ij)}\|_2 \leq 1, \quad i = 1, \ldots, m, \; j = 1, \ldots, n, \end{array} \tag{B.5}$$

where $u \in \mathbb{R}^{2mn}$ is the dual variable. The two problems have the same optimal value, because Slater's constraint qualification is satisfied, cf. [3]. The SOCP in Eq. (B.4) can, in principle, be solved using standard interior-point algorithms, but the large dimensions typically render such an approach intractable.

3.1 The First-Order Method

Instead of using interior point algorithms, we adapt a first-order algorithm developed by Nesterov [17, 18] (similar to the approaches in [2] and [22]). Nesterov's algorithm is an efficient scheme for minimization of saddle point problems over bounded convex sets. The basic idea of this algorithm is to make a *smooth* $O(\epsilon)$-approximation with Lipschitz continuous derivatives to the non-differentiable TV function, and then subsequently minimize this approximation using an optimal first-order method for minimization of convex functions with Lipschitz continuous derivatives.

To adapt the TV denoising problem to Nesterov's method, we follow [3, §5.4] and rewrite Eq. (B.4) as a saddle point problem of the form

$$\min_{x \in Q_{\mathrm{p}}} \max_{u \in Q_{\mathrm{d}}} u^T D \, x,$$

where we have defined the primal and dual feasible sets

$$\begin{array}{rcl} Q_{\mathrm{p}} & = & \{x \,|\, \|x - b\|_2 \leq \delta\}, \\ Q_{\mathrm{d}} & = & \{u \,|\, \|u_{(ij)}\|_2 \leq 1, \; i = 1, \ldots, m, \; j = 1, \ldots, n\}. \end{array}$$

To each set Q_{p} and Q_{d} we associate a so-called *prox-function*, which we choose as, respectively,

$$f_{\mathrm{p}}(x) = \tfrac{1}{2}\|x - b\|_2^2 \qquad \text{and} \qquad f_{\mathrm{d}}(u) = \tfrac{1}{2}\|u\|_2^2.$$

These functions are bounded above as

$$\Delta_{\mathrm{p}} = \max_{x \in Q_{\mathrm{p}}} f_{\mathrm{p}}(x) = \tfrac{1}{2}\delta^2 \qquad \text{and} \qquad \Delta_{\mathrm{d}} = \max_{u \in Q_{\mathrm{d}}} f_{\mathrm{d}}(u) = \tfrac{1}{2}mn.$$

As a smooth approximation for $\mathcal{T}(x)$ we then use an additive modification of $\mathcal{T}(x)$ with the prox-function associated with Q_{d}:

$$\mathcal{T}_\mu(x) = \max_{u \in Q_{\mathrm{d}}} \{u^T D \, x - \mu \, f_{\mathrm{d}}(u)\}. \tag{B.6}$$

The approximation $\mathcal{T}_\mu(x)$ then bounds $\mathcal{T}(x)$ as $\mathcal{T}_\mu(x) \leq \mathcal{T}(x) \leq \mathcal{T}_\mu(x) + \mu\Delta_{\mathrm{d}}$, meaning that if we set $\mu = \epsilon/(2\Delta_{\mathrm{d}}) = \epsilon/(mn)$ then we have an $(\epsilon/2)$-approximation of $\mathcal{T}(x)$. Furthermore, following [18], it can be shown that $\mathcal{T}_\mu(x)$ has Lipschitz continuous derivatives with constant

$$\mathcal{L}_\mu = \mu^{-1}\|D\|_2^2 \leq 8/\mu,$$

and its gradient is given by

$$\nabla\mathcal{T}_\mu(x) = D^T u,$$

where u is the solution to (B.6) for a given x.

Given data b and a tolerance ϵ.

Set $x^{[0]} = b$ (a feasible starting point), $\mu = \frac{\epsilon}{2\Delta_{\mathrm{d}}}$, and $\mathcal{L}_\mu = \frac{\|D\|_2^2}{\mu}$.

For $k = 0, 1, 2, \ldots$

1) Evaluate $g^{[k]} = \nabla\mathcal{T}_\mu(x^{[k]})$.

2) Find $y^{[k]} = \arg\min_{x\in Q_{\mathrm{p}}} \Big\{ (x - x^{[k]})^T g^{[k]} + \frac{1}{2}\mathcal{L}_\mu \|x - x^{[k]}\|_2^2 \Big\}$.

3) Find $z^{[k]} = \arg\min_{x\in Q_{\mathrm{p}}} \Big\{ \mathcal{L}_\mu\, f_{\mathrm{p}}(x) + \sum_{i=0}^{k} \frac{i+1}{2}(x - x^{[i]})^T g^{[k]} \Big\}$.

4) Update $x^{[k+1]} = \frac{2}{k+3}\, z^{[k]} + \frac{k+1}{k+3}\, y^{[k]}$.

Fig. B.1: Nesterov's first-order method for discrete TV denoising. We stop the iterations when the duality gap is less than ϵ.

Nesterov's optimal first-order method for minimizing the convex function $\mathcal{T}_\mu(x)$ with Lipschitz continuous derivatives is listed in Fig. B.1. We terminate the algorithm when the duality gap satisfies

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \|D_{(ij)}\, x\|_2 + \delta\|D^T u\|_2 - u^T D\, b < \epsilon.$$

When the iterations are stopped by this criterion, leading to the solution x^ϵ, then we are ensured that the found solution is close to the exact solution $x^\star$ in the sense that $\mathcal{T}(x^\epsilon) - \mathcal{T}(x^\star) < \epsilon$. We remark that with our formulation of the problem it is difficult to relate the parameter ϵ to the error $\|x^\epsilon - x^\star\|_2$ a priori (while this is possible in the dual formulation in [24] where the primal variable is a function of the dual variable).

By specifying the threshold ϵ for the duality gap, we can determine the parameter $\mu = \epsilon/(mn)$ used in the TV denoising algorithm to evaluate $\mathcal{T}_\mu(x)$ (B.6). Nesterov showed in [18] that at most

$$\mathcal{N} = \frac{4\|D\|_2}{\epsilon}\sqrt{\Delta_{\mathrm{p}}\,\Delta_{\mathrm{d}}} \tag{B.7}$$

iterations are required to reach an ϵ-optimal solution. For the discrete TV denoising algorithm we obtain the bound

$$\mathcal{N}_{\mathrm{denoise}} = \frac{2\|D\|_2}{\epsilon}\delta\sqrt{mn} \leq \frac{4\sqrt{2mn}}{\epsilon}\,\delta. \tag{B.8}$$

We return to the choice of ϵ in Section 7.

3.2 Efficient Implementation

The key to an efficient implementation of our algorithm is to evaluate $g^{[k]}$ in step 1) and solve the two subproblems 2) and 3) efficiently. This is ensured by our choice of prox-functions f_{p} and f_{d}. By a simple change of variables it turns out that all three quantities can be written as the solution to a simple quadratically constrained problem of the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\,\theta^T\theta - \theta^T c \\ \text{subject to} & \|\theta\|_2 \leq \eta, \end{array}$$

whose solution is simply given by $\theta = c\,/\max\{1, \|c\|_2/\eta\}$. In step 1) we must evaluate $g^{[k]} = \nabla\mathcal{T}_\mu(x^{[k]})$ and it is easy to show that the gradient is given by $\nabla\mathcal{T}_\mu(x^{[k]}) = D^T u^{[k]}$, where $u^{[k]}$ is given by

$$u^{[k]} = \arg\max_{u\in Q_{\mathrm{d}}} u^T D\,x^{[k]} - \frac{\mu}{2}\|u\|_2^2.$$

The mn sub-vectors $u_{(ij)}^{[k]}$ of $u^{[k]}$ are thus given by

$$u_{(ij)}^{[k]} = D_{(ij)}x^{[k]}\,/\max\{\mu, \|D_{(ij)}x^{[k]}\|_2\}.$$

In step 2) it follows from a simple variable transformation that

$$y^{[k]} = \left(\mathcal{L}_\mu\left(x^{[k]} - b\right) - g^{[k]}\right)/\,\max\Big\{\mathcal{L}_\mu,\, \big\|\mathcal{L}_\mu\left(x^{[k]} - b\right) - g^{[k]}\big\|_2/\,\delta\Big\} + b,$$

and in step 3) we similarly obtain

$$z^{[k]} = -w^{[k]}/\,\max\Big\{\mathcal{L}_\mu,\, \big\|w^{[k]}\big\|_2\,/\,\delta\Big\} + b,$$

where we have introduced $w^{[k]} = \sum_{i=0}^{k} \frac{1}{2}(i+1)\, g^{[i]}$.

The computations in each of the steps 1) to 4) are done efficiently in $O(mn)$ operations. If needed, the algorithm is also very easy to parallelize; the subproblem 1) can be divided in several separate problems, and steps 2) and 3) can be executed in parallel. The memory requirements are also very modest, requiring only memory for storing the five mn-vectors $g^{[k]}$, $w^{[k]}$, $x^{[k]}$, $y^{[k]}$, $z^{[k]}$, plus a temporary mn-vector – which is equivalent to the storage for 6 images in total. By exploiting the structure of D, it is not necessary to store the vector $u^{[k]}$ but only $u_{(ij)}^{[k]}$.

4 Inpainting

In this section we extend the total-variation denoising algorithm to include *inpainting*, i.e., the process of filling in missing or damaged parts of a (possibly noisy) image, cf. [5]. The basic idea is still to compute a reconstruction that is "smooth" in the TV sense, and identical to the data in all the non-corrupted pixels (or close to these data if they are noisy).

Specifically, let $\mathcal{I}$ be the index set for x corresponding to the corrupted pixels in X. The complementary index set $\mathcal{I}_c$ is the set of non-corrupted pixels. The basic TV inpainting problem can then be formulated as

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m}\sum_{j=1}^{n} \|D_{(ij)}\, x\|_2 \\ \text{subject to} & \|\,(x-b)_{\mathcal{I}_c}\|_2 \le \delta, \end{array}$$

with the dual problem

$$\begin{array}{ll} \text{maximize} & -\delta\|(D^T u)_{\mathcal{I}_c}\|_2 + b_{\mathcal{I}_c}^T (D^T u)_{\mathcal{I}_c} \\ \text{subject to} & \|u_{(ij)}\|_2 \le 1, \quad i = 1,\ldots,m,\ j = 1,\ldots,n \\ & (D^T u)_{\mathcal{I}} = 0. \end{array}$$

In this primal-dual formulation, the dual feasible set is not simple because of the equality constraint $(D^T u)_{\mathcal{I}} = 0$ and hence the subproblem in step 1) of Fig. B.1 will be complicated. Instead we bound the primal feasible set by adding an artificial norm-constraint on the pixels in the inpainting region, leading to the revised formulation

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m}\sum_{j=1}^{n} \|D_{(ij)}\, x\|_2 \\ \text{subject to} & \|(x-b)_{\mathcal{I}_c}\|_2 \le \delta \\ & \|(x-d)_{\mathcal{I}}\|_2 \le \gamma, \end{array} \tag{B.9}$$

for some suitable vector d and parameter $\gamma > 0$. The dual problem corresponding

to (B.9) is then

$$\begin{array}{ll} \text{maximize} & -\delta\,\|(D^Tu)_{\mathcal{I}_c}\|_2 + b_{\mathcal{I}_c}^T(D^Tu)_{\mathcal{I}_c} - \gamma\,\|(D^Tu)_{\mathcal{I}}\|_2 + d_{\mathcal{I}}^T(D^Tu)_{\mathcal{I}} \\ \text{subject to} & \|u_{(ij)}\|_2 \le 1, \quad i = 1,\ldots,m,\ j = 1,\ldots,n\,, \end{array} \tag{B.10}$$

and now we have simple constraints (similar to the denoising problem).

It is important that d and γ in (B.9) are chosen such that $\|(x-d)_{\mathcal{I}}\|_2 < \gamma$ holds for the solution of the original problem. The pixel intensity in the inpainted region is always bounded by the intensity of the non-corrupted pixels, i.e., the vector of inpainted pixels satisfies

$$x_{\mathcal{I}}^\star \in P = \left\{ z \mid \min_{i\in\mathcal{I}_c} b_i \le z_j \le \max_{i\in\mathcal{I}_c} b_i, \quad \forall j \in \mathcal{I} \right\}.$$

If we then set the elements of the vector d to

$$d_j = \tfrac{1}{2}\left(\max_{i\in\mathcal{I}_c} b_i + \min_{i\in\mathcal{I}_c} b_i \right) \quad \forall\, j \in \mathcal{I},$$

i.e., d is the midpoint in the set P, then we have

$$\|(x^\star - d)_{\mathcal{I}}\|_2 \le \max_{x_{\mathcal{I}}\in P} \|x_{\mathcal{I}} - d_{\mathcal{I}}\|_2 = \frac{1}{2}\left(\max_{i\in\mathcal{I}_c} b_i - \min_{i\in\mathcal{I}_c} b_i \right)\sqrt{|I|} := \gamma,$$

which we then select as our γ. These settings guarantee that we have an artificial norm-constraint that is inactive at the solution. The primal set is now $Q_{\mathrm{p}}' = \{x \mid \|(x-b)_{\mathcal{I}_c}\|_2 \le \delta,\ \|(x-d)_{\mathcal{I}}\|_2 \le \gamma\}$, and as the prox-function for this set we use

$$f_{\mathrm{p}}'(x) = \tfrac{1}{2}\|(x-b)_{\mathcal{I}_c}\|_2^2 + \tfrac{1}{2}\|(x-d)_{\mathcal{I}}\|_2^2 \tag{B.11}$$

with upper bound $\Delta_{\mathrm{p}}' = \frac{1}{2}(\gamma^2 + \delta^2)$. As prox-function for Q_{d} (which is unchanged) we again use $f_{\mathrm{d}}(u) = \frac{1}{2}\|u\|_2^2$ and μ is chosen similarly as in §3.

Regarding the implementation issues, only step 2) and step 3) in the algorithm from Fig. B.1 change in the TV inpainting algorithm. Note that the two cone constraints in (B.9) are non-overlapping and that the norms in the prox-function (B.11) are partitioned in the same way as the constraints. Hence, the two index sets of $y^{[k]}$ in step 2) can be computed separately, and they are given by

$$\begin{aligned} y_{\mathcal{I}_c}^{[k]} &= \big(\mathcal{L}_\mu(x^{[k]} - b) - g^{[k]}\big)_{\mathcal{I}_c} / \max\big\{\mathcal{L}_\mu,\ \|\big(\mathcal{L}_\mu(x^{[k]} - b) - g^{[k]}\big)_{\mathcal{I}_c}\|_2 / \delta\big\} + b_{\mathcal{I}_c} \\ y_{\mathcal{I}}^{[k]} &= \big(\mathcal{L}_\mu(x^{[k]} - d) - g^{[k]}\big)_{\mathcal{I}} / \max\big\{\mathcal{L}_\mu,\ \|\big(\mathcal{L}_\mu(x^{[k]} - d) - g^{[k]}\big)_{\mathcal{I}}\|_2 / \gamma\big\} + d_{\mathcal{I}}\,. \end{aligned}$$

Similarly in step 3) we have

$$\begin{aligned} z_{\mathcal{I}_c}^{[k]} &= -w_{\mathcal{I}_c}^{[k]} / \max\big\{\mathcal{L}_\mu,\ \|w_{\mathcal{I}_c}^{[k]}\|_2/\delta\big\} + b_{\mathcal{I}_c}\,, \\ z_{\mathcal{I}}^{[k]} &= -w_{\mathcal{I}}^{[k]} / \max\big\{\mathcal{L}_\mu,\ \|w_{\mathcal{I}}^{[k]}\|_2/\gamma\big\} + d_{\mathcal{I}}\,. \end{aligned}$$

The upper bound for the number of iterations in the discrete TV inpainting algorithm becomes

$$\mathcal{N}_{\text{inpaint}} = 2\|D\|_2\sqrt{(\gamma^2+\delta^2)mn}\cdot\frac{1}{\epsilon} \leq \frac{4\sqrt{2mn}}{\epsilon}\sqrt{\gamma^2+\delta^2}. \tag{B.12}$$

Note that γ enters the bound in the same way as δ. However, while δ is typically small – of the same size as the errors in the data – the parameter γ is of the same size as the norm of the inpainted pixels $x_{\mathcal{I}}$. This illustrates the difficulty of the inpainting problem, in terms of computational complexity – compared to the denoising problem – when using Nesterov's method with our choices of prox-functions.

Similarly to §3, the complexity of each of the subproblem is $O(mn)$ with the same memory requirement.

5 Deblurring for Reflexive Boundary Conditions

In addition to denoising and inpainting, it is natural to consider TV *deblurring* of images, where the blurring is modelled by a linear operator, i.e., the blurred image is given by

$$b = K\,x^{\text{exact}} + \text{noise},$$

in which $K \in \mathbb{R}^{mn\times mn}$ is a known matrix that represents the linear blurring in the image B [15]. TV deblurring then amounts to computing a reconstruction which is, once again, "smooth" in the TV sense and fits the noisy data b within a tolerance δ that acts as the regularization parameter. Hence the discrete TV deblurring problem can be formulated as

$$\begin{aligned} &\text{minimize} && \textstyle\sum_{i=1}^{m}\sum_{j=1}^{n}\|D_{(ij)}\,x\|_2 \\ &\text{subject to} && \|Kx-b\|_2 \leq \delta. \end{aligned}$$

Here we only consider spatially invariant blurring with a doubly symmetric point spread function and reflexive boundary conditions, for which the matrix K can be diagonalized by a two-dimensional discrete cosine transform (DCT) [15]. The algorithm is easily extended to other matrices K that can be diagonalized efficiently by an orthogonal or unitary similarity transform (e.g., the discrete Fourier transform for general point spread functions and periodic boundary conditions), or by singular value decomposition of smaller matrices, such as is the case for separable blur where K is a Kronecker product.

We thus assume that K can be diagonalized by an orthogonal similarity transform,

$$CKC^T = \Lambda = \text{diag}(\lambda_i), \tag{B.13}$$

where the matrix C represents the two-dimensional DCT, and Λ is a real diagonal matrix with the eigenvalues of K. Then by a change of variables $\bar{x} = Cx$ and $\bar{b} = Cb$ we obtain the equivalent TV deblurring problem in the DCT basis

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m}\sum_{j=1}^{n} \|D_{(ij)}C^T\,\bar{x}\|_2 \\ \text{subject to} & \|\Lambda\,\bar{x} - \bar{b}\|_2 \le \delta. \end{array}$$

We note that multiplications with C and C^T are implemented very efficiently by means of the DCT algorithm with complexity $mn\log(\max\{m,n\})$. In our software we use the C package FTTW [10], [12], and it is needed only for TV deblurring. FFTW is known as the fastest free software implementation of the Fast Fourier Transform algorithm. It can compute transforms of real- and complex-valued arrays (including the DCT) of arbitrary size and dimension, and it does this by supporting a variety of algorithms and choosing the one it estimates or measures to be preferable in the particular circumstance.

5.1 Rank Reduction

Often Λ is singular – either exactly or within the accuracy of the finite-precision computations – in which case the feasible set $\{x \mid \|\Lambda\,\bar{x} - \bar{b}\|_2 \le \delta\}$ is unbounded, and as such the problem cannot be solved using Nesterov's method. Moreover, when the condition number $\mathrm{cond}(\Lambda) = \max_i|\lambda_i| / \min_i|\lambda_i|$ is large (or infinite), we experience numerical difficulties and slow convergence of the algorithm.

To overcome these difficulties we apply the well-known approach of *rank reduction* and divide the eigenvalues into two partitions: One set with sufficiently large values indexed by $\mathcal{I} = \{\, i \,|\, |\lambda_i| > \rho\|K\|_2\}$, and the complementary set indexed by $\mathcal{I}_c$. Here, $\|K\|_2 = \max_j|\lambda_j|$, and ρ is a parameter satisfying $0 < \rho < 1$. We also define the diagonal matrix Λ^ρ whose diagonal elements are given by

$$(\Lambda^\rho)_{ii} = \begin{cases} \lambda_i & \text{if } i \in \mathcal{I} \\ 0 & \text{else,} \end{cases}$$

and we note that Λ^ρ is the closest rank-$|\mathcal{I}|$ approximation to Λ. The default value of ρ in our software is $\rho = 10^{-3}$

We then solve a slightly modified deblurring problem obtained by replacing the matrix K with the implicitly defined rank-deficient approximation

$$K^\rho = C^T\Lambda^\rho\,C\,.$$

The corresponding rank-reduced TV deblurring problem is thus

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{m}\sum_{j=1}^{n} \|D_{(ij)}C^T\,\bar{x}\|_2 \\ \text{subject to} & \|(\Lambda\,\bar{x} - \bar{b})_{\mathcal{I}}\|_2 \le \delta \\ & \|\bar{x}_{\mathcal{I}_c}\|_2 \le \gamma, \end{array} \qquad \text{(B.14)}$$

where $\bar{x}_{\mathcal{I}_c}$ should be considered as unconstrained variables. The parameter γ must therefore be chosen sufficiently large such that the constraint $\|\bar{x}_{\mathcal{I}_c}\|_2 \leq \gamma$ is inactive at the solution. The extra constraint is added for the same reason as in the inpainting problem, namely, to keep the dual feasible set simple.

In addition to improving the numerical stability and reducing the number of iterations, rank-reduced deblurring can also be seen as another way of imposing regularization on the ill-posed problem by reducing the condition number for the problem from $\text{cond}(\Lambda)$ to $\text{cond}(\Lambda^\rho) \leq 1/\rho$.

Choosing γ to guarantee that the γ-bound is inactive is difficult without making γ too large and thereby increasing the number of iterations. We assume without loss of generality that we can scale K such that $\|x^{\text{exact}}\|_2 \approx \|b\|_2$. This means that a solution which is properly regularized will also have $\|\bar{x}\|_2 = \|x\|_2 \approx \|\bar{b}\|_2 \approx \|b\|_2$. Our software therefore scales K and selects

$$\gamma = \sqrt{mn}\|b\|_\infty,$$

which guarantees that γ is sufficiently large. If the artificial γ-bound in (B.14) is active at the solution, then this is a sign that the problem might not be sufficiently regularized due to a too large value of δ.

We remark that the first inequality constraint in problem (B.14) is infeasible unless $\|(\Lambda\,\bar{x})_{\mathcal{I}} - \bar{b}_{\mathcal{I}}\|_2^2 + \|\bar{b}_{\mathcal{I}_c}\|_2^2 \leq \delta^2$, i.e., δ must always be large enough to ensure that $\|\bar{b}_{\mathcal{I}_c}\|_2 \leq \delta$, which is checked by our software. This is no practical difficulty, because δ must always be chosen to reflect the noise in the data. The requirement $\|\bar{b}_{\mathcal{I}_c}\|_2 \leq \delta$ simply states that δ must be larger than the norm of the component of b in the null space of K^ρ, and according to the model (B.13) this component is dominated by the noise.

With the notation $\Lambda_{\mathcal{I}} = \text{diag}(\lambda_i)_{i\in\mathcal{I}}$, the dual problem of (B.14) is

$$\begin{array}{ll} \text{maximize} & -\delta\,\|\Lambda_{\mathcal{I}}^{-1}(CD^Tu)_{\mathcal{I}}\|_2 - \gamma\|(CD^Tu)_{\mathcal{I}_c}\|_2 + \bar{b}_{\mathcal{I}}^T\Lambda_{\mathcal{I}}^{-1}(CD^Tu)_{\mathcal{I}} \\ \text{subject to} & \|u_{(ij)}\|_2 \leq 1, \quad i = 1,\ldots,m,\ j = 1,\ldots,n. \end{array} \tag{B.15}$$

As the prox-function for the primal set $Q_{\text{p}}'' = \{\bar{x} \mid \|(\Lambda\,\bar{x} - \bar{b})_{\mathcal{I}}\|_2 \leq \delta,\ \|\bar{x}_{\mathcal{I}_c}\|_2 \leq \gamma\}$ we use

$$f_{\text{p}}''(\bar{x}) = \tfrac{1}{2}\|\bar{x}\|_2^2.$$

The corresponding upper bound $\Delta_{\text{p}}'' = \max_{\bar{x}\in Q_{\text{p}}''} f_{\text{p}}''(\bar{x})$ can be evaluated numerically as the solution to a trust-region subproblem discussed below. We can bound it as

$$\Delta_{\text{p}}'' \leq \tfrac{1}{2}\left(\|\Lambda_{\mathcal{I}}^{-1}\bar{b}_{\mathcal{I}}\|_2^2 + \gamma^2\right) \leq \tfrac{1}{2}\left(\frac{\|b\|_2^2}{\rho^2\|K\|_2^2} + \gamma^2\right).$$

The upper bound for the number of iterations is

$$\mathcal{N}_{\text{deblur}} = \sqrt{8}\|D\|_2\sqrt{\Delta_{\text{p}}''\,mn}\cdot\frac{1}{\epsilon} \leq 4\sqrt{2mn}\left(\frac{\|b\|_2^2}{\rho\|K\|_2^2} + mn\|b\|_\infty^2\right)\cdot\frac{1}{\epsilon}. \tag{B.16}$$

5.2 Implementation

Compared to the TV denoising algorithm from §3 there are a few changes in the implementation. In step 1) the computation of $u_{(ij)}^{[k]}$ now takes the form

$$u_{(ij)}^{[k]} = D_{(ij)}C^T\bar{x}^{[k]} / \max\{\mu, \|D_{(ij)}C^T\bar{x}^{[k]}\|_2\},$$

which is computed in $mn\log(\max\{m,n\})$ complexity. For the computations in steps 2) and 3), first note that the two cones in Q_{p}'' are non-overlapping because $\mathcal{I}\cap\mathcal{I}_{\mathrm{c}} = \emptyset$, and the subproblems can therefore be treated as two separated cases as we had for the inpainting algorithm in §4. The minimizers $y_{\mathcal{I}}^{[k]}$ and $z_{\mathcal{I}}^{[k]}$ can be found (via simple changes of variables) as the solution to the well-studied *trust-region subproblem* [8], i.e., as the solution to a problem of the form

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\,\theta^T\theta - c^T\theta \\ &\text{subject to} && \|L\,\theta - y\|_2 \le \eta \end{aligned} \tag{B.17}$$

where $L = \mathrm{diag}(\ell_i)$ is a diagonal matrix. We first check whether c satisfies the constraint, i.e., if $\|L\,c - y\|_2 \le \eta$ then $\theta = c$. Otherwise, we find the global minimum of the problem, using Newton's method to compute the unique root $\lambda > -\min_i\{\ell_i\}$ of the so-called *secular equation* [8, §7.3.3]

$$q^T(L^{-2} + \lambda\, I)^{-2}q = \sum_{i=1}^{mn}\frac{q_i^2}{(\ell_i^{-2}+\lambda)^2} = \eta,$$

where I is the identity matrix and

$$q = L^{-1}c - L^{-2}y\,.$$

Once the root λ has been found, the solution to (B.17) is given by

$$\theta = L^{-1}\left(b + \left(L^{-2}+\lambda\right)^{-1}q\right).$$

As the starting value for λ in Newton's method, we can use the solution from the previous (outer) iteration in Nesterov's method. Our experience is that this limits the number of Newton iterations in the trust-region method to just a few iterations each with complexity $O(mn)$, i.e., in practice the cost of computing the solution to steps 2) and 3) is still $O(mn)$.

The minimizers $y_{\mathcal{I}_{\mathrm{c}}}^{[k]}$ and $z_{\mathcal{I}_{\mathrm{c}}}^{[k]}$ are both computed as the solution to the quadratic constrained problems. For step 2) we obtain

$$\begin{aligned} y_{\mathcal{I}}^{[k]} &= \theta \text{ in (B.17) with } c = x_{\mathcal{I}}^{[k]} - g_{\mathcal{I}}^{[k]}\mathcal{L}_\mu^{-1},\ L = \Lambda_{\mathcal{I}}, \text{ and } \eta = \delta, \\ y_{\mathcal{I}_{\mathrm{c}}}^{[k]} &= \left(\mathcal{L}_\mu\, x_{\mathcal{I}_{\mathrm{c}}}^{[k]} - g_{\mathcal{I}_{\mathrm{c}}}^{[k]}\right) / \max\left\{\mathcal{L}_\mu,\ \|(\mathcal{L}_\mu\, x_{\mathcal{I}_{\mathrm{c}}}^{[k]} - g_{\mathcal{I}_{\mathrm{c}}}^{[k]})\|_2 \,/\, \gamma\right\}, \end{aligned}$$

and in step 3) we similarly have

$$\begin{aligned} z_{\mathcal{I}}^{[k]} &= \theta \text{ in (B.17) with } c = -w_{\mathcal{I}}^{[k]}\mathcal{L}_{\mu}^{-1},\ L = \Lambda_{\mathcal{I}}, \text{ and } \eta = \delta, \\ z_{\mathcal{I}_c}^{[k]} &= -w_{\mathcal{I}_c}^{[k]} / \max\{\mathcal{L}_{\mu},\ \|w_{\mathcal{I}_c}^{[k]}\|_2 / \gamma\}\,. \end{aligned}$$

The bound Δ_{p}'' on the primal set can be obtained a priori as

$$\Delta_{\mathrm{p}}'' = \tfrac{1}{2}\left(\|\theta\|_2^2 + \gamma^2\right),$$

where θ here is the solution to the problem

$$\begin{aligned} \text{minimize} \quad & -\tfrac{1}{2}\,\theta^T \Lambda_{\mathcal{I}}\Lambda_{\mathcal{I}}\theta + b_{\mathcal{I}}^T\Lambda_{\mathcal{I}}\theta \\ \text{subject to} \quad & \|\theta\|_2 \le \eta \end{aligned}$$

which can be solved using the same method as the previous trust region problem.

The complexity of step 1) in the TV deblurring algorithm increases, compared to the previous two algorithms, since we need to compute a two-dimensional DCT of the current iterate $x^{[k]}$ as well as an inverse two-dimensional DCT of $g^{[k]}$, i.e., the complexity per iteration of the algorithm is thus dominated by these $mn\log(\max\{m,n\})$ computations. The memory requirements of the algorithm is increased by the vectors holding q, q element-wise squared, and the diagonal elements of L^{-2} to avoid re-computation, plus and an extra temporary vector, leading to a total memory requirement of about $10mn$.

6 Numerical Examples

In this section we give numerical examples that illustrate the three TV algorithms from the previous sections. All the algorithms are implemented in the C programming language, and the examples are run on a 2 GHz Intel Core 2 Duo computer with 2 GB of memory running the Linux operating system and using a single processor. We provide the three m-files `TVdenoise`, `TVinpaint`, and `TVdeblur` such that the C functions can be used from Matlab, and we also provide corresponding demo Matlab scripts that generate the examples in this section.

In the first example we consider the **TV denoising algorithm** from §3. The top images in Fig. B.2 show the pure 512×512 image and the same image corrupted by additive white Gaussian noise with standard deviation $\sigma = 25$, leading to a signal-to-noise ratio $20\log_{10}(\|X\|_{\mathrm{F}}/\|X - B\|_{\mathrm{F}}) = 15$ dB. For our TV reconstructions, we choose the parameter δ such that it reflects the noise level in the image [13],

$$\delta = \tau\sqrt{mn}\,\sigma\ , \tag{B.18}$$

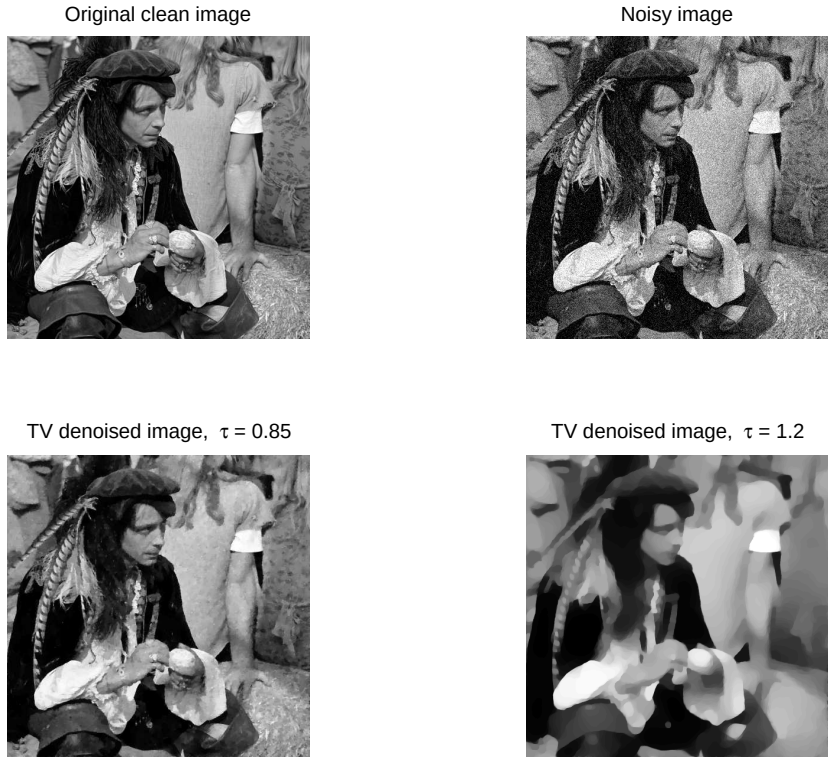


Fig. B.2: Example of TV denoising. Top: clean and noisy images of size 512×512. Bottom: TV reconstructions for two different choices of the parameter τ in Eq. (B.18).

where σ is the standard deviation of the noise, and τ is factor close to one. The two bottom images in Fig. B.2 show TV reconstructions for $\tau = 0.85$ and 1.2; the first choice leads to a good reconstruction, while the second choice is clearly too large, leading to a reconstruction that is too smooth in the TV sense (i.e., large domains with the same intensity, separated by sharp contours).

In the second example we illustrate the **TV inpainting algorithm** from §4, using the same clean image as above. Figure B.3 shows the damaged image and the TV reconstruction. The white pixels in the corrupted image show the missing pixels, and we also added noise with standard deviation $\sigma = 15$ to the intact pixels. There is a total of $|\mathcal{I}| = 27,452$ damaged pixels, corresponding to about 10% of the total amount of pixels. In the reconstruction we used

$$\delta = \tau \sqrt{|\mathcal{I}_c|}\sigma, \tag{B.19}$$

which is a slight modification of (B.18) to reflect the presence of corrupted pixels.

Fig. B.3: Example of TV inpainting: damaged and noisy 512×512 image (same clean image as in Fig. B.2), and the TV reconstruction.

In the example we used $\tau = 0.85$.

The third example illustrates the **TV deblurring algorithm** from §5, again using the same clean image. Figure B.4 shows the blurred and noise image and three TV reconstructions. We use Gaussian blur with standard deviation 3.0, leading to a coefficient matrix K with a numerically infinite condition number, and the standard deviation of the Gaussian noise is $\sigma = 3$. The regularization parameter δ is chosen by the same equation (B.18) as in denoising.

For $\tau = 0.2$, Fig. B.4 shows that we obtain an under-regularized solution dominated by inverted noise. The choice $\tau = 0.45$ gives a sufficiently piecewise-smooth image with satisfactory reconstruction of details, while $\tau = 1.0$ leads to an over-regularized image with too few details.

The computations associated with the blurring use the algorithm given in [15], and from the same source we use the Matlab functions `dcts2`, `idcts2`, and `dctshift` for the associated computations with the DCT.

7 Performance Studies

The choice of ϵ obviously influences the computing time, and we choose to design our software such that the number of iterations remains unchanged when the image size is scaled – i.e., we want the bounds $\mathcal{N}_{\text{denoise}}$ (B.8), $\mathcal{N}_{\text{inpaint}}$ (B.12), and $\mathcal{N}_{\text{deblur}}$ (B.16) to be independent of the problem size mn. In order to achieve this, instead of setting an absolute ϵ in the stopping criterion we use a *relative accuracy* ϵ_{rel} (with default value $\epsilon_{\text{rel}} = 10^{-3}$ for denoising and inpainting and

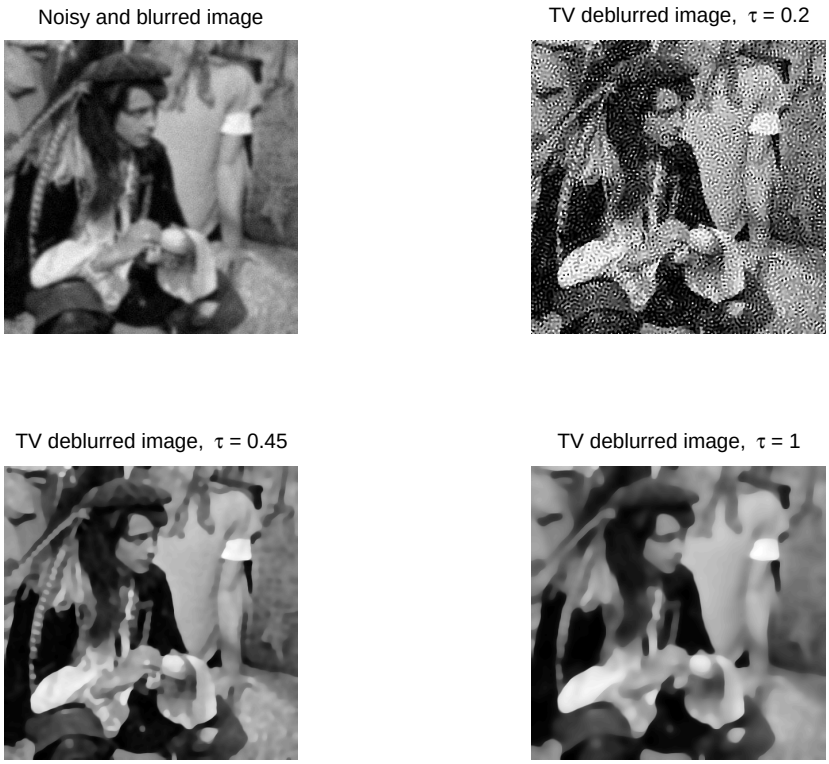


Fig. B.4: Example of TV deblurring: blurred and noisy 512×512 image (same clean image as in Fig. B.2), and TV reconstructions with three different values of τ.

$\epsilon_{\text{rel}} = 10^{-2}$ for deblurring), and then we set

$$\epsilon = \begin{cases} \|b\|_\infty \, mn \, \epsilon_{\text{rel}}, & \text{for denoising and deblurring} \\ \|b_{\mathcal{I}_c}\|_\infty \, mn \, \epsilon_{\text{rel}}, & \text{for inpainting.} \end{cases} \tag{B.20}$$

This choice, together with (B.18) and (B.19), leads to the bounds

$$\begin{aligned} \mathcal{N}_{\text{denoise}} &\leq \frac{4\sqrt{2}}{\epsilon_{\text{rel}}} \frac{\tau\,\sigma}{\|b\|_\infty} \\ \mathcal{N}_{\text{inpaint}} &\leq \frac{4\sqrt{2}}{\epsilon_{\text{rel}}} \sqrt{\left(\frac{\tau\,\sigma}{\|b_{\mathcal{I}_c}\|_\infty}\right)^2 \frac{|\mathcal{I}_c|}{mn} + \left(\frac{\max_{i\in\mathcal{I}_c} b_i - \min_{i\in\mathcal{I}_c} b_i}{2\,\|b_{\mathcal{I}_c}\|_\infty}\right)^2 \frac{|\mathcal{I}|}{mn}} \\ \mathcal{N}_{\text{deblur}} &\leq \frac{4\sqrt{2}}{\epsilon_{\text{rel}}} \sqrt{1 + \left(\frac{1}{\rho \max_i |\lambda_i|}\right)^2} \approx \frac{4\sqrt{2}}{\epsilon_{\text{rel}}} \frac{1}{\rho \max_i |\lambda_i|}. \end{aligned}$$

For denoising, the bound is proportional to the relative noise level, as desired. For inpainting, the situation is more complex, but if the noise dominates then we have the same bound as in denoising, and otherwise the bound is proportional to the square root of the fraction of missing pixels. For deblurring, the bound is dominated by the term involving the smallest eigenvalue $\rho \max_i |\lambda_i|$ in the rank-deficient approximation.

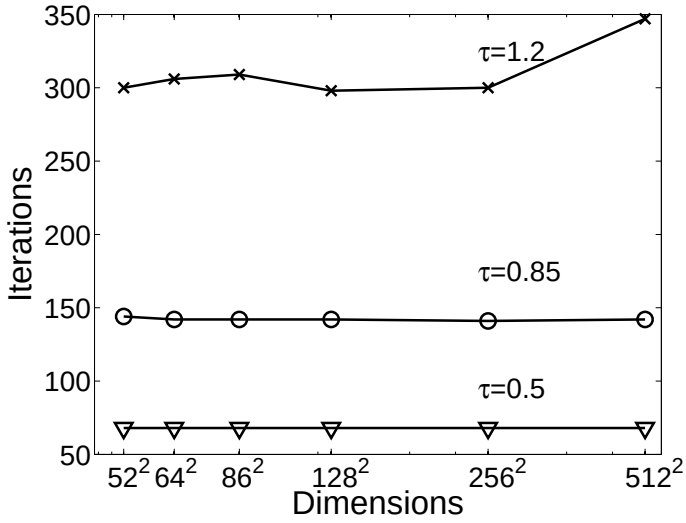


Fig. B.5: The number of iterations in `TVdenoise` needed to compute an ϵ-accurate solution to the TV denoising problem, for varying image dimensions and three values of the parameter τ. The standard deviation of the image noise is $\sigma = 25$, and as stopping criterion we used the default value $\epsilon_{\text{rel}} = 10^{-3}$. For the tree values of τ, the bounds for $\mathcal{N}_{\text{denoise}}$ are 278, 472, and 666, respectively.

To demonstrate the computational performance of our **TV denoising algorithm**, we created several smaller problems by extracting sub-images of the original clean image, and in each instance we added Gaussian white noise with standard deviation $\sigma = 25$ (similar to the previous section). We then solved these TV denoising problems using the default parameter $\epsilon_{\text{rel}} = 10^{-3}$ and for three different values of τ, and the actual number of iterations needed to solve the problem to ϵ-accuracy are shown in Fig. B.5. We see that with the choice of ϵ in (B.20), the actual number of iterations is indeed almost independent of the problem size (except for unrealistic large τ). We also see that the actual number of iterations is approximately proportional to τ, and the bounds for $\mathcal{N}_{\text{denoise}}$ are somewhat pessimistic overestimates.

While the number of iterations is almost independent of the problem size, the computing time increases with the problem size because each iteration has

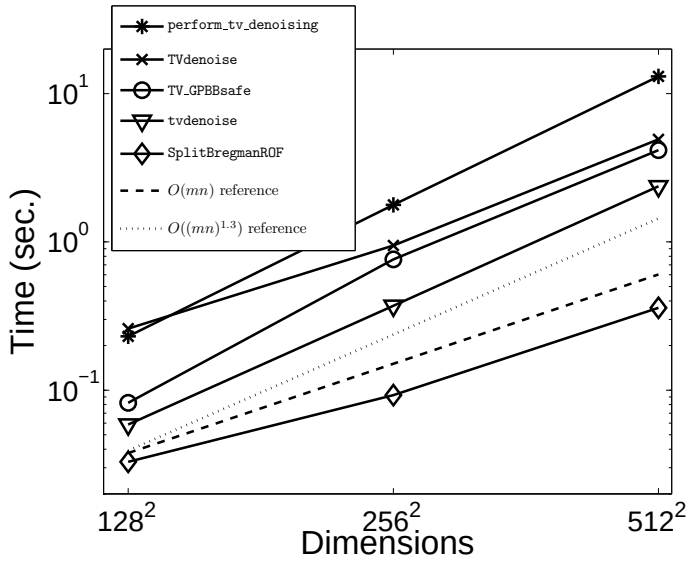


Fig. B.6: The computing times (in seconds) for our TV denoising algorithm `TVdenoise` as a function of problem size mn, for the case $\sigma = 25$, $\tau = 0.85$, and $\epsilon_{\text{rel}} = 10^{-3}$. The dashed reference line without markers confirms that the computing time for `TVdenoise` is approximately linear in the problem size. We also show the computing times `tvdenoise`, `perform_tv_denoising`, `TV_GPBBsafe` (from `TVGP`) and `SplitBregmanROF` listed in Table B.1. The dotted reference line without markers shows that the computing time for first two of the mentioned algorithms is approximately $O((mn)^{1.3})$, whereas `SplitBregmanROF` scales approximately linear.

$O(mn)$ complexity. Figure B.6 shows the computing time for our TV denoising algorithm `TVdenoise`, and the dashed reference line confirms that the computing time is approximately linear in the problem size mn.

We compared our code with the codes `tvdenoise`, `perform_tv_denoising`, `TV_GPBBsafe` (from `TVGP`) and `SplitBregmanROF` from Table B.1 (`TV_GPBBsafe` was chosen because it is the fastest method from `TVGP` for which convergence is guaranteed). These codes solve the Lagrange formulation of the TV denoising by minimizing problems on the form

$$\mathcal{T}(x) + \frac{1}{2\lambda}\|x - b\|_2^2 \,. \tag{B.21}$$

There is equivalence between the regularized and the constrained TV denoising formulations. If we set

$$\delta = \lambda \, \|D^T u^\star\|_2, \tag{B.22}$$

where $u^\star$ is the solution to the dual problem (B.5), then the two problems (B.4) and (B.21) are equivalent [13].

First we solved (B.5) to high accuracy with $\epsilon_{\text{rel}} = 10^{-6}$ for 100 different noise realizations, and then used (B.22) to obtain the corresponding Lagrange multiplier λ. We then picked the highest number of iterations for `tvdenoise`, `perform_tv_denoising`, and `TV_GPBBsafe` such that these codes returned a solution $x_{\mathcal{R}}$ slightly less accurate than the solution x from our code, i.e.,

$$\mathcal{R}_{\text{denoise}}(x) \leq \mathcal{R}_{\text{denoise}}(x_{\mathcal{R}})$$

where

$$\mathcal{R}_{\text{denoise}}(x) = \sum_{i=2}^{m-1} \sum_{j=2}^{n-1} \|D_{(ij)}\, x\|_2 + \frac{1}{2\lambda} \|(x-b)_{\mathcal{J}}\|_2^2 \,, \tag{B.23}$$

where $\mathcal{J}$ is the index set of all inner pixels. The image boundaries are removed in (B.23) to reduce the effect of the boundary conditions imposed by the different algorithms.

The average computing times are shown in Fig. B.6, and we see that the codes `tvdenoise`, `perform_tv_denoising`, and `TV_GPBBsafe` (for larger images) have a complexity of about $O((mn)^{1.3})$ as confirmed by the dotted reference line. For large images `perform_tv_denoising` is the slowest of these codes, while `tvdenoise` and `TV_GPBBsafe` are faster. The code `SplitBregmanROF` is the fastest and it scales with a complexity of about $O(mn)$. For the image dimensions shown, our code is faster than `perform_tv_denoising` but slower than `tvdenoise`, `TV_GPBBsafe`, and `SplitBregmanROF`. However, due to the lower complexity our algorithm scales as good as `SplitBregmanROF`.

For the **TV inpainting algorithm** the computing times depend on image dimensions and noise level as well as on the number and distribution of the missing pixels. We illustrate this with an example with noise level $\sigma = 15$ (similar to Fig. B.3, and with the parameters $\tau = 0.85$ and $\epsilon_{\text{rel}} = 10^{-3}$). The problem shown in Fig. B.3 (with the text mask) is solved in 28.1 seconds. However, if we generate a mask with same number of missing pixels located in a circle (of radius 93 pixels) in the middle of the image, then the computing time is only 6.8 seconds. Finally, with no missing pixels the problem reduces to the denoising problem, and it is solved in 3.2 seconds.

For comparison we also used the script `tv_dode_2D` from Table B.1, which solves the problem in 729.5 seconds using default parameters. The Lagrange multiplier λ was selected such that the two components in (B.23) for `TVinpaint` were slightly smaller than those for `tv_dode_2D`.

Table B.2 lists the computing times, the actual number of iterations, and the upper bound $\mathcal{N}_{\text{inpaint}}$ for the three variations of the inpainting problem. We see that $\mathcal{N}_{\text{inpaint}}$ is indeed an upper bound for the number of iterations, and that it can be very pessimistic if the problem is "easy."

For the **TV deblurring algorithm** the computing times depend on image dimensions, the noise level σ, and the parameters τ and ρ. The performance

Table B.2: Performance studies for inpainting, using our software `TVinpaint` and the script `tv_dode_2D` from Table B.1.

	Time	Its.	$\mathcal{N}_{\text{inpaint}}$
`TVdenoise`			
Inpaint text	28.1 s	751	954
Inpaint circle	6.8 s	190	958
Denoise	3.2 s	93	283
`tv_dode_2D`			
Inpaint text	729.5 s	142	

Table B.3: Performance studies for deblurring of the image in Fig. B.4.

τ	Time	Its.	$\mathcal{N}_{\text{deblur}}$
0.20	15.7 s	174	1767
0.45	13.7 s	152	1766
1.00	19.4 s	222	1764

results for the examples in Fig. B.4, obtained with the default $\rho = 10^{-3}$, are listed in Table B.3. The bound $\mathcal{N}_{\text{deblur}}$ is extremely pessimistic, because it is independent of δ (and thus τ), and we see that the actual number of iterations depends on τ.

It follows from the complexity bound for $\mathcal{N}_{\text{deblur}}$ that the number of iterations also depends on the relative threshold ρ in our rank reduction. Table B.4 reports the performance results for the same deblurring problem as above with varying ρ and fixed $\tau = 0.6$. As expected we see that the computing time depends on ρ. The smaller the ρ the more ill conditioned the problem and therefor the longer the computing time.

The last column shows the relative error $R_\rho = \|x_\rho - x_{10^{-7}}\|_2 / \|x_{10^{-7}}\|_2$ in the solutions for $\rho = 10^{-1}, 10^{-2}, \ldots, 10^{-6}$ compared to the solution for $\rho = 10^{-7}$. Interestingly, the relative error between the reconstructions computed for $\rho = 10^{-3}$ and 10^{-7} is only about 3% (the images are virtually identical to the eye), while there is a factor of almost 10 in computing time. Hence we choose the default value $\rho = 10^{-3}$ to allow fast experiments with the factor τ; when a suitable τ has been found the user may choose a smaller ρ to improve the accuracy of the solution. (For $\rho \geq 10^{-2}$ the rank reduction has a substantial and undesired regularizing effect on the solution.)

We compared our code with the code `FTVdG` from Table B.1, which solves

Table B.4: Performance studies for deblurring when varying the rank reduction threshold ρ and using $\tau = 0.6$.

ρ	Time	Its.	$\mathcal{N}_{\text{deblur}}$	R_ρ
10^{-1}	11.6 s	138	$6.2 \cdot 10^2$	0.042
10^{-2}	11.7 s	134	$6.4 \cdot 10^2$	0.037
10^{-3}	15.6 s	173	$1.7 \cdot 10^3$	0.033
10^{-4}	25.8 s	308	$1.6 \cdot 10^4$	0.028
10^{-5}	48.5 s	552	$1.6 \cdot 10^5$	0.021
10^{-6}	83.0 s	945	$1.7 \cdot 10^6$	0.012
10^{-7}	143.2 s	1574	$1.7 \cdot 10^7$	

the TV deblurring problem by minimizing

$$\mathcal{T}(x) + \frac{1}{2\lambda}\|\widetilde{K}x - b\|_2^2, \tag{B.24}$$

where the matrix $\widetilde{K}$ represents spatial invariant blurring with periodic boundary conditions. Using the default settings, we first select λ such that the TV – ignoring boundary elements – of the `FTVdG` solution $x_{\texttt{FTVdG}}$ is approximately the same as for our solution $x_{\texttt{TVdeblur}}$. The solutions are shown in Fig. B.7 and the corresponding results are summarized in Table B.5, where

$$\widetilde{\mathcal{R}}(x) = \sum_{i=2}^{m-1}\sum_{j=2}^{n-1} \|D_{(ij)}\, x\|_2 + \frac{1}{2\lambda}\|\widetilde{K}x - b\|_2^2.$$

These results demonstrate that although we can reproduce the value of the TV with the default settings of `FTVdG`, we are not able to obtain the same reconstruction, reflected in the fact that $\widetilde{\mathcal{R}}(x_{\texttt{FTVdG}}) > \widetilde{\mathcal{R}}(x_{\texttt{TVdeblur}})$.

The table also shows results for a test with modified `FTVdG` settings $\beta = 2^{16}$ and $\epsilon = 10^{-5}$, cf. [23]. Here we needed to use a slightly different λ such that the above-mentioned TV requirement still holds. Table B.5 shows that even with the modified settings, we are not able to obtain a much better solution as measured by $\widetilde{\mathcal{R}}(x_{\texttt{FTVdG}})$. In fact, it was not possible to adjust the settings for `FTVdG` such that $\widetilde{\mathcal{R}}(x_{\texttt{FTVdG}}) < 1.1\,\widetilde{\mathcal{R}}(x_{\texttt{TVdeblur}})$.

8 Conclusion

Total variation (TV) formulations provide a good basis for reconstruction of noisy, corrupted, and blurred images. In this paper we present easy-to-use public

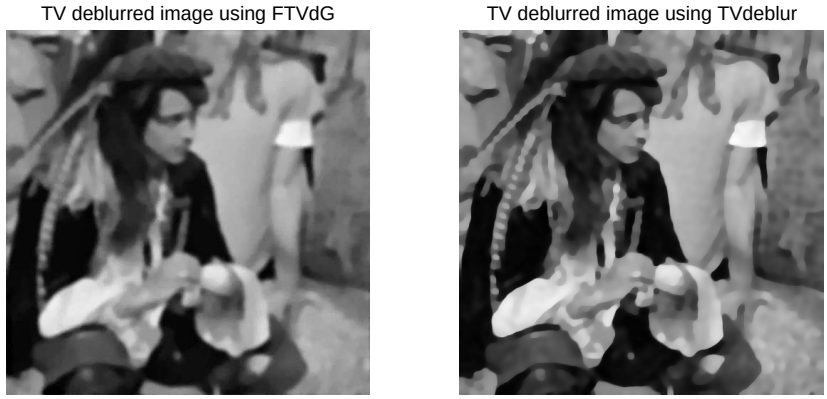


Fig. B.7: Example of TV deblurring of the noisy and blurred image from Fig. B.4 using `FTVdG` (left) and `TVdeblur` (right) with $\rho = 10^{-3}$.

Table B.5: Comparison of the deblurring algorithms `TVdeblur` and `FTVdG`. The parameters are chosen such that the solutions have the same TV equal to 10^6.

	$\|\widetilde{K}x - b\|_2^2$	$\widetilde{\mathcal{R}}(x)$	Time (s)
`TVdeblur`	$4.66 \cdot 10^3$	$1.63 \cdot 10^6$	13.7
`FTVdG` (default)	$5.81 \cdot 10^3$	$1.95 \cdot 10^6$	10.6
`FTVdG` (modified)	$5.79 \cdot 10^3$	$1.99 \cdot 10^6$	27.9

domain software for TV denoising, inpainting, and deblurring, using recently developed first-order optimization algorithms with complexity $O(1/\epsilon)$, where ϵ is the accuracy of the solution. Each iteration in our algorithms only requires moderate computation, of the order $O(mn)$ for denoising and inpainting, and $O(mn \log \max\{m, n\})$ for deblurring. Image deblurring often involves highly ill-conditioned matrices, and to improve both speed and numerical stability we use the technique of rank-reduction for such problems.

Our codes are written in C with Matlab interfaces, and they are available from http://www.netlib.org/numeralgo in the file na28. The codes are robust, user friendly (they require no extra parameters), and they are suited for large problems. The Matlab files have been tested on Matlab versions 7.5–7.8, and they require version 7.5 or later.

Acknowledgements

We wish to thank Michela Redivo Zaglia, Giuseppe Rodriguez, and an anonymous referee for many valuable comments that helped to improve the paper and the software.

Appendix A: The Matlab Functions

TVdenoise

```
X = TVdenoise(B,delta)
[X,info] = TVdenoise(B,delta,eps_rel)
```

This function solves the TV denoising problem

$$\text{minimize } \texttt{TV(X)} \qquad \text{subject to} \qquad \|\texttt{X} - \texttt{B}\|_{\text{F}} \leq \texttt{delta}$$

where `B` is a noisy image, `X` is the reconstruction, and `delta` is an upper bound for the residual norm. The `TV` function is the 1-norm of the gradient magnitude, computed via neighbor pixel differences. At the image borders, we imposed reflexive boundary conditions for the gradient computations.

The parameter `delta` should be of the same size as the norm of the image noise. If the image is $m \times n$, and σ is the standard deviation of the image noise in a pixel, then we recommend to use $\texttt{delta} = \tau\sqrt{mn}\,\sigma$, where τ is slightly smaller than one, say, $\tau = 0.85$.

The function returns an ϵ-optimal solution `X`, meaning that if $\texttt{X}^\star$ is the exact solution, then our solution `X` satisfies

$$\texttt{TV(X)} - \texttt{TV}(\texttt{X}^\star) \leq \epsilon = \max(\texttt{B(:)})\, mn\, \texttt{eps\_rel},$$

where `eps_rel` is a specified relative accuracy (default $\texttt{eps\_rel} = 10^{-3}$). The solution status is returned in the stuct `info`; write `help TVdenoise` for more information.

TVinpaint

```
X = TVinpaint(B,M,delta)
[X,info] = TVinpaint(B,M,delta,eps_rel)
```

This function solves the TV inpainting problem

$$\text{minimize } \texttt{TV(X)} \qquad \text{subject to} \qquad \|\texttt{X(Ic)} - \texttt{B(Ic)}\|_{\text{F}} \leq \texttt{delta}$$

where `B` is a noisy image with missing pixels, `Ic` are the indices to the intact pixels, `X` is the reconstruction, and `delta` is an upper bound for the residual

norm. The `TV` function is the 1-norm of the gradient magnitude, computed via neighbor pixel differences. At the image borders, we imposed reflexive boundary conditions for the gradient computations.

The information about the intact and missing pixels is given in the form of the mask `M`, which is a matrix of the same size as `B`, and whose nonzero elements indicate missing pixels.

The parameter `delta` should be of the same size as the norm of the image noise. If the image is $m \times n$, and σ is the standard deviation of the image noise in a pixel, then we recommend to use $\texttt{delta} = \tau\sqrt{mn}\,\sigma$, where τ is slightly smaller than one, say, $\tau = 0.85$.

The function returns an ϵ-optimal solution `X`, meaning that if $\texttt{X}^\star$ is the exact solution, then our solution `X` satisfies

$$\texttt{TV(X)} - \texttt{TV}(\texttt{X}^\star) \leq \epsilon = \max(\texttt{B(Ic)})\, mn\, \texttt{eps\_rel},$$

where `eps_rel` is the specified relative accuracy (default $\texttt{eps\_rel} = 10^{-3}$). The solution status is returned in the stuct `info`; write `help TVinpaint` for more information.

TVdeblur

```
X = TVdeblur(B,PSF,delta)
[X,info] = TVdeblur(B,PSF,delta,eps_rel,rho,gamma)
```

This function solves the TV deblurring problem

$$\text{minimize } \texttt{TV(X)} \qquad \text{subject to} \qquad \|\texttt{PSF} \star \texttt{X} - \texttt{B}\|_{\mathrm{F}} \leq \delta$$

where `B` is a blurred noisy image, `X` is the reconstruction, and `delta` is an upper bound for the residual norm. The `TV` function is the 1-norm of the gradient magnitude, computed via neighbor pixel differences. At the image borders, we imposed reflexive boundary conditions for the gradient computations.

`PSF`$\star$`X` is the image `X` convolved with the doubly symmetric point spread function `PSF` using reflexive boundary conditions. In the code, the blurring matrix that represents `PSF` is replaced by a rank-deficient well-conditioned approximation obtained by neglecting all eigenvalues smaller than `rho` times the largest eigenvalue (default $\texttt{rho} = 10^{-3}$).

The parameter `delta` should be of the same size as the norm of the image noise. If the image is $m \times n$, and σ is the standard deviation of the image noise in a pixel, then we recommend to use $\delta = \tau\sqrt{mn}\,\sigma$, where τ is smaller than one, say $\tau = 0.55$.

The parameter `gamma` is a an upper bound on the norm of the solution's component in the subspace corresponding to the neglected eigenvalues. The

default value is `gamma` $= \sqrt{mn}\,\max(\texttt{B}(:))$ which should be sufficient for most problems.

The function returns an ϵ-optimal solution `X`, meaning that if $\texttt{X}^\star$ is the exact solution, then our solution `X` satisfies

$$\texttt{TV}(\texttt{X}) - \texttt{TV}(\texttt{X}^\star) \leq \epsilon = \max(\texttt{B}(:))\, mn\, \texttt{eps\_rel},$$

where `eps_rel` is a specified relative accuracy (default `eps_rel` $= 10^{-2}$). The solution status is returned in the stuct `info`; write `help TVdeblur` for more information.

Appendix B: The Norm of the Derivative Matrix

The matrix D defined in Eq. (B.2) can always be written as [15]

$$D = \Pi \begin{pmatrix} I_n \otimes L_m \\ L_n \otimes I_m \end{pmatrix},$$

where Π is a permutation matrix, I_p is the identity matrix of order p, and L_p is the chosen $p \times p$ first-derivative matrix with SVD $L_p = U_p \Sigma_p V_p^T$. We note that since L_p is a sparse matrix with 1 and -1 as the only two nonzero elements per row, it follows that $\|L_p\|_\infty = 2$. The 2-norm of D satisfies $\|D\|_2^2 = \lambda_{\max}(D^T D)$, the largest eigenvalue of $D^T D$, and hence we consider this matrix:

$$\begin{aligned} D^T D &= (I_n \otimes L_m)^T (I_n \otimes L_m) + (L_n \otimes I_m)^T (L_n \otimes I_m) \\ &= I_n \otimes L_m^T L_m + L_n^T L_n \otimes I_m \\ &= V_n V_n^T \otimes V_m \Sigma_m^2 V_m^T + V_n \Sigma_n^2 V_n^T \otimes V_m V_m^T \\ &= (V_n \otimes V_m)(I_n \otimes \Sigma_m^2 + \Sigma_n^2 \otimes I_m)(V_n \otimes V_m)^T \end{aligned}$$

Since the middle matrix is diagonal, it follows that

$$\lambda_{\max}(D^T D) = \lambda_{\max}(\Sigma_m^2) + \lambda_{\max}(\Sigma_n^2) = \|L_m\|_2^2 + \|L_n\|_2^2 \leq \|L_m\|_\infty^2 + \|L_n\|_\infty^2 = 8.$$

We note that a completely different proof is given in [4, Thm. 3.1].

References

[1] F. Alter, S. Durand, and J. Froment, *Adapted total variation for artifact free decompression of JPEG images*, J. Math. Imaging Vision **23**, 199–211 (2005).

[2] J.-F. Aujol, *Some first-order algorithms for total variation based image restoration*, J. Math. Imaging Vision **34**, 307–327 (2009).

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[4] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision **20**, 89–97 (2004).

[5] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 2005.

[6] T. F. Chan, G. H. Golub, and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comp. **20**, 1964–1977 (1999).

[7] P. L. Combettes and T. Pennanen, *Generalized Mann iterates for constructing fixed points in Hilbert spaces*, J. Math. Anal. Appl. **275**, 521–536 (2002).

[8] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.

[9] J. Darbon and M. Sigelle, *Image restoration with discrete constrained total variation. Part I: Fast and exact Optimization.* J. Math. Imaging Vision **26**, 261–276 (2006).

[10] FFTW – freely available C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, available from http://www.fftw.org.

[11] M. Fornasier and C.-B. Schönlieb, *Subspace correction methods for total variation and ℓ_1-minimization*, SIAM J. Numer. Anal. **47**, 3397–3428 (2009).

[12] M. Frigo and S. G. Johnson, *The design and implementation of FFTW3*, Proceedings of the IEEE **93**, 216–231 (2005).

[13] D. Goldfarb and W. Yin, *Second-order cone programming methods for total variation-based image restoration*, SIAM J. Sci. Comput. **27**, 622–645 (2005).

[14] T. Goldstein and S. Osher, *The split Bregman method for L_1 regularized problems*, SIAM J. Imag. Sci. **2**, 323–343 (2009).

[15] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, SIAM, Philadelphia, 2006.

[16] D. Krishnan, L. Ping, and A. M. Yip, *A primal-dual active-set methods for non-negativity constrained total variation deblurring problems*, IEEE Trans. Image Process. **16**, 2766–2777 (2007).

[17] Yu. Nesterov, *Introductory Lectures on Convex Optimization*, Kluwer Academic Publishers, 2004.

[18] Yu. Nesterov, *Smooth minimization of nonsmooth functions*, Math. Program. Ser. A **103**, 127–152 (2005).

[19] Yu. Nesterov, *Excessive gap technique in non-smooth convex optimization*, SIAM J. Optim. **16**, 235–249 (2005).

[20] Yu. Nesterov, *Gradient methods for minimizing composite objective functions*, CORE Discussion Papers series, Université Catholique de Louvain, Center for Operations Research and Econometrics, 2007. Available from http://www.uclouvain.be/en-44660.html

[21] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.

[22] P. Weiss, L. Blanc-Féraud, and G. Aubert, *Efficient schemes for total variation minimization under constraints in image processing*, SIAM J. Sci. Comput. **31**, 2047–2080 (2009).

[23] J. Yang, Y. Zhang, W. Yin, Y. Wang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imag. Sci. **1**, 248–272 (2008).

[24] M. Zhu, S. Wright, and T. F. Chan, *Duality-based algorithms for total-variation-regularized image restoration*, Comput. Optim. Appl. **47:3** 377-400 (2010).

Paper C

Implementation of an Optimal First-Order Method for Strongly Convex Total Variation Regularization

T. L. Jensen, J. H. Jørgensen, P. C. Hansen and S. H. Jensen

This paper is accepted for publication in
BIT Numerical Mathematics

The layout is revised.

Minor spelling, gramma and notation errors have been corrected.

Abstract

We present a practical implementation of an optimal first-order method, due to Nesterov, for large-scale total variation regularization in tomographic reconstruction, image deblurring, etc. The algorithm applies to μ-strongly convex objective functions with L-Lipschitz continuous gradient. In the framework of Nesterov both μ and L are assumed known – an assumption that is seldom satisfied in practice. We propose to incorporate mechanisms to estimate locally sufficient μ and L during the iterations. The mechanisms also allow for the application to non-strongly convex functions. We discuss the convergence rate and iteration complexity of several first-order methods, including the proposed algorithm, and we use a 3D tomography problem to compare the performance of these methods. In numerical simulations we demonstrate the advantage in terms of faster convergence when estimating the strong convexity parameter μ for solving ill-conditioned problems to high accuracy, in comparison with an optimal method for non-strongly convex problems and a first-order method with Barzilai-Borwein step size selection.

1 Introduction

Large-scale discretizations of inverse problems [1] arise in a variety of applications such as medical imaging, non-destructive testing, and geoscience. Due to the inherent instability of these problems, it is necessary to apply regularization in order to compute meaningful reconstructions, and this work focuses on the use of total variation which is a powerful technique when the sought solution is required to have sharp edges (see, e.g., [2, 3] for applications in image reconstruction).

Many total variation algorithms have already been developed, including time marching [3], fixed-point iteration [4], and various minimization-based methods such as sub-gradient methods [5, 6], interior-point methods for second-order cone programming (SOCP) [7], methods exploiting duality [8–10], and graph-cut methods [11, 12].

The numerial difficulty of a problem depends on the linear forward operator. Most methods are dedicated either to denoising, where the operator is simply the identity, or to deblurring where the operator is represented by a fast transform. For general linear operators with no exploitable matrix structure, such as in tomographic reconstruction, the selection of algorithms is not as large. Furthermore, the systems that arise in real-world tomography applications, especially in 3D, are so large that memory-requirements preclude the use of second-order methods with quadratic convergence.

Recently, Nesterov's optimal first-order method [13, 14] has been adapted to, and analyzed for, a number of imaging problems [15, 16]. In [16] it is shown that Nesterov's method outperforms standard first-order methods by an order

of magnitude, but this analysis does not cover tomography problems. A drawback of Nesterov's algorithm (see, e.g., [17]) is the explicit need for the strong convexity parameter and the Lipschitz constant of the objective function, both of which are generally not available in practice.

This paper describes a practical implementation of Nesterov's algorithm, augmented with efficient heuristic methods to estimate the unknown Lipschitz constant and strong convexity parameter. The Lipschitz constant is handled using backtracking, similar to the technique used in [18]. To estimate the unknown strong convexity parameter – which is more difficult – we propose a heuristic based on adjusting an estimate of the strong convexity parameter using a local strong convexity inequality. Furthermore, we equip the heuristic with a restart procedure to ensure convergence in case of an inadequate estimate.

We call the algorithm *UPN* (Unknown Parameter Nesterov) and compare it with two versions of the well-known gradient projection algorithm; *GP:* a simple version using a backtracking line search for the stepsize and *GPBB:* a more advanced version using Barzilai-Borwein stepsize selection [19] and the nonmonotone backtracking procedure from [20].

We also compare with a variant of the proposed algorithm, UPN_0, where the strong convexity information is not enforced. UPN_0 is optimal among first-order methods for the class of Lipschitz smooth, convex (but *not* strongly convex) functions. There are several other variants of optimal first-order methods for Lipschitz smooth problems, see, *e.g.*, [13, 14, 18, 21–25] and the overview in [25, 26], but they all share similar practical convergence [26, §6.1]. We therefore consider UPN_0 to represent this class of methods. We have implemented the four algorithms in C with a MEX interface to MATLAB, and the software is available from `www.imm.dtu.dk/~pch/TVReg/`.

Our numerical tests demonstrate that the proposed method *UPN* is significantly faster than *GP*, as fast as *GPBB* for moderately ill-conditioned problems, and significantly faster for ill-conditioned problems. Compared to UPN_0, *UPN* is consistently faster, when solving to high accuracy.

We start with introductions to the discrete total variation problem, to smooth and strongly convex functions, and to some basic first-order methods in Sections 2, 3, and 4, respectively. Section 5 introduces important inequalities while the new algorithm is described in Section 6. Finally, in Section 7 we report our numerical experiments with the proposed method applied to an image deblurring problem and a tomographic reconstruction problem.

Throughout the paper we use the following notation. The smallest singular value of a matrix A is denoted $\sigma_{\min}(A)$. The smallest and largest eigenvalues of a symmetric semi-definite matrix M are denoted by $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$. For an optimization problem, f is the objective function, $x^\star$ denotes a minimizer, $f^\star = f(x^\star)$ is the optimum objective, and x is called an ϵ-suboptimal solution if $f(x) - f^\star \leq \epsilon$.

2 The Discrete Total Variation Reconstruction Problem

The Total Variation (TV) of a real function $\mathcal{X}(t)$ with $t \in \Omega \subset \mathbb{R}^p$ is defined as

$$\mathcal{T}(\mathcal{X}) = \int_{\Omega} \|\nabla \mathcal{X}(t)\|_2 \, dt. \tag{C.1}$$

Note that the Euclidean norm is not squared, which means that $\mathcal{T}(\mathcal{X})$ is non-differentiable. In order to handle this we consider a smoothed version of the TV functional. Two common choices are to replace the Euclidean norm of the vector z by either $(\|z\|_2^2 + \beta^2)^{1/2}$ or the Huber function

$$\Phi_\tau(z) = \begin{cases} \|z\|_2 - \frac{1}{2}\tau & \text{if} \quad \|z\|_2 \geq \tau, \\ \frac{1}{2\tau}\|z\|_2^2 & \text{else.} \end{cases} \tag{C.2}$$

In this work we use the latter, which can be considered a prox-function smoothing [14] of the TV functional [27]; thus, the approximated TV functional is given by

$$\mathcal{T}_\tau(\mathcal{X}) = \int_{\Omega} \Phi_\tau \left(\nabla \mathcal{X}\right) \, dt. \tag{C.3}$$

In this work we consider the case $t \in \mathbb{R}^3$. To obtain a discrete version of the TV reconstruction problem, we represent $\mathcal{X}(t)$ by an $N = m \times n \times l$ array X, and we let $x = \text{vec}(X)$. Each element or voxel of the array X, with index j, has an associated matrix (a discrete differential operator) $D_j \in \mathbb{R}^{3\times N}$ such that the vector $D_j\, x \in \mathbb{R}^3$ is the forward difference approximation to the gradient at x_j. By stacking all D_j we obtain the matrix D of dimensions $3N \times N$:

$$D = \begin{pmatrix} D_1 \\ \vdots \\ D_N \end{pmatrix}. \tag{C.4}$$

We use periodic boundary conditions in D, which ensures that only a constant x has a TV of 0. Other choices of boundary conditions could easily be implemented.

When the discrete approximation to the gradient is used and the integration in (C.3) is replaced by summations, the discrete and smoothed TV function is given by

$$T_\tau(x) = \sum_{j=1}^{N} \Phi_\tau(D_j x). \tag{C.5}$$

The gradient $\nabla T_\tau(x) \in \mathbb{R}^N$ of this function is given by

$$\nabla T_\tau(x) = \sum_{j=1}^{N} D_j^T D_j\, x / \max\{\tau, \|D_j x\|_2\}. \tag{C.6}$$

We assume that the sought reconstruction has voxel values in the range $[0, 1]$, so we wish to solve a bound-constrained problem, i.e., having the feasible region $\mathcal{Q} = \{x \in \mathbb{R}^N \mid 0 \le x_j \le 1 \,\forall\, j\}$. Given a linear system $A\,x \approx b$ where $A \in \mathbb{R}^{M\times N}$ and $N = mnl$, we define the associated *discrete TV regularization problem* as

$$x^\star = \operatorname*{argmin}_{x\in\mathcal{Q}} \ \phi(x), \qquad \phi(x) = \tfrac{1}{2}\|A\,x - b\|_2^2 + \alpha\, T_\tau(x), \tag{C.7}$$

where $\alpha > 0$ is the TV regularization parameter. This is the problem we want to solve, for the case where the linear system of equations arises from discretization of an inverse problem.

3 Smooth and Strongly Convex Functions

To set the stage for the algorithm development in this paper, we consider the convex optimization problem $\min_{x\in\mathcal{Q}} f(x)$ where f is a convex function and $\mathcal{Q}$ is a convex set. We recall that a continuously differentiable function f is convex if

$$f(x) \ge f(y) + \nabla f(y)^T (x - y), \qquad \forall\, x, y \in \mathbb{R}^N. \tag{C.8}$$

Definition 3.1. *A continuously differentiable convex function f is said to be* strongly convex *with* strong convexity parameter μ *if there exists a $\mu > 0$ such that*

$$f(x) \ge f(y) + \nabla f(y)^T (x - y) + \tfrac{1}{2}\mu\|x - y\|_2^2, \qquad \forall\, x, y \in \mathbb{R}^N. \tag{C.9}$$

Definition 3.2. *A continuously differentiable convex function f has* Lipschitz continuous *gradient with* Lipschitz constant L*, if*

$$f(x) \le f(y) + \nabla f(y)^T (x - y) + \tfrac{1}{2}L\|x - y\|_2^2, \qquad \forall\, x, y \in \mathbb{R}^N. \tag{C.10}$$

Remark 3.3. *The condition* (C.10) *is equivalent [13, Theorem 2.1.5] to the more standard way of defining Lipschitz continuity of the gradient, namely, through convexity and the condition $\|\nabla f(x) - \nabla f(y)\|_2 \le L\|x - y\|_2, \forall\, x, y \in \mathbb{R}^N$.*

Remark 3.4. *Lipschitz continuity of the gradient is a smoothness requirement on f. A function f that satisfies* (C.10) *is said to be smooth, and L is also known as the* smoothness constant*.*

The set of functions that satisfy (C.9) and (C.10) is denoted $\mathcal{F}_{\mu,L}$. It is clear that $\mu \leq L$ and also that if $\mu_1 \geq \mu_0$ and $L_1 \leq L_0$ then $f \in \mathcal{F}_{\mu_1,L_1} \Rightarrow f \in \mathcal{F}_{\mu_0,L_0}$. Given fixed choices of μ and L, we introduce the ratio $Q = L/\mu$ (sometimes referred to as the "modulus of strong convexity" [28] or the "condition number for f" [13]) which is an upper bound for the condition number of the Hessian matrix. The number Q plays a major role for the convergence rate of the optimization methods we will consider.

Lemma 3.5. *For the quadratic function* $f(x) = \frac{1}{2}\|A\,x - b\|_2^2$ *with* $A \in \mathbb{R}^{M\times N}$ *we have*

$$L = \|A\|_2^2, \qquad \mu = \lambda_{\min}(A^T A) = \begin{cases} \sigma_{\min}(A)^2 & \text{if} \quad \operatorname{rank}(A) = N, \\ 0, & \text{else}, \end{cases} \tag{C.11}$$

and if $\operatorname{rank}(A) = N$ *then* $Q = \kappa(A)^2$*, the square of the condition number of* A*.*

Proof. Follows from $f(x) = f(y) + (A\,y - b)^T A(x-y) + \frac{1}{2}(x-y)^T A^T A(x-y)$, the second order Taylor expansion of f about y, where equality holds for quadratic f. ■

Lemma 3.6. *For the smoothed TV function* (C.5) *we have*

$$L = \|D\|_2^2/\tau, \qquad \mu = 0, \tag{C.12}$$

where $\|D\|_2^2 \leq 12$ *in the 3D case.*

Proof. The result for L follows from [14, Thm. 1] since the smoothed TV functional can be written as [15, 27]

$$T_\tau(x) = \max_u \left\{ u^T Dx - \frac{\tau}{2}\,\|u\|_2^2 \,:\, \|u_i\|_2 \leq 1,\ \forall\, i = 1,\ldots,N \right\}$$

with $u = (u_1^T, \ldots, u_N^T)^T$ stacked according to D. The inequality $\|D\|_2^2 \leq 12$ follows from a straightforward extension of the proof in the Appendix of [15]. For μ pick $y = \alpha e \in \mathbb{R}^N$ and $x = \beta e \in \mathbb{R}^N$, where $e = (1,\ldots,1)^T$, and $\alpha \neq \beta \in \mathbb{R}$. Then we get $T_\tau(x) = T_\tau(y) = 0$, $\nabla T_\tau(y) = 0$ and obtain

$$\tfrac{1}{2}\mu\|x-y\|_2^2 \leq T_\tau(x) - T_\tau(y) - \nabla T_\tau(y)^T(x-y) = 0,$$

and hence $\mu = 0$. ■

Theorem 3.7. *For the function* $\phi(x)$ *defined in* (C.7) *we have a strong convexity parameter* $\mu = \lambda_{\min}(A^T A)$ *and Lipschitz constant* $L = \|A\|_2^2 + \alpha\,\|D\|_2^2/\tau$*. If* $\operatorname{rank}(A) < N$ *then* $\mu = 0$*, otherwise* $\mu = \sigma_{\min}(A)^2 > 0$ *and*

$$Q = \kappa(A)^2 + \frac{\alpha}{\tau}\,\frac{\|D\|_2^2}{\sigma_{\min}(A)^2}, \tag{C.13}$$

where $\kappa(A) = \|A\|_2/\sigma_{\min}(A)$ *is the condition number of* A*.*

Proof. Assume $\text{rank}(A) = N$ and consider $f(x) = g(x) + h(x)$ with $g \in \mathcal{F}_{\mu_g, L_g}$ and $h \in \mathcal{F}_{\mu_h, L_h}$. Then $f \in \mathcal{F}_{\mu_f, L_f}$, where $\mu_f = \mu_g + \mu_h$ and $L_f = L_g + L_h$. From μ_f and L_f and using Lemmas 3.5 and 3.6 with $g(x) = \frac{1}{2}\|A\,x - b\|_2^2$ and $h(x) = \alpha T_\tau(x)$ we obtain the condition number for ϕ given in (C.13). If $\text{rank}(A) < N$ then the matrix $A^T A$ has at least one zero eigenvalue, and thus $\mu = 0$. ■

Remark 3.8. *Due to the inequalities used to derive* (C.13)*, there is no guarantee that the given μ and L are the tightest possible for ϕ.*

4 Some Basic First-Order Methods

A basic first-order method is the gradient projection method of the form

$$x^{(k+1)} = P_{\mathcal{Q}}\left(x^{(k)} - p_k \nabla f(x^{(k)})\right), \qquad k = 0, 1, 2, \ldots\,. \tag{C.14}$$

where $P_{\mathcal{Q}}$ is the Euclidean projection onto the convex set $\mathcal{Q}$ [13]. The following theorem summarizes the convergence properties.

Theorem 4.1. *Let $f \in \mathcal{F}_{\mu,L}$, $p_k = 1/L$ and $x^\star \in \mathcal{Q}$ be the constrained minimizer of f, then for the gradient projection method* (C.14) *we have*

$$f(x^{(k)}) - f^\star \leq \frac{L}{2k}\|x^{(0)} - x^\star\|_2^2. \tag{C.15}$$

Moreover, if $\mu \neq 0$ then

$$f(x^{(k)}) - f^\star \leq \left(1 - \frac{\mu}{L}\right)^k \left(f(x^{(0)}) - f^\star\right). \tag{C.16}$$

Proof. The two bounds follow from [29] and [28, §7.1.4], respectively. ■

To improve the convergence of the gradient (projection) method, Barzilai and Borwein [19] suggested a scheme in which the step $p_k \nabla f(x^{(k)})$ provides a simple and computationally cheap approximation to the Newton step $(\nabla^2 f(x^{(k)}))^{-1} \nabla f(x^{(k)})$. For general unconstrained problems with $f \in \mathcal{F}_{\mu,L}$, possibly with $\mu = 0$, non-monotone line search combined with the Barzilai-Borwein (BB) strategy produces algorithms that converge [30]; but it is difficult to give a precise iteration complexity for such algorithms. For strictly quadratic unconstrained problems the BB strategy requires $\mathcal{O}\big(Q \log \epsilon^{-1}\big)$ iterations to obtain an ϵ-suboptimal solution [31]. In [32] it was argued that, in practice, $\mathcal{O}\big(Q \log \epsilon^{-1}\big)$ iterations "is the best that could be expected". This comment is also supported by the statement in [13, p. 69] that all "reasonable step-size rules" have the same iteration complexity as the standard gradient method. Note that the classic gradient method (C.14) has $\mathcal{O}(L/\epsilon)$ complexity for $f \in \mathcal{F}_{0,L}$. To summarize, when

Algorithm 1: *GPBB*

input : $x^{(0)}$, K
output: $x^{(k+1)}$

$p_0 = 1$;
for $k = 0, 1, 2, \ldots$ **do**
  `// BB strategy`
  if $k > 0$ **then**
    $p_k \leftarrow \frac{\|x^{(k)} - x^{(k-1)}\|_2^2}{\left(x^{(k)} - x^{(k-1)}\right)^T \left(\nabla f(x^{(k)}) - \nabla f(x^{(k-1)})\right)}$;
  $\beta \leftarrow 0.95$;
  $\bar{x} \leftarrow P_{\mathcal{Q}}(x^{(k)} - p_k \nabla f(x^{(k)}))$;
  $\hat{f} \leftarrow \max\{f(x^{(k)}), f(x^{(k-1)}), \ldots, f(x^{(k-K)})\}$;
  while $f(\bar{x}) \geq \hat{f} - \sigma \nabla f(x^{(k)})^T (x^{(k)} - \bar{x})$ **do**
    $\beta \leftarrow \beta^2$;
    $\bar{x} \leftarrow P_{\mathcal{Q}}(x^{(k)} - \beta p_k \nabla f(x^{(k)}))$;
  $x^{(k+1)} \leftarrow \bar{x}$;

using the BB strategy we should not expect better complexity than $\mathcal{O}(L/\epsilon)$ for $f \in \mathcal{F}_{0,L}$, and $\mathcal{O}\big(Q \log \epsilon^{-1}\big)$ for $f \in \mathcal{F}_{\mu,L}$.

In Algorithm 1 we give the (conceptual) algorithm *GPBB*, which implements the BB strategy with non-monotone line search [33, 34] using the backtracking procedure from [20] (initially combined in [30]). The algorithm needs the real parameter $\sigma \in [0, 1]$ and the nonnegative integer K, the latter specifies the number of iterations over which an objective decrease is guaranteed.

An alternative approach is to consider first-order methods with optimal complexity. The optimal complexity is defined as the worst-case complexity for a first-order method applied to any problem in a certain class [13, 28] (there are also more technical aspects involving the problem dimensions and a black-box assumption). In this paper we focus on the classes $\mathcal{F}_{0,L}$ and $\mathcal{F}_{\mu,L}$.

Recently there has been a great deal of interest in optimal first-order methods for convex optimization problems with $f \in \mathcal{F}_{0,L}$ [25, 35]. For this class it is possible to reach an ϵ-suboptimal solution within $\mathcal{O}(\sqrt{L/\epsilon})$ iterations. Nesterov's methods can be used as stand-alone optimization algorithm, or in a composite objective setup [18, 24, 25], in which case they are called accelerated methods (because the designer violates the black-box assumption). Another option is to apply optimal first-order methods to a smooth approximation of a non-smooth function leading to an algorithm with $\mathcal{O}(1/\epsilon)$ complexity [14]; for practical considerations, see [15, 27].

Algorithm 2: *Nesterov*

input : $x^{(0)}$, μ, L, θ_0
output: $x^{(k+1)}$

$y^{(0)} \leftarrow x^{(0)}$;
for $k = 0, 1, 2, \ldots$ **do**
  $x^{(k+1)} \leftarrow P_{\mathcal{Q}}\big(y^{(k)} - L^{-1}\nabla f(y^{(k)})\big)$;
  $\theta_{k+1} \leftarrow$ positive root of $\theta^2 = (1-\theta)\theta_k^2 + \frac{\mu}{L}\theta$;
  $\beta_k \leftarrow \theta_k(1-\theta_k)/(\theta_k^2 + \theta_{k+1})$;
  $y^{(k+1)} \leftarrow x^{(k+1)} + \beta_k(x^{(k+1)} - x^{(k)})$;

Optimal methods specific for the function class $\mathcal{F}_{\mu,L}$ with $\mu > 0$ are also known [13, 23]; see also [24] for the composite objective version. However, these methods have gained little practical consideration; for example in [24] all the simulations are conducted with $\mu = 0$. Optimal methods require $\mathcal{O}\big(\sqrt{Q}\log\epsilon^{-1}\big)$ iterations while the classic gradient method requires $\mathcal{O}\big(Q\log\epsilon^{-1}\big)$ iterations [13, 28]. For quadratic problems, the conjugate gradient method achieves the same iteration complexity as the optimal first-order method [28].

In Algorithm 2 we state the basic optimal method *Nesterov* [13] with known μ and L; it requires an initial $\theta_0 \geq \sqrt{\mu/L}$. Note that it uses two sequences of vectors, $x^{(k)}$ and $y^{(k)}$. The convergence rate is provided by the following theorem.

Theorem 4.2. *If $f \in \mathcal{F}_{\mu,L}$, $1 > \theta_0 \geq \sqrt{\mu/L}$, and $\gamma_0 = \frac{\theta_0(\theta_0 L - \mu)}{1-\theta_0}$, then for algorithm Nesterov we have*

$$f(x^{(k)}) - f^\star \leq \frac{4L}{(2\sqrt{L} + k\sqrt{\gamma_0})^2}\left(f(x^{(0)}) - f^\star + \frac{\gamma_0}{2}\|x^{(0)} - x^\star\|_2^2\right). \tag{C.17}$$

Moreover, if $\mu \neq 0$, then

$$f(x^{(k)}) - f^\star \leq \left(1 - \sqrt{\frac{\mu}{L}}\right)^k \left(f(x^{(0)}) - f^\star + \frac{\gamma_0}{2}\|x^{(0)} - x^\star\|_2^2\right). \tag{C.18}$$

Proof. See [13, (2.2.19), Theorem 2.2.3] and Appendix 8 for an alternative proof. ■

Except for different constants Theorem 4.2 mimics the result in Theorem 4.1, with the crucial differences that the denominator in (C.17) is squared and μ/L in (C.18) has a square root. Comparing the convergence rates in Theorems 4.1 and 4.2, we see that the rates are linear but differ in the linear rate, $(1 - Q^{-1})$ and $(1 - \sqrt{Q^{-1}})$, respectively. For ill-conditioned problems, it is important

whether the complexity is a function of Q or $\sqrt{Q}$, see, e.g., [28, §7.2.8], [21]. This motivates the interest in specialized optimal first-order methods for solving ill-conditioned problems.

5 First-Order Inequalities for the Gradient Map

For unconstrained convex problems the (norm of the) gradient is a measure of how close we are to the minimum, through the first-order optimality condition, cf. [36]. For constrained convex problems $\min_{x\in\mathcal{Q}} f(x)$ there is a similar quantity, namely, the *gradient map* defined by

$$G_\nu(x) = \nu\left(x - P_{\mathcal{Q}}\left(x - \nu^{-1}\nabla f(x)\right)\right). \tag{C.19}$$

Here $\nu > 0$ is a parameter and ν^{-1} can be interpreted as the step size of a gradient step. The gradient map is a generalization of the gradient to constrained problems in the sense that if $\mathcal{Q} = \mathbb{R}^N$ then $G_\nu(x) = \nabla f(x)$, and the equality $G_\nu(x^\star) = 0$ is a necessary and sufficient optimality condition [29]. In what follows we review and derive some important first-order inequalities which will be used to analyze the proposed algorithm. We start with a rather technical result.

Lemma 5.1. *Let $f \in \mathcal{F}_{\mu,L}$, fix $x \in \mathcal{Q}$, $y \in \mathbb{R}^N$, and set $x^+ = P_{\mathcal{Q}}(y - \bar{L}^{-1}\nabla f(y))$, where $\bar{\mu}$ and $\bar{L}$ are related to x, y and x^+ by the inequalities*

$$f(x) \geq f(y) + \nabla f(y)^T(x-y) + \tfrac{1}{2}\bar{\mu}\|x-y\|_2^2, \tag{C.20}$$

$$f(x^+) \leq f(y) + \nabla f(y)^T(x^+-y) + \tfrac{1}{2}\bar{L}\|x^+-y\|_2^2. \tag{C.21}$$

Then

$$f(x^+) \leq f(x) + G_{\bar{L}}(y)^T(y-x) - \tfrac{1}{2}\bar{L}^{-1}\|G_{\bar{L}}(y)\|_2^2 - \tfrac{1}{2}\bar{\mu}\|y-x\|_2^2. \tag{C.22}$$

Proof. Follows directly from [13, Theorem 2.2.7]. ■

Note that if $f \in \mathcal{F}_{\mu,L}$, then in Lemma 5.1 we can always select $\bar{\mu} = \mu$ and $\bar{L} = L$ to ensure that the inequalities (C.20) and (C.21) are satisfied. However, for specific x, y and x^+, there can exist $\bar{\mu} \geq \mu$ and $\bar{L} \leq L$ such that (C.20) and (C.21) hold. We will use these results to design an algorithm for unknown parameters μ and L.

The lemma can be used to obtain the following lemma. The derivation of the bounds is inspired by similar results for composite objective functions in [24], and the second result is similar to [13, Corollary 2.2.1].

Lemma 5.2. *Let $f \in \mathcal{F}_{\mu,L}$, fix $y \in \mathbb{R}^N$, and set $x^+ = P_{\mathcal{Q}}(y - \bar{L}^{-1}\nabla f(y))$. Let $\bar{\mu}$ and $\bar{L}$ be selected in accordance with* (C.20) *and* (C.21) *respectively. Then*

$$\tfrac{1}{2}\bar{\mu}\|y - x^\star\|_2 \leq \|G_{\bar{L}}(y)\|_2. \tag{C.23}$$

If $y \in \mathcal{Q}$ then

$$\tfrac{1}{2}\bar{L}^{-1}\|G_{\bar{L}}(y)\|_2^2 \leq f(y) - f(x^+) \leq f(y) - f^\star. \tag{C.24}$$

Proof. From Lemma 5.1 with $x = x^\star$ we use $f(x^+) \geq f^\star$ and obtain

$$\tfrac{1}{2}\bar{\mu}\|y - x^\star\|_2^2 \leq G_{\bar{L}}(y)^T(y - x^\star) - \tfrac{1}{2}\bar{L}^{-1}\|G_{\bar{L}}(y)\|_2^2 \leq \|G_{\bar{L}}(y)\|_2\|y - x^\star\|_2,$$

and (C.23) follows; Eq. (C.24) follows from Lemma 5.1 using $y = x$ and $f^\star \leq f(x^+)$. ■

As mentioned in the beginning of the section, the results of the corollary say that we can relate the norm of the gradient map at y to the error $\|y - x^*\|_2$ as well as to $f(y) - f^*$. This motivates the use of the gradient map in a stopping criterion:

$$\|G_{\bar{L}}(y)\|_2 \leq \bar{\epsilon}, \tag{C.25}$$

where y is the current iterate, and $\bar{L}$ is linked to this iterate using (C.21). The parameter $\bar{\epsilon}$ is a user-specified tolerance based on the requested accuracy. Lemma 5.2 is also used in the following section to develop a restart criterion to ensure convergence.

6 Nesterov's Method With Parameter Estimation

The parameters μ and L are explicitly needed in *Nesterov*. In case of an unregularized least-squares problem we can in principle compute μ and L as the smallest and largest squared singular value of A, though it might be computational expensive. When a regularization term is present it is unclear whether the tight μ and L can be computed at all. Bounds can be obtained using the result in Theorem 3.7.

A practical approach is to estimate μ and L during the iterations. To this end, we introduce the estimates μ_k and L_k of μ and L in each iteration k. We discuss first how to choose L_k, then μ_k, and finally we state the complete algorithm *UPN* and its convergence properties.

To ensure convergence, the main inequalities (C.43) and (C.44) must be satisfied. Hence, according to Lemma 5.1 we need to choose L_k such that

$$f(x^{(k+1)}) \leq f(y^{(k)}) + \nabla f(y^{(k)})^T(x^{(k+1)} - y^{(k)}) + \tfrac{1}{2}L_k\|x^{(k+1)} - y^{(k)}\|_2^2. \tag{C.26}$$

Algorithm 3: *BT*

input : $y, \bar{L}$
output: $x, \tilde{L}$
$\tilde{L} \leftarrow \bar{L}$;
$x \leftarrow P_{\mathcal{Q}}\left(y - \tilde{L}^{-1}\nabla f(y)\right)$;
while $f(x) > f(y) + \nabla f(y)^T(x-y) + \frac{1}{2}\tilde{L}\|x-y\|_2^2$ **do**
  $\tilde{L} \leftarrow \rho_L \tilde{L}$;
  $x \leftarrow P_{\mathcal{Q}}\left(y - \tilde{L}^{-1}\nabla f(y)\right)$;

This is easily accomplished using *backtracking* on L_k [18]. The scheme, *BT*, takes the form given in Algorithm 3, where $\rho_L > 1$ is an adjustment parameter. If the loop is executed n_{UPNBT} times, the dominant computational cost of *BT* is $n_{UPNBT} + 2$ function evaluations and 1 gradient evaluation.

For choosing the estimate μ_k we introduce the auxilliary variable $\mu_k^\star$ as the value that causes Definition 3.1 (of strong convexity) for $x^\star$ and $y^{(k)}$ to hold with equality

$$f(x^\star) = f(y^{(k)}) + \nabla f(y^{(k)})^T(x^\star - y^{(k)}) + \tfrac{1}{2}\mu_k^\star\|x^\star - y^{(k)}\|_2^2. \tag{C.27}$$

From (C.44) with Lemma 5.1 and (C.45) we find that we must choose $\mu_k \leq \mu_k^\star$ to obtain a convergent algorithm. However, as $x^\star$ is, of course, unknown, this task is not straightforward, if at all possible. Instead, we propose a *heuristic* where we select μ_k such that

$$f(x^{(k)}) \geq f(y^{(k)}) + \nabla f(y^{(k)})^T(x^{(k)} - y^{(k)}) + \tfrac{1}{2}\mu_k\|x^{(k)} - y^{(k)}\|_2^2. \tag{C.28}$$

This is indeed possible since $x^{(k)}$ and $y^{(k)}$ are known iterates. Furthermore, we want the estimate μ_k to be decreasing in order to approach a better estimate of μ. This can be achieved by the choice

$$\mu_k = \min\{\mu_{k-1}, M(x^{(k)}, y^{(k)})\}, \tag{C.29}$$

where we have defined the function

$$M(x,y) = \begin{cases} \frac{f(x) - f(y) - \nabla f(y)^T(x-y)}{\frac{1}{2}\|x-y\|_2^2} & \text{if} \quad x \neq y, \\ \infty & \text{else.} \end{cases} \tag{C.30}$$

In words, the heuristic chooses the largest μ_k that satisfies (C.9) for $x^{(k)}$ and $y^{(k)}$, as long as μ_k is not larger than μ_{k-1}. The heuristic is simple and computationally inexpensive and we have found that it is effective for determining

a useful estimate. Unfortunately, convergence of *Nesterov* equipped with this heuristic is not guaranteed, since the estimate can be too large. To ensure convergence we include a restart procedure *RUPN* that detects if μ_k is too large, inspired by the approach in [24, §5.3] for composite objectives. *RUPN* is given in Algorithm 4.

To analyze the restart strategy, assume that μ_i for all $i = 1, \ldots, k$ are *small enough*, *i.e.*, they satisfy $\mu_i \leq \mu_i^\star$ for $i = 1, \ldots, k$, and μ_k satisfies

$$f(x^\star) \geq f(x^{(0)}) + \nabla f(x^{(0)})^T (x^\star - x^{(0)}) + \tfrac{1}{2}\mu_k \|x^\star - x^{(0)}\|_2^2. \tag{C.31}$$

When this holds we have the convergence result (using (C.46))

$$f(x^{(k+1)}) - f^\star \leq \prod_{i=1}^{k} \left(1 - \sqrt{\mu_i / L_i}\right) \left(f(x^{(1)}) - f^\star + \tfrac{1}{2}\gamma_1 \|x^{(1)} - x^\star\|_2^2\right). \tag{C.32}$$

We start from iteration $k = 1$ for reasons which will presented shortly (see Appendix 8 for details and definitions). If the algorithm uses a projected gradient step from the initial $x^{(0)}$ to obtain $x^{(1)}$, the rightmost factor of (C.32) can be bounded as

$$\begin{aligned} & f(x^{(1)}) - f^\star + \tfrac{1}{2}\gamma_1 \|x^{(1)} - x^\star\|_2^2 \\ & \quad \leq \quad G_{L_0}(x^{(0)})^T (x^{(0)} - x^\star) - \tfrac{1}{2} L_0^{-1} \|G_{L_0}(x^{(0)})\|_2^2 + \tfrac{1}{2}\gamma_1 \|x^{(1)} - x^\star\|_2^2 \\ & \quad \leq \quad \|G_{L_0}(x^{(0)})\|_2 \|x^{(0)} - x^\star\|_2 - \tfrac{1}{2} L_0^{-1} \|G_{L_0}(x^{(0)})\|_2^2 + \tfrac{1}{2}\gamma_1 \|x^{(0)} - x^\star\|_2^2 \\ & \quad \leq \quad \left(\frac{2}{\mu_k} - \frac{1}{2L_0} + \frac{2\gamma_1}{\mu_k^2}\right) \|G_{L_0}(x^{(0)})\|_2^2. \end{aligned} \tag{C.33}$$

Here we used Lemma 5.1, and the fact that a projected gradient step reduces the Euclidean distance to the solution [13, Theorem 2.2.8]. Using Lemma 5.2 we arrive at the bound

$$\tfrac{1}{2}\tilde{L}_{k+1}^{-1} \|G_{\tilde{L}_{k+1}}(x^{(k+1)})\|_2^2 \leq \prod_{i=1}^{k} \left(1 - \sqrt{\frac{\mu_i}{L_i}}\right) \left(\frac{2}{\mu_k} - \frac{1}{2L_0} + \frac{2\gamma_1}{\mu_k^2}\right) \|G_{L_0}(x^{(0)})\|_2^2, \tag{C.34}$$

where $\tilde{L}_{k+1}$ is defined in Algorithm *UPN*. If the algorithm detects that (C.34) is not satisfied, it can only be because there was at least one μ_i for $i = 1, \ldots, k$ which was *not small enough*. If this is the case, we restart the algorithm with a new $\bar{\mu} \leftarrow \rho_\mu \mu_k$, where $0 < \rho_\mu < 1$ is a parameter, using the current iterate $x^{(k+1)}$ as initial vector.

The complete algorithm *UPN* (Unknown-Parameter Nesterov) is given in Algorithm 5. *UPN* is based on Nesterov's optimal method where we have included backtracking on L_k and the heuristic (C.29). An initial vector $x^{(0)}$ and initial parameters $\bar{\mu} \geq \mu$ and $\bar{L} \leq L$ must be specified along with the requested accuracy $\bar{\epsilon}$. The changes from *Nesterov* to *UPN* are at the following lines:

Algorithm 4: *RUPN*

$\gamma_1 = \theta_1(\theta_1 L_1 - \mu_1)/(1-\theta_1)$;
if $\mu_k \neq 0$ **and** inequality (C.34) not satisfied **then**
  abort execution of *UPN*;
  restart *UPN* with input $(x^{(k+1)},\ \rho_\mu \mu_k,\ L_k,\ \bar{\epsilon})$;

Algorithm 5: *UPN*

input : $x^{(0)}, \bar{\mu}, \bar{L}, \bar{\epsilon}$
output: $x^{(k+1)}$ or $\tilde{x}^{(k+1)}$
$[x^{(1)}, L_0] \leftarrow BT(x^{(0)}, \bar{L})$;
$\mu_0 = \bar{\mu}, \quad y^{(1)} \leftarrow x^{(1)}, \quad \theta_1 \leftarrow \sqrt{\mu_0/L_0}$;
for $k = 1, 2, \ldots$ **do**
  $[x^{(k+1)}, L_k] \leftarrow BT(y^{(k)}, L_{k-1})$;
  $[\tilde{x}^{(k+1)}, \tilde{L}_{k+1}] \leftarrow BT(x^{(k+1)}, L_k)$;
  if $\|G_{\tilde{L}_{k+1}}(x^{(k+1)})\|_2 \leq \bar{\epsilon}$ **then abort**, **return** $\tilde{x}^{(k+1)}$;
  if $\|G_{L_k}(y^{(k)})\|_2 \leq \bar{\epsilon}$ **then abort**, **return** $x^{(k+1)}$;
  $\mu_k \leftarrow \min\{\mu_{k-1}, M(x^{(k)}, y^{(k)})\}$;
  RUPN;
  $\theta_{k+1} \leftarrow$ positive root of $\theta^2 = (1-\theta)\theta_k^2 + (\mu_k/L_k)\,\theta$;
  $\beta_k \leftarrow \theta_k(1-\theta_k)/(\theta_k^2 + \theta_{k+1})$;
  $y^{(k+1)} \leftarrow x^{(k+1)} + \beta_k(x^{(k+1)} - x^{(k)})$;

1: Initial projected gradient step to obtain the bound (C.33) and thereby the bound (C.34) used for the restart criterion.

5: Extra projected gradient step explicitly applied to obtain the stopping criterion $\|G_{\tilde{L}_{k+1}}(x^{(k+1)})\|_2 \leq \bar{\epsilon}$.

6,7: Used to relate the stopping criterion in terms of $\bar{\epsilon}$ to ϵ, see Appendix 8.

8: The heuristic choice of μ_k in (C.29).

9: The restart procedure for inadequate estimates of μ.

We note that in a practical implementation, the computational work involved in one iteration step of *UPN* may – in the worst case situation – be twice that of one iteration of *GPBB*, due to the two calls to *BT*. However, it may be possible to implement these two calls more efficiently than naively calling *BT* twice. We will instead focus on the iteration complexity of *UPN* given in the following theorem.

Theorem 6.1. *Algorithm UPN, applied to $f \in \mathcal{F}_{\mu,L}$ under conditions $\bar{\mu} \geq \mu$, $\bar{L} \leq L$, $\bar{\epsilon} = \sqrt{(\mu/2)\,\epsilon}$, stops using the gradient map magnitude measure and returns an ϵ-suboptimal solution with iteration complexity*

$$\mathcal{O}\Big(\sqrt{Q}\log Q\Big) + \mathcal{O}\Big(\sqrt{Q}\log \epsilon^{-1}\Big)\,. \tag{C.35}$$

Proof. See Appendix 8. ■

The term $\mathcal{O}\big(\sqrt{Q}\log Q\big)$ in (C.35) follows from application of several inequalities involving the problem dependent parameters μ and L to obtain the overall bound (C.34). Algorithm *UPN* is suboptimal since the optimal complexity is $\mathcal{O}\big(\sqrt{Q}\log \epsilon^{-1}\big)$ but it has the advantage that it can be applied to problems with unknown μ and L.

7 Numerical Experiments

7.1 An Image Deblurring Example

We exemplify the use of the algorithm *UPN* to solve a total variation regularized image deblurring problem, where the goal is to determine a sharp image x from a blurred and noisy one $b = Ax + e$. The matrix A models linear motion blur, which renders A sparse, and we use reflexive boundary conditions. For this type of blur no fast transform can be exploited. We add Gaussian noise e with relative noise level $\|e\|_2/\|b\|_2 = 0.01$ and reconstruct using $\alpha = 5.0$ and the default setting of $\tau = 10^{-4} \cdot 255$, where $[0, 255]$ is the dynamic pixel intensity range. The result is shown in Fig. C.1. We recognize well-known features of TV-regularized reconstructions: Sharp edges are well-preserved, while fine texture has been over-regularized and has a "patchy" appearance.

To investigate the convergence of the methods, we need the true minimizer $x^\star$ with $\phi(x^\star) = \phi^\star$, which is unknown for the test problem. However, for comparison it is enough to use a reference solution much closer to the true minimizer than the iterates. Thus, to compare the accuracy of the solutions obtained with the accuracy parameter $\bar{\epsilon}$, we use a reference solution computed with accuracy $(\bar{\epsilon} \cdot 10^{-2})$, and with abuse of notation we use $x^\star$ to denote this reference solution.

In Fig. C.1 both *UPN* and UPN_0 are seen to be faster than *GP* and *GPBB*, and for a high-accuracy solution *UPN* also outperforms UPN_0. For *UPN*, *GP* and *GPBB* we observe linear rates of convergence, but *UPN* converges much faster. UPN_0 shows a sublinear convergence rate, however the initial phase is steep enough that it takes *UPN* almost 1000 iterations to catch up. We note that the potential of *UPN* seems to be in the case where a high-accuracy solution is needed.

Having demonstrated the performance of the proposed algorithm in an image deblurring problem, we focus in the remainder on a 3D tomography test problem, for which we further study the convergence behavior including the influence of the regularization and smoothing parameters.

7.2 Experiments with 3D Tomographic Reconstruction

Tomography problems arise in numerous areas, such as medical imaging, nondestructive testing, materials science, and geophysics [37–39]. These problems amount to reconstructing an object from its projections along a number of specified directions, and these projections are produced by X-rays, seismic waves, or other "rays" penetrating the object in such a way that their intensity is partially absorbed by the object. The absorbtion thus gives information about the object.

The following generic model accounts for several applications of tomography. We consider an object in 3D with linear attenuation coefficient $\mathcal{X}(t)$, with $t \in \Omega \subset \mathbb{R}^3$. The intensity decay b_i of a ray along the line ℓ_i through Ω is governed by a line integral,

$$b_i = \log(I_0/I_i) = \int_{\ell_i} \mathcal{X}(t)\, d\ell = b_i, \tag{C.36}$$

where I_0 and I_i are the intensities of the ray before and after passing through the object. When a large number of these line integrals are recorded, then we are able to reconstruct an approximation of the function $\mathcal{X}(t)$.

We discretize the problem as described in Section 2, such that $\mathcal{X}$ is approximated by a piecewise constant function in each voxel in the domain $\Omega = [0,1] \times [0,1] \times [0,1]$. Then the line integral along ℓ_i is computed by summing the contributions from all the voxels penetrated by ℓ_i. If the path length of the ith ray through the jth voxel is denoted by a_{ij}, then we obtain the linear equations

$$\sum_{j=1}^{N} a_{ij} x_j = b_i, \qquad i = 1, \ldots, M, \tag{C.37}$$

where M is the number of rays or measurements and N is the number of voxels. This is a linear system of equations $A\,x = b$ with a sparse coefficient matrix $A \in \mathbb{R}^{M \times N}$.

A widely used test image in medical tomography is the "Shepp-Logan phantom," which consists of a number superimposed ellipses. In the MATLAB function `shepplogan3d` [40] this 2D image is generalized to 3D by superimposing ellipsoids instead. The voxels are in the range $[0, 1]$, and Fig. C.2 shows an example with $43 \times 43 \times 43$ voxels.

We construct the matrix A for a parallel-beam geometry with orthogonal projections of the object along directions well distributed over the unit sphere.

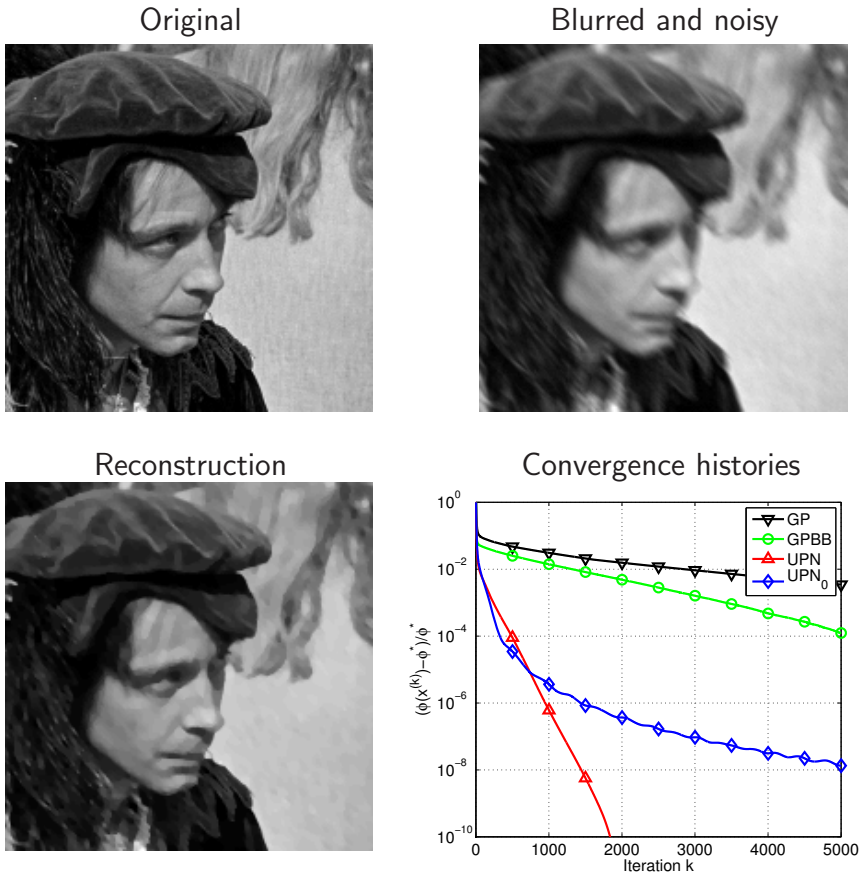


Fig. C.1: Example of total variation deblurring for motion blur with reflexive boundary conditions. Methods are Gradient Projection (*GP*), Gradient Projection Barzilai-Borwein (*GPBB*), Unknown Parameter Nesterov (*UPN*), and *UPN* with $\mu_k = 0$ (UPN_0). Both *UPN* and UPN_0 are much faster than *GP* and *GPBB*, and for a high-accuracy solution *UPN* also outperforms UPN_0.

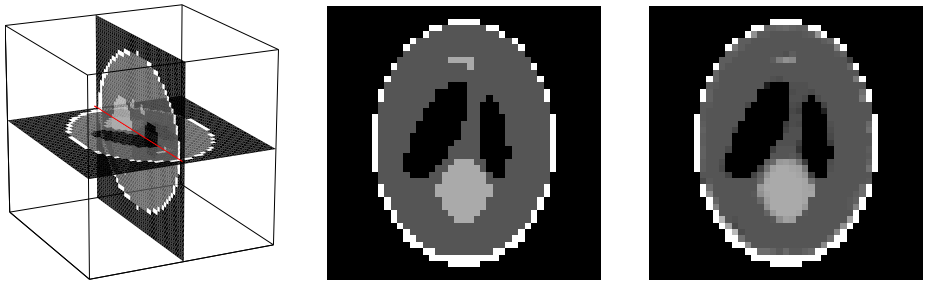


Fig. C.2: Left: Two orthogonal slices through the 3D Shepp-Logan phantom discretized on a 43^3 grid used in our test problems. Middle: Central horizontal slice. Right: Example of solution for $\alpha = 1$ and $\tau = 10^{-4}$. A less smooth solution can be obtained using a smaller α. Original voxel/pixel values are 0.0, 0.2, 0.3 and 1.0. Color range in display is set to $[0.1, 0.4]$ for better constrast.

The projection directions are the direction vectors of so-called *Lebedev quadrature* points on the unit sphere, and the directions are evenly distributed over the sphere; we use the MATLAB implementation `getLebedevSphere` [41]. For setting up the tomography system matrix for a parallel beam geometry, we use the Matlab implementation `tomobox` [42].

This section describes our numerical experiments with the four methods *UPN*, *UPN*$_0$, *GP* and *GPBB* applied to the TV regularization problem (C.7). We use the two test problems listed in Table C.1, which are representative across a larger class of problems (other directions, number of projections, noise levels, etc.) that we have run simulations with. The smallest eigenvalue of A^TA for **T1** is $2.19 \cdot 10^{-5}$ (as computed by MATLAB's `eigs`), confirming that $\text{rank}(A) = N$ for **T1**. We emphasize that this computation is only conducted to support the analysis of the considered problems since – as we have argued in the introduction – it carries a considerable computational burden to compute. In all simulations we create noisy data from an exact object x_{exact} through the forward mapping $b = Ax_{\text{exact}} + e$, subject to additive Gaussian white noise of relative noise level $\|e\|_2/\|b\|_2 = 0.01$. As initial point of the optimization algorithm we use the fifth iteration of the conjugate gradient method applied to the least squares problem.

Table C.1: Specifications of the two test problems; the object domain consists of $m \times n \times l$ voxels and each projection is a $p \times p$ image. Any zero rows have been purged from A.

Problem	$m = n = l$	p	projections	dimensions of A	rank
T1	43	63	37	99361×79507	$= 79507$
T2	43	63	13	33937×79507	< 79507

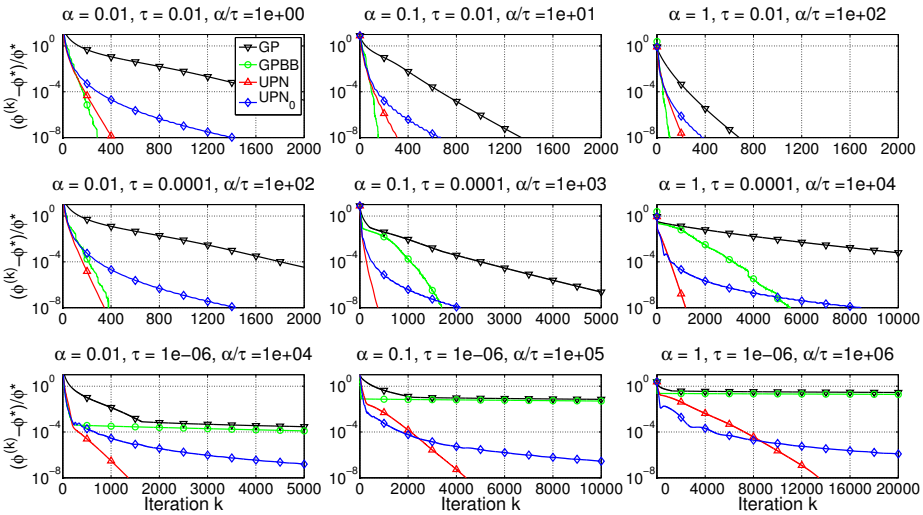


Fig. C.3: Convergence histories $(\phi(x^{(k)}) - \phi^\star)/\phi^\star$ vs. k for **T1** with $\alpha = 0.01$, 0.1 and 1 and $\tau = 10^{-2}$, 10^{-4} and 10^{-6}. Methods are Gradient Projection (*GP*), Gradient Projection Barzilai-Borwein (*GPBB*), Unknown Parameter Nesterov (*UPN*), and *UPN* with $\mu_k = 0$ (*UPN*$_0$). As the ratio α/τ increases, which implies an increased Q and a computationally more difficult problem, *UPN* and *UPN*$_0$ scale significantly better. For high accuracy solutions *UPN* is always competetive.

We compare the algorithm *UPN* with *GP* (the gradient projection method (C.14) with backtracking line search on the step size), *GPBB* and *UPN*$_0$. The latter is *UPN* with $\mu_i = 0$ for all $i = 0, \cdots, k$ and $\theta_1 = 1$ and is optimal for the class $\mathcal{F}_{0,L}$.

7.3 Influence of α and τ on the convergence

For a given A the theoretical modulus of strong convexity given in (C.13) varies only with α and τ. We therefore expect better convergence rates (C.16) and (C.18) for smaller α and larger τ. In Fig. C.3 we show the convergence histories for **T1** with all combinations of $\alpha = 0.01$, 0.1, 1 and $\tau = 10^{-2}$, 10^{-4}, 10^{-6}.

For low α/τ ratios, i.e., small condition number of the Hessian, *GPBB* and *GP* requires a comparable or smaller number of iterations than *UPN* and *UPN*$_0$. As α/τ increases, both *GPBB* and *GP* exhibit slower convergence, while *UPN* is less affected. In all cases *UPN* shows linear convergence, at least in the final stage, while *UPN*$_0$ shows sublinear convergence. Due to these observations, we consistently observe that for sufficiently high accuracy, *UPN* requires the lowest number of iterations. This also follows from the theory since *UPN* scales as

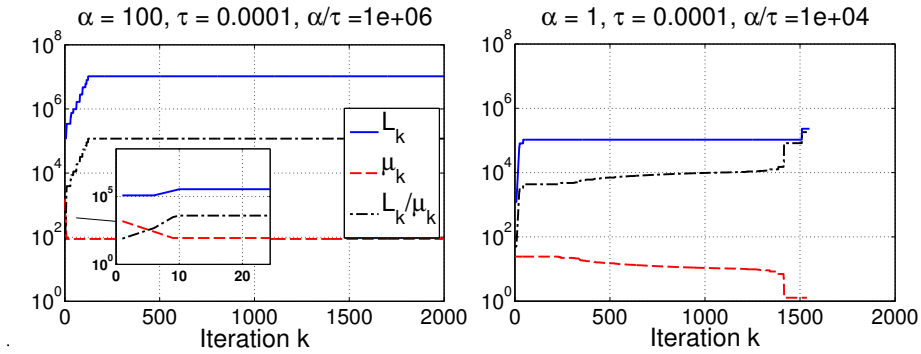


Fig. C.4: The μ_k, L_k histories for **T1**. Left: $\alpha = 100$ and $\tau = 10^{-4}$. Right: $\alpha = 1$ and $\tau = 10^{-4}$.

$\mathcal{O}(\log \epsilon^{-1})$, whereas UPN_0 scales at a higher complexity of $\mathcal{O}(\sqrt{\epsilon^{-1}})$.

We conclude that for small condition numbers there is no gain in using *UPN* compared to *GPBB*. For larger condition numbers, and in particular if a high-accuracy solution is required, *UPN* converges significantly faster. Assume that we were to choose only one of the four algorithms to use for reconstruction across the condition number range. When *UPN* requires the lowest number of iterations, it requires *significantly* fewer, and when not, *UPN* only requires slightly more iterations than the best of the other algorithms. Therefore, *UPN* appears to be the best choice. Obviously, the choice of algorithm also depends on the demanded accuracy of the solution. If only a low accuracy, say $(\phi^{(k)} - \phi^\star)/\phi^\star = 10^{-2}$ is sufficient, all four methods perform more or less equally well.

7.4 Restarts and μ_k and L_k histories

To ensure convergence of *UPN* we introduced the restart functionality *RUPN*. In practice, we almost never observe a restart, e.g., in none of the experiments reported so far a restart occurred. An example where restarts do occur is obtained if we increase α to 100 for **T1** (still $\tau = 10^{-4}$). Restarts occur in the first 8 iterations, and each time μ_k is reduced by a constant factor of $\rho_\mu = 0.7$. In Fig. C.4, left, the μ_k and L_k histories are plotted vs. k and the restarts are seen in the zoomed inset as the rapid, constant decrease in μ_k. From the plot we also note that after the decrease in μ_k and an initial increase in L_k, both estimates are constant for the remaining iterations, indicating that the heuristics determine sufficient values.

For comparison the μ_k and L_k histories for **T1** with $\alpha = 1$ and $\tau = 10^{-4}$ are seen in Fig. C.4, right. No restarts occurred here, and μ_k decays gradually, except for one final jump, while L_k remains almost constant.

7.5 A non-strongly convex example

Test problem **T2** corresponds to only 13 projections, which causes A to not have full column rank. This leads to $\lambda_{\min}(A^TA) = 0$, and hence $\phi(x)$ is not strongly convex. The optimal convergence rate is therefore given by (C.17); but how does the lack of strong convexity affect *UPN*, which was specifically constructed for strongly convex problems? *UPN* does not recognize that the problem is not strongly convex but simply relies on the heuristic (C.29) at the kth iteration. We investigate the convergence by solving **T2** with $\alpha = 1$ and $\tau = 10^{-4}$. Convergence histories are given in Fig. C.5, left. The algorithm *UPN* still converges linearly, although slightly slower than in the **T1** experiment ($\alpha = 1, \tau = 10^{-4}$) in Fig. C.3. The algorithms *GP* and *GPBB* converge much more slowly, while at low accuracies UPN_0 is comparable to *UPN*. But the linear convergence makes *UPN* converge faster for high accuracy solutions.

7.6 Influence of the heuristic

An obvious question is how the use of the heuristic for estimating μ affects *UPN* compared to *Nesterov*, where μ (and L) are assumed known. From Theorem 3.7 we can compute a strong convexity parameter and a Lipschitz parameter for $\phi(x)$ assuming we know the largest and smallest magnitude eigenvalues of A^TA. Recall that these μ and L are not necessarily the tightest possible, according to Remark 3.8. For **T1** we have computed $\lambda_{\max}(A^TA) = 1.52 \cdot 10^3$ and $\lambda_{\min}(A^TA) = 2.19 \cdot 10^{-5}$ (by means of `eigs` in MATLAB). Using $\alpha = 1$, $\tau = 10^{-4}$ and $\|D\|_2^2 \leq 12$ from Lemma 3.6 we fix

$$\mu_k = \lambda_{\min}(A^TA) = 2.19 \cdot 10^{-5}, \qquad L_k = \lambda_{\max}(A^TA) + 12\frac{\alpha}{\tau} = 1.22 \cdot 10^5,$$

for all k, and solve test problem **T1** using *UPN* with the heuristics switched off in favor of these *true* strong convexity and Lipschitz parameters. Convergence histories are plotted in Fig. C.5, right.

The convergence is much slower than using *UPN* with the heuristics switched on. We ascribe this behavior to the very large modulus of strong convexity that arise from the true μ and L. It appears that *UPN* works better than the actual degree of strong convexity as measured by μ, by heuristically choosing in each step a μ_k that is sufficient *locally* instead of being restricted to using a *globally* valid μ.

Another question is how much is actually gained in using the heuristic for μ in *UPN* compared to simply using a fixed "guess" throughout the iterations. To answer that question we investigate the number iterations required to obtain $\bar{\epsilon} = 10^{-4}$, 10^{-6} and 10^{-8} solutions for **T1** and **T2** using only the backtracking procedure on L and simply a fixed value $\mu_k \in [10^{-4}, 10^4]$ for all iterations k, see Fig. C.6.

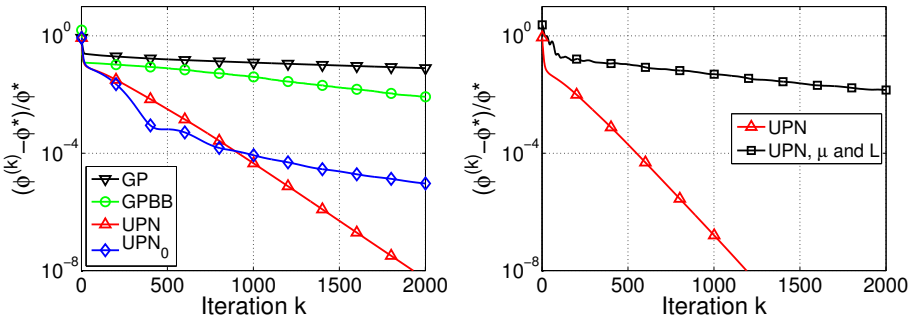


Fig. C.5: Left: Convergence histories of *GP*, *GPBB*, *UPN* and UPN_0 on **T2** with $\alpha = 1$ and $\tau = 10^{-4}$. Right: Convergence histories of *UPN* and *UPN* using true μ and L on **T1** with $\alpha = 1$ and $\tau = 10^{-4}$.

The choice of fixed μ_k has a large impact on the required number of iterations, and there is a distinct optimal choice between 1 and 10. Choosing a fixed μ_k away from the optimal one leads to more iterations and the number of additional iterations grows faster for more accurate solutions. For comparison the figure also shows the corresponding number of iterations required by *UPN* plotted as function of the final *UPN*-estimate for μ. For all three **T1** cases *UPN* comes very close to the optimal number of iterations, without demanding an accurate guess of μ by the user. For **T2** we observe similar trends, although *UPN* requires slightly more iterations than with the optimal choice of fixed μ_k.

We conclude that there exists a choice of fixed μ_k that gives good performance; however, for an inaccurate guess of this value, the number of iterations will be much higher, in particular if an accurate solution is required. *UPN* avoids the need for such a guess and provides the solution using a near-optimal number of iterations. We emphasize that obtaining a *true* strong convexity parameter μ is not of particular interest here, nor is the final *UPN*-estimate for μ, as the goal is simply to obtain fast convergence.

8 Conclusion

We presented an implementation of an optimal first-order optimization algorithm for large-scale problems, suited for functions that are smooth and strongly convex. While the underlying algorithm by Nesterov depends on knowledge of two parameters that characterize the smoothness and strong convexity, we have implemented methods that estimate these parameters during the iterations, thus making the algorithm of practical use.

We tested the performance of the algorithm and compared it with two vari-

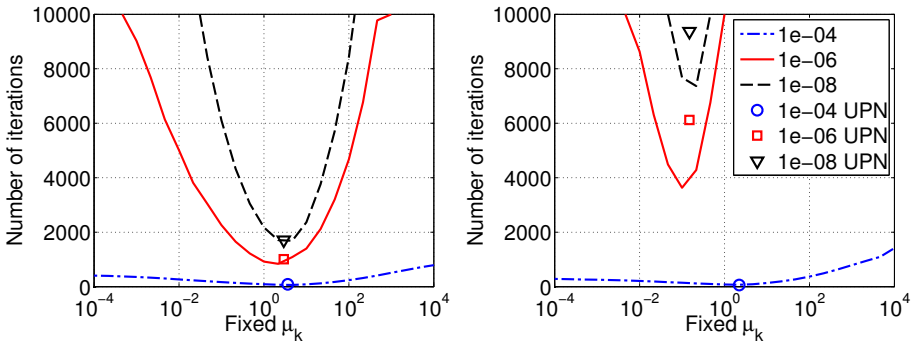


Fig. C.6: Number of iterations needed to obtain TV-solutions ($\alpha = 0.01$) to tolerances $\bar{\epsilon} = 10^{-4}$, 10^{-6} and 10^{-8} using fixed μ_k, left **T1**, right **T2**. Also shown are the number of iterations needed by *UPN* as function of the final estimate of μ. Choices of μ_k not equal to the unknown optimal value lead to many more iterations. *UPN* needs a near-optimal number of iterations without requiring the user to choose a value for μ.

ants of the gradient projection algorithm and a variant of an optimal/accelerated algorithm. We applied the algorithms to total variation regularized tomographic reconstruction of a generic threedimensional test problem. The tests show that, with regards to the number of iterations, the proposed algorithm is competitive with other first-order algorithms, and superior for difficult problems, i.e., ill-conditioned problems solved to high accuracy. Simulations also show that even for problems that are not strongly convex, in practice we achieve the favorable convergence rate associated with strong convexity. The software is available as a C-implementation with an interface to MATLAB from `www.imm.dtu.dk/~pch/TVReg/`.

Acknowledgements

We wish to thank both referees for their constructive comments which helped improve the presentation of the material.

Appendix A: The Optimal Convergence Rate

Here we provide an analysis of an optimal method for smooth, strongly convex functions without the use of estimation functions as in [13]. This approach is similar to the analysis of optimal methods for smooth functions in [25, 29]. The motivation for the following derivations is to introduce the iteration dependent L_k and μ_k estimates of L and μ. This will support the analysis of how L_k and

μ_k should be selected. We start with the following relations to the "hidden" supporting variables $z^{(k)}$ and γ_k [13, pp. 73–75, 89],

$$y^{(k)} - x^{(k)} = \frac{\theta_k \gamma_k}{\gamma_{k+1}}(z^{(k)} - y^{(k)}), \tag{C.38}$$

$$\begin{aligned} \gamma_{k+1} &= (1-\theta_k)\gamma_k + \theta_k \mu_k = \theta_k^2 L_k \\ \gamma_{k+1} z^{(k+1)} &= (1-\theta_k)\gamma_k z^{(k)} + \theta_k \mu_k y^{(k)} - \theta_k G_{L_k}(y^{(k)}). \end{aligned} \tag{C.39}$$

In addition we will make use of the relations

$$\begin{aligned} \frac{\gamma_{k+1}}{2}\|z^{(k+1)} - y^{(k)}\|_2^2 &= \frac{1}{2\gamma_{k+1}}\Big((1-\theta_k)^2\gamma_k^2\|z^{(k)} - y^{(k)}\|_2^2 \\ &\quad -2\theta_k(1-\theta_k)\gamma_k G_{L_k}(y^{(k)})^T(z^{(k)} - y^{(k)}) \\ &\quad +\theta_k^2\|G_{L_k}(y^{(k)}))\|_2^2\Big), \end{aligned} \tag{C.40}$$

$$(1-\theta_k)\frac{\gamma_k}{2} - \frac{1}{2\gamma_{k+1}}(1-\theta_k)^2\gamma_k^2 = \frac{(1-\theta_k)\gamma_k\theta_k\mu_k}{2\gamma_{k+1}}. \tag{C.41}$$

which originate from (C.39). We will also later need the relation

$$\begin{aligned} &(1-\theta_k)\frac{\gamma_k}{2}\|z^{(k)} - y^{(k)}\|_2^2 - \frac{\gamma_{k+1}}{2}\|z^{(k+1)} - y^{(k)}\|_2^2 + \theta_k G_{L_k}(y^{(k)})^T(y^{(k)} - x^\star) \\ &= (1-\theta_k)\frac{\gamma_k}{2}\|z^{(k)} - y^{(k)}\|_2^2 - \frac{\gamma_{k+1}}{2}\|z^{(k+1)} - y^{(k)}\|_2^2 \\ &\quad + \left(-\gamma_{k+1}z^{(k+1)} + (1-\theta_k)\gamma_k z^{(k)} + \theta_k\mu_k y^{(k)}\right)^T (y^{(k)} - x^\star) \\ &= \left((1-\theta_k)\frac{\gamma_k}{2} - \frac{\gamma_{k+1}}{2} + \theta_k\mu_k\right)(y^{(k)})^T y^{(k)} + (1-\theta_k)\frac{\gamma_k}{2}(z^{(k)})^T z^{(k)} \\ &\quad -\frac{\gamma_{k+1}}{2}(z^{(k+1)})^T z^{(k+1)} + \gamma_{k+1}(z^{(k+1)})^T x^\star - (1-\theta_k)\gamma_k(z^{(k)})^T x^\star \\ &\quad -\theta_k\mu_k(y^{(k)})^T x^\star \\ &= (1-\theta_k)\frac{\gamma_k}{2}\left(\|z^{(k)} - x^\star\|_2^2 - (x^\star)^T x^\star\right) - \frac{\gamma_{k+1}}{2}\left(\|z^{(k+1)} - x^\star\|_2^2 - (x^\star)^T x^\star\right) \\ &\quad + \frac{\theta_k\mu_k}{2}\left(\|y^{(k)} - x^\star\|_2^2 - (x^\star)^T x^\star\right) \\ &\quad \left(+(1-\theta_k)\frac{\gamma_k}{2} - \frac{\gamma_{k+1}}{2} + \frac{\theta_k\mu_k}{2}\right)(y^{(k)})^T y^{(k)} \\ &= (1+\theta_k)\frac{\gamma_k}{2}\|z^{(k)} - x^\star\|_2^2 - \frac{\gamma_{k+1}}{2}\|z^{(k)} - x^\star\|_2^2 + \theta_k\frac{\mu_k}{2}\|y^{(k)} - x^\star\|_2^2, \end{aligned} \tag{C.42}$$

where we again used (C.39). We can now start the analysis of the algorithm by considering the inequality in Lemma 5.1,

$$\begin{aligned}(1-\theta_k)f(x^{(k+1)}) \quad &\leq \quad (1-\theta_k)f(x^{(k)}) + (1-\theta_k)G_{L_k}(y^{(k)})^T(y^{(k)}-x^{(k)}) \\ &\quad -(1-\theta_k)\frac{1}{2L_k}\|G_{L_k}(y^{(k)})\|_2^2, \end{aligned} \tag{C.43}$$

where we have omitted the strong convexity part, and the inequality

$$\begin{aligned}\theta_k f(x^{(k+1)}) \quad &\leq \quad \theta_k f(x^\star) + \theta_k G_{L_k}(y^{(k)})^T(y^{(k)}-x^\star) \\ &\quad -\theta_k\frac{1}{2L_k}\|G_{L_k}(y^{(k)})\|_2^2 - \theta_k\frac{\mu_k^\star}{2}\|y^{(k)}-x^\star\|_2^2. \end{aligned} \tag{C.44}$$

Adding these bounds and continuing, we obtain

$$\begin{aligned}f(x^{(k+1)}) \;&\leq\; (1-\theta_k)f(x^{(k)}) + (1-\theta_k)G_{L_k}(y^{(k)})^T(y^{(k)}-x^{(k)}) \\ &\quad + \theta_k f^\star + \theta_k G_{L_k}(y^{(k)})^T(y^{(k)}-x^\star) \\ &\quad -\theta_k\frac{\mu_k^\star}{2}\|x^\star-y^{(k)}\|_2^2 - \frac{1}{2L_k}\|G_{L_k}(y^{(k)})\|_2^2 \\ &=\; (1-\theta_k)f(x^{(k)}) + (1-\theta_k)\frac{\theta_k\gamma_k}{\gamma_{k+1}}G_{L_k}(y^{(k)})^T(z^{(k)}-y^{(k)}) \\ &\quad + \theta_k f^\star + \theta_k G_{L_k}(y^{(k)})^T(y^{(k)}-x^\star) \\ &\quad -\theta_k\frac{\mu_k^\star}{2}\|x^\star-y^{(k)}\|_2^2 - \frac{1}{2L_k}\|G_{L_k}(y^{(k)})\|_2^2 \\ &\leq\; (1-\theta_k)f(x^{(k)}) + (1-\theta_k)\frac{\theta_k\gamma_k}{\gamma_{k+1}}G_{L_k}(y^{(k)})^T(z^{(k)}-y^{(k)}) \\ &\quad + \theta_k f^\star + \theta_k G_{L_k}(y^{(k)})^T(y^{(k)}-x^\star) - \theta_k\frac{\mu_k^\star}{2}\|x^\star-y^{(k)}\|_2^2 \\ &\quad -\frac{1}{2L_k}\|G_{L_k}(y^{(k)})\|_2^2 + \frac{(1-\theta_k)\theta_k\gamma_k\mu_k}{2\gamma_{k+1}}\|z^{(k)}-y^{(k)}\|_2^2 \\ &=\; (1-\theta_k)f(x^{(k)}) + (1-\theta_k)\frac{\theta_k\gamma_k}{\gamma_{k+1}}G_{L_k}(y^{(k)})^T(z^{(k)}-y^{(k)}) \\ &\quad + \theta_k f^\star + \theta_k G_{L_k}(y^{(k)})^T(y^{(k)}-x^\star) - \theta_k\frac{\mu_k^\star}{2}\|x^\star-y^{(k)}\|_2^2 \\ &\quad -\frac{1}{2L_k}\|G_{L_k}(y^{(k)})\|_2^2 \\ &\quad + \left((1-\theta_k)\frac{\gamma_k}{2} - \frac{1}{2\gamma_{k+1}}(1-\theta_k)^2\gamma_k^2\right)\|z^{(k)}-y^{(k)}\|_2^2 \end{aligned}$$

$$
\begin{aligned}
&= (1-\theta_k)f(x^{(k)}) + (1-\theta_k)\frac{\gamma_k}{2}\|z^{(k)}-y^{(k)}\|_2^2 - \frac{\gamma_{k+1}}{2}\|z^{(k+1)}-y^{(k)}\|_2^2 \\
&\quad + \theta_k f^\star + \theta_k G_{L_k}(y^{(k)})^T(y^{(k)}-x^\star) - \theta_k\frac{\mu_k^\star}{2}\|x^\star-y^{(k)}\|_2^2 \\
&= (1-\theta_k)f(x^{(k)}) + \theta_k f^\star - \theta_k\frac{\mu_k^\star}{2}\|x^\star-y^{(k)}\|_2^2 \\
&\quad + (1-\theta_k)\frac{\gamma_k}{2}\|z^{(k)}-x^\star\|_2^2 - \frac{\gamma_{k+1}}{2}\|z^{(k+1)}-x^\star\|_2^2 \\
&\quad + \theta_k\frac{\mu_k}{2}\|y^{(k)}-x^\star\|_2^2,
\end{aligned}
$$

where we have used (C.38), a trivial inequality, (C.41), (C.40), (C.39), and (C.42). If $\mu_k \leq \mu_k^\star$ then

$$
f(x^{(k+1)}) - f^\star + \frac{\gamma_{k+1}}{2}\|z^{(k+1)}-x^\star\|_2^2 \leq (1-\theta_k)\left(f(x^{(k)}) - f^\star + \frac{\gamma_k}{2}\|z^{(k)}-x^\star\|_2^2\right) \tag{C.45}
$$

in which case we can combine the bounds to obtain

$$
f(x^{(k)}) - f^\star + \frac{\gamma_k}{2}\|z^{(k)}-x^\star\|_2^2 \leq \left(\prod_{i=0}^{k-1}(1-\theta_i)\right)\left(f(x^{(0)}) - f^\star + \frac{\gamma_k}{2}\|z^{(0)}-x^\star\|_2^2\right), \tag{C.46}
$$

where we have also used $x^{(0)} = y^{(0)}$ and (C.38) to obtain $x^{(0)} = z^{(0)}$. For completeness, we will show why this is an optimal first-order method. Let $\mu_k = \mu_k^\star = \mu$ and $L_k = L$. If $\gamma_0 \geq \mu$ then using (C.39) we obtain $\gamma_{k+1} \geq \mu$ and $\theta_k \geq \sqrt{\mu/L} = \sqrt{Q^{-1}}$. Simultaneously, we also have $\prod_{i=0}^{k-1}(1-\theta_k) \leq \frac{4L}{\left(2\sqrt{L}+k\sqrt{\gamma_0}\right)^2}$ [13, Lemma 2.2.4], and the bound is then

$$
\begin{aligned}
f(x^{(k)}) - f^\star \quad \leq \quad & \min\left(\left(1-\sqrt{Q^{-1}}\right)^k, \frac{4L}{\left(2\sqrt{L}+k\sqrt{\gamma_0}\right)^2}\right) \\
& \cdot \left(f(x^{(0)}) - f^\star + \frac{\gamma_0}{2}\|x^{(0)}-x^\star\|_2^2\right).
\end{aligned} \tag{C.47}
$$

This is the optimal convergence rate for the class $\mathcal{F}_{0,L}$ and $\mathcal{F}_{\mu,L}$ simultaneously [13, 28].

Appendix B: Complexity Analysis

In this Appendix we prove Theorem 6.1, *i.e.*, we derive the complexity for reaching an ϵ-suboptimal solution for the algorithm *UPN*. The total worst-case complexity is given by a) the complexity for the worst case number of restarts and b) the worst-case complexity for a successful termination.

With a slight abuse of notation in this Appendix, $\mu_{k,r}$ denotes the kth iterate in the rth restart stage, and similarly for $L_{k,r}$, $\tilde{L}_{k,r}$, $x^{(k,r)}$, etc. The value $\mu_{0,0}$ is the initial estimate of the strong convexity parameter when no restart has occurred. In the worst case, the heuristic choice in (C.29) never reduces μ_k, such that we have $\mu_{k,r} = \mu_{0,r}$. Then a total of R restarts are required, where

$$\rho_\mu^R \, \mu_{0,0} = \mu_{0,R} \leq \mu \quad \Longleftrightarrow \quad R \geq \log(\mu_{0,0}/\mu)/\log(1/\rho_\mu).$$

In the following analysis we shall make use of the relation

$$\exp\left(-\frac{n}{\delta^{-1}-1}\right) \leq (1-\delta)^n \leq \exp\left(-\frac{n}{\delta^{-1}}\right), \qquad 0<\delta<1, \quad n \geq 0\,.$$

Appendix B.1: Termination Complexity

After sufficiently many restarts (at most R), $\mu_{0,r}$ will be sufficiently small in which case (C.34) holds and we obtain

$$\begin{aligned}
\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2^2 \;&\leq\; \prod_{i=1}^{k}\left(1-\sqrt{\frac{\mu_{i,r}}{L_{i,r}}}\right)\left(\frac{4\tilde{L}_{k+1,r}}{\mu_{k,r}} - \frac{2\tilde{L}_{k+1,r}}{2L_{0,r}} + \frac{2\tilde{L}_{k+1,r}\gamma_{1,r}}{\mu_{k,r}^2}\right) \\
&\qquad \cdot\|G_{L_0}(x^{(0,r)})\|_2^2 \\
&\leq\; \left(1-\sqrt{\frac{\mu_{k,r}}{L_{k,r}}}\right)^k\left(\frac{4\tilde{L}_{k+1,r}}{\mu_{k,r}} - \frac{2\tilde{L}_{k+1,r}}{2L_{0,r}} + \frac{2\tilde{L}_{k+1,r}\gamma_{1,r}}{\mu_{k,r}^2}\right) \\
&\qquad \cdot\|G_{L_{0,r}}(x^{(0,r)})\|_2^2 \\
&\leq\; \exp\left(-\frac{k}{\sqrt{L_{k,r}/\mu_{k,r}}}\right)\left(\frac{4\tilde{L}_{k+1,r}}{\mu_{k,r}} - \frac{\tilde{L}_{k+1,r}}{L_{0,r}} + \frac{2\tilde{L}_{k+1,r}\gamma_{1,r}}{\mu_{k,r}^2}\right) \\
&\qquad \cdot\|G_{L_{0,r}}(x^{(0,r)})\|_2^2,
\end{aligned}$$

where we have used $L_{i,r} \leq L_{i+1,r}$ and $\mu_{i,r} \geq \mu_{i+1,r}$. To guarantee $\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2 \leq \bar{\epsilon}$ we require the latter bound to be smaller than $\bar{\epsilon}^2$, i.e.,

$$\begin{aligned}
\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2^2 \;&\leq\; \exp\left(-\frac{k}{\sqrt{L_{k,r}/\mu_{k,r}}}\right)\left(\frac{4\tilde{L}_{k+1,r}}{\mu_{k,r}} - \frac{\tilde{L}_{k+1}}{L_{0,r}} + \frac{2\tilde{L}_{k+1,r}\gamma_{1,r}}{\mu_{k,r}^2}\right) \\
&\qquad \cdot\|G_{L_{0,r}}(x^{(0,r)})\|_2^2 \leq \bar{\epsilon}^2.
\end{aligned}$$

Solving for k, we obtain

$$k = \mathcal{O}\big(\sqrt{Q}\log Q\big) + \mathcal{O}\big(\sqrt{Q}\log\bar{\epsilon}^{-1}\big), \tag{C.48}$$

where we have used $\mathcal{O}\big(\sqrt{L_{k,r}/\mu_{k,r}}\big) = \mathcal{O}\big(\sqrt{\tilde{L}_{k+1,r}/\mu_{k,r}}\big) = \mathcal{O}\big(\sqrt{Q}\big)$.

Appendix B.2: Restart Complexity

How many iterations are needed before we can detect that a restart is needed? The restart detection rule (C.34) gives

$$\begin{aligned}\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2^2 &> \prod_{i=1}^{k}\left(1-\sqrt{\frac{\mu_{i,r}}{L_{i,r}}}\right)\left(\frac{4\tilde{L}_{k+1,r}}{\mu_{k,r}}-\frac{2\tilde{L}_{k+1,r}}{2L_{0,r}}+\frac{2\tilde{L}_{k+1,r}\gamma_{1,r}}{\mu_{k,r}^2}\right)\\ &\quad\cdot\|G_{L_{0,r}}(x^{(0,r)})\|_2^2\\ &\geq \left(1-\sqrt{\frac{\mu_{1,r}}{L_{1,r}}}\right)^{k}\left(\frac{4\tilde{L}_{1,r}}{\mu_{1,r}}-\frac{2\tilde{L}_{1,r}}{2L_{0,r}}+\frac{2\tilde{L}_{1,r}\gamma_{1,r}}{\mu_{1,r}^2}\right)\\ &\quad\cdot\|G_{L_{0,r}}(x^{(0,r)})\|_2^2\\ &\geq \exp\left(-\frac{k}{\sqrt{L_{1,r}/\mu_{1,r}}-1}\right)\left(\frac{4L_{1,r}}{\mu_{1,r}}-\frac{2L_{1,r}}{2L_{0,r}}+\frac{2L_{1,r}\gamma_{1,r}}{\mu_{1,r}^2}\right)\\ &\quad\cdot\|G_{L_{0,r}}(x^{(0,r)})\|_2^2,\end{aligned}$$

where we have used $L_{i,r}\leq L_{i+1,r}$, $L_{i,r}\leq\tilde{L}_{i+1,r}$ and $\mu_{i,r}\geq\mu_{i+1,r}$. Solving for k, we obtain

$$k>\left(\sqrt{\frac{L_{1,r}}{\mu_{1,r}}}-1\right)\left(\log\left(\frac{4L_{1,r}}{\mu_{1,r}}-\frac{L_{1,r}}{L_{0,r}}+\frac{4\gamma_{1,r}L_{1,r}}{\mu_{1,r}^2}\right)+\log\frac{\|G_{L_{0,r}}(x^{(0,r)})\|_2^2}{\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2^2}\right). \tag{C.49}$$

Since we do not terminate but restart, we have $\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2\geq\bar{\epsilon}$. After r restarts, in order to satisfy (C.49) we must have k of the order

$$\mathcal{O}\big(\sqrt{Q_r}\big)\,\mathcal{O}\big(\log Q_r\big)+\mathcal{O}\big(\sqrt{Q_r}\big)\,\mathcal{O}\big(\log\bar{\epsilon}^{-1}\big),$$

where

$$Q_r=\mathcal{O}\left(\frac{L_{1,r}}{\mu_{1,r}}\right)=\mathcal{O}\big(\rho_\mu^{R-r}Q\big)\,.$$

The worst-case number of iterations for running R restarts is then given by

$$\begin{aligned}
&\sum_{r=0}^{R} \mathcal{O}\Big(\sqrt{Q\rho_\mu^{R-r}}\Big)\ \mathcal{O}(\log Q\rho_\mu^{R-r}) + \mathcal{O}\left(\sqrt{Q\rho_\mu^{R-r}}\right)\ \mathcal{O}(\log \bar{\epsilon}^{-1}) \\
&= \sum_{i=0}^{R} \mathcal{O}\Big(\sqrt{Q\rho_\mu^{i}}\Big)\ \mathcal{O}(\log Q\rho_\mu^{i}) + \mathcal{O}\Big(\sqrt{Q\rho_\mu^{i}}\Big)\ \mathcal{O}(\log \bar{\epsilon}^{-1}) \\
&= \mathcal{O}\Big(\sqrt{Q}\Big)\left\{\sum_{i=0}^{R} \mathcal{O}\Big(\sqrt{\rho_\mu^{i}}\Big)\left[\mathcal{O}\big(\log Q\rho_\mu^{i}\big) + \mathcal{O}\big(\log \bar{\epsilon}^{-1}\big)\right]\right\} \\
&= \mathcal{O}\Big(\sqrt{Q}\Big)\left\{\sum_{i=0}^{R} \mathcal{O}\Big(\sqrt{\rho_\mu^{i}}\Big)\Big[\mathcal{O}(\log Q) + \mathcal{O}\big(\log \bar{\epsilon}^{-1}\big)\Big]\right\} \\
&= \mathcal{O}\Big(\sqrt{Q}\Big)\Big\{\mathcal{O}(1)\Big[\mathcal{O}(\log Q) + \mathcal{O}\big(\log \bar{\epsilon}^{-1}\big)\Big]\Big\} \\
&= \mathcal{O}\Big(\sqrt{Q}\Big)\,\mathcal{O}(\log Q) + \mathcal{O}\Big(\sqrt{Q}\Big)\,\mathcal{O}\big(\log \bar{\epsilon}^{-1}\big) \\
&= \mathcal{O}\Big(\sqrt{Q}\log Q\Big) + \mathcal{O}\Big(\sqrt{Q}\log \bar{\epsilon}^{-1}\Big), \qquad \text{(C.50)}
\end{aligned}$$

where we have used

$$\sum_{i=0}^{R} \mathcal{O}\Big(\sqrt{\rho_\mu^{i}}\Big) = \sum_{i=0}^{R} \mathcal{O}\Big(\sqrt{\rho_\mu}^{\,i}\Big) = \mathcal{O}\left(\frac{1-\sqrt{\rho_\mu^{R+1}}}{1-\sqrt{\rho_\mu}}\right) = \mathcal{O}(1).$$

Appendix B.3: Total Complexity

The total iteration complexity of *UPN* is given by (C.50) plus (C.48):

$$\mathcal{O}\big(\sqrt{Q}\log Q\big) + \mathcal{O}\big(\sqrt{Q}\log \bar{\epsilon}^{-1}\big). \qquad \text{(C.51)}$$

It is common to write the iteration complexity in terms of reaching an ϵ-suboptimal solution satisfying $f(x) - f^\star \leq \epsilon$. This is different from the stopping criteria $\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2 \leq \bar{\epsilon}$ or $\|G_{L_{k,r}}(y^{(k,r)})\|_2 \leq \bar{\epsilon}$ used in the *UPN* algorithm. Consequently, we will derive a relation between ϵ and $\bar{\epsilon}$. Using Lemmas 5.1 and 5.2, in case we stop using $\|G_{L_{k,r}}(y^{(k,r)})\|_2 \leq \bar{\epsilon}$ we obtain

$$f\big(x^{(k+1,r)}\big) - f^\star \leq \left(\frac{2}{\mu} - \frac{1}{2L_{k,r}}\right)\|G_{L_{k,r}}(y^{(k,r)})\|_2^2 \leq \frac{2}{\mu}\|G_{L_{k,r}}(y^{(k,r)})\|_2^2 \leq \frac{2}{\mu}\bar{\epsilon}^2,$$

and in case we stop using $\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2 \leq \bar{\epsilon}$, we obtain

$$\begin{aligned} f\big(\tilde{x}^{(k+1,r)}\big) - f^\star &\leq \left(\frac{2}{\mu} - \frac{1}{2\tilde{L}_{k+1,r}}\right) \|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2^2 \\ &\leq \frac{2}{\mu}\|G_{\tilde{L}_{k+1,r}}(x^{(k+1,r)})\|_2^2 \leq \frac{2}{\mu}\bar{\epsilon}^2. \end{aligned}$$

To return with either $f(\tilde{x}^{(k+1,r)}) - f^\star \leq \epsilon$ or $f(x^{(k+1,r)}) - f^\star \leq \epsilon$ we require the latter bounds to hold and thus select $(2/\mu)\,\bar{\epsilon}^2 = \epsilon$. The iteration complexity of the algorithm in terms of ϵ is then

$$\begin{aligned} \mathcal{O}\Big(\sqrt{Q}\log Q\Big) + \mathcal{O}\Big(\sqrt{Q}\log\big((\mu\epsilon)^{-1}\big)\Big) &= \mathcal{O}\Big(\sqrt{Q}\log Q\Big) + \mathcal{O}\Big(\sqrt{Q}\log\mu^{-1}\Big) \\ &\quad + \mathcal{O}\Big(\sqrt{Q}\log\epsilon^{-1}\Big) \\ &= \mathcal{O}\Big(\sqrt{Q}\log Q\Big) + \mathcal{O}\Big(\sqrt{Q}\log\epsilon^{-1}\Big), \end{aligned}$$

where we have used $\mathcal{O}(1/\mu) = \mathcal{O}(L/\mu) = \mathcal{O}(Q)$.

References

[1] P. C. Hansen, *Discrete Inverse Problems: Insight and Algorithms.* SIAM, Philadephia, 2010.

[2] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods.* SIAM, Philadelphia, 2005.

[3] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.

[4] C. R. Vogel and M. E. Oman, "Iterative methods for total variation denoising," *SIAM J. Sci. Comput.*, vol. 17, pp. 227–238, 1996.

[5] F. Alter, S. Durand, and J. Froment, "Adapted total variation for artifact free decompression of JPEG images," *J. Math. Imaging Vis.*, vol. 23, pp. 199–211, 2005.

[6] P. L. Combettes and J. Luo, "An adaptive level set method for nondifferentiable constrained image recovery," *IEEE Trans. Image Proces.*, vol. 11, pp. 1295–1304, 2002.

[7] D. Goldfarb and W. Yin, "Second-order cone programming methods for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 27, pp. 622–645, 2005.

[8] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imaging Vis.*, vol. 20, pp. 89–97, 2004.

[9] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 20, pp. 1964–1977, 1998.

[10] M. Hintermüller and G. Stadler, "An infeasible primal-dual algorithm for total bounded variation-based INF-convolution-type image restoration," *SIAM J. Sci. Comput.*, vol. 28, pp. 1–23, 2006.

[11] A. Chambolle, "Total variation minimization and a class of binary MRF models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, ser. Lecture Notes in Computer Science, A. Rangarajan, B. Vemuri, and A. L. Yuille, Eds., vol. 3757. Springer-Verlag, Berlin, 2005, pp. 136–152.

[12] J. Darbon and M. Sigelle, "Image restoration with discrete constrained total variation – Part I: Fast and exact optimization," *J. Math. Imaging Vis.*, vol. 26, pp. 261–276, 2006.

[13] Y. Nesterov, *Introductory Lectures on Convex Optimization.* Kluwer Academic Publishers, Dordrecht, 2004.

[14] ——, "Smooth minimization of nonsmooth functions," *Math. Prog. Series A*, vol. 103, pp. 127–152, 2005.

[15] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algo.*, vol. 53, pp. 67–92, 2010.

[16] P. Weiss, L. Blanc-Féraud, and G. Aubert, "Efficient schemes for total variation minimization under constraints in image processing," *SIAM J. Sci. Comput.*, vol. 31, pp. 2047–2080, 2009.

[17] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vis.*, vol. 40, pp. 120–145, 2011.

[18] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, 2009.

[19] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA J. Numer. Anal.*, vol. 8, pp. 141–148, 1988.

[20] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for Newton's method," *SIAM J. Numer. Anal.*, vol. 23, pp. 707–716, 1986.

[21] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, 2007.

[22] G. Lan, Z. Lu, and R. D. C. Monteiro, "Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming," *Math. Prog. Ser. A*, vol. 126, no. 1, pp. 1–29, 2011.

[23] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$," *Doklady AN SSSR (translated as Soviet Math. Docl.)*, vol. 269, pp. 543–547, 1983.

[24] ——, "Gradient methods for minimizing composite objective function," 2007, cORE Discussion Paper No 2007076, www.ecore.be/DPs/dp_1191313936.pdf.

[25] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," 2008, manuscript. www.math.washington.edu/~tseng/papers/apgm.pdf.

[26] S. Becker, E. J. Candès, and M. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, no. 3, pp. 165–218, 2011.

[27] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imag. Sci.*, vol. 4, no. 1, pp. 1–39, 2011.

[28] A. S. Nemirovsky and D. B. Yudin, *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, New York, 1983.

[29] L. Vandenberghe, "Optimization methods for large-scale systems," 2009, lecture Notes. www.ee.ucla.edu/~vandenbe/ee236c.html.

[30] M. Raydan, "The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem," *SIAM J. Optim.*, vol. 7, pp. 26–33, 1997.

[31] Y.-H. Dai and L.-Z. Liao, "R-linear convergence of the Barzilai and Borwein gradient method," *IMA J. Numer. Anal.*, vol. 22, pp. 1–10, 2002.

[32] R. Fletcher, "Low storage methods for unconstrained optimization," in *Computational Solution of Nonlinear Systems of Equations*, E. L. Ellgower and K. Georg, Eds. Amer. Math. Soc., Providence, 1990, pp. 165–179.

[33] E. G. Birgin, J. M. Martínez, and M. Raydan, "Nonmonotone spectral projected gradient methods on convex sets," *SIAM J. Optim.*, vol. 10, pp. 1196–1211, 2000.

[34] M. Zhu, S. J. Wright, and T. F. Chan, "Duality-based algorithms for total-variation-regularized image restoration," *Comput. Optim. Appl.*, 2008, dOI: 10.1007/s10589-008-9225-2.

[35] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. on Image Process.*, vol. 18, pp. 2419–2434, 2009.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[37] G. T. Herman, *Fundamentals of Computerized Tomography: Image Reconstruction from Projections, 2. Ed.* Springer, New York, 2009.

[38] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. SIAM, Philadelphia, 2001.

[39] G. Nolet, Ed., *Seismic Tomography with Applications in Global Seismology and Exploration Geophysics*. D. Reidel Publishing Company, Dordrecht, 1987.

[40] M. Schabel, "3D Shepp-Logan phantom," 20 Sep 2006, www.mathworks.com/matlabcentral/fileexchange/9416-3d-shepp-logan-phantom.

[41] R. Parrish, "getLebedevSphere," 26 Mar 2010, www.mathworks.com/matlabcentral/fileexchange/27097-getlebedevsphere.

[42] J. H. Jørgensen, "tomobox," 17 Aug 2010, www.mathworks.com/matlabcentral/fileexchange/28496-tomobox.

Paper D

Multiple Descriptions using Sparse Decompositions

T. L. Jensen, J. Østergaard, J. Dahl and S. H. Jensen

This paper is published in
Proc. of the European Signal Processing Conference (EUSIPCO),
Aalborg, Denmark, pp. 110–114, 2010.

The layout is revised.

Minor spelling, gramma and notation errors have been corrected.

Abstract

In this paper, we consider the design of multiple descriptions (MDs) using sparse decompositions. In a description erasure channel only a subset of the transmitted descriptions is received. The MD problem concerns the design of the descriptions such that they individually approximate the source and furthermore are able to refine each other. In this paper, we form descriptions using convex optimization with l_1-norm minimization and Euclidean distortion constraints on the reconstructions and show that with this method we can obtain non-trivial descriptions. We give an algorithm based on recently developed first-order method to the proposed convex problem such that we can solve large-scale instances for image sequences.

1 Introduction

Sparse decomposition is an important method in modern signal processing and has been applied to different applications such as estimation and coding [1], linear prediction [2] and blind source separation [3]. For estimation and encoding the argument for sparse approaches has been to follow natural statistics, see *e.g.*, [4]. The advent of compressed sensing [5, 6] has further added to the interest in sparse decompositions since the recovery of the latent variables requires a sparse acquisition method.

One method to acquire a sparse decomposition with a dictionary is to solve a convex relaxation of the minimum cardinality problem, that is the l_1-*compression problem*

$$\begin{aligned} \text{min.} \quad & \|z\|_1 \\ \text{s.t.} \quad & \|Dz - y\|_2 \leq \delta \,, \end{aligned} \tag{D.1}$$

where $D \in \mathbb{R}^{M \times N}$ is an overcomplete dictionary, $\delta > 0$ is a selected reconstruction error level (distortion), $(N \geq M)$, $z \in \mathbb{R}^N$ is the latent variable and $y \in \mathbb{R}^M$ is the signal we wish to decompose into a sparse representation. There are several other sparse acquisition methods, including approximations of minimum cardinality and pursuit methods.

In this paper, we apply sparse decomposition to the *multiple-description* (MD) problem [7]. The MD problem is on encoding a source into multiple descriptions and each description is then transmitted over a different channel. Unknown to the encoder, a channel may break which correspond to a description erasure such that only a subset of the transmitted descriptions is received. The problem is to design the descriptions such that the decoded descriptions approximate the source for all possible subsets of descriptions.

An important concept for the MD problem is the trade-off associated with the description design; in order for the descriptions to approximate the source, they should be similar to the source, and consequently the descriptions need to be similar to each other. But, if the descriptions are too similar to each other, it is not possible to obtain any refinement when the individual descriptions are combined.

Let J be the number of channels and let $\mathcal{J}_J = \{1, \ldots, J\}$. Then $\mathcal{I}_J = \{\ell \,|\, \ell \subseteq \mathcal{J}_J, \ell \neq \emptyset\}$ describes the non-trivial subsets of descriptions which can be received. Further, let z_j, $\forall j \in \mathcal{J}_J$, denote the jth description and define $z_\ell = \{z_j \mid j \in \ell\}$, $\forall \ell \in \mathcal{I}_J$. At the decoder, the descriptions z_ℓ, $\ell \in \mathcal{I}_J$, are used to reconstruct an approximation of the source y via the reconstruction functions $g_\ell(z_\ell)$. The approximations satisfy the distortion constraint $d(g_\ell(z_\ell), y) \leq \delta_\ell$, $\forall \ell \in \mathcal{I}_J$, with $d(\cdot, \cdot)$ denoting a distortion measure. An example with $J = 2$ is presented in Fig. D.1.

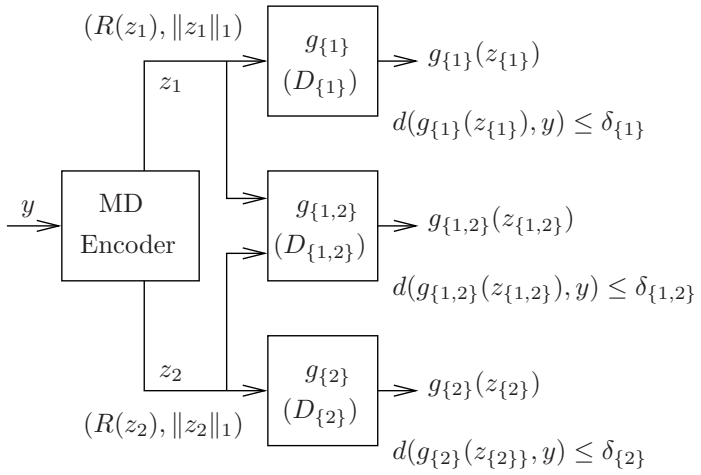


Fig. D.1: The MD (l_1-compression) problem for $J = 2$.

In a statistical setting, the MD problem is to design the descriptions z_j, $\forall j \in \mathcal{J}_J$, such that the total rate $\sum_{j \in \mathcal{J}_J} R(z_j)$ is minimized and the fidelity constraints are satisfied. This problem is only completely solved with the squared error fidelity criterion, memoryless Gaussian sources and two descriptions [8]. Another direction is to form descriptions in a deterministic setting. Algorithms specifically designed for video or image coding may be based on, *e.g.*, Wiener filters with prediction compensation [9], matching pursuit [10, 11] or compressed sensing [12, 13].

The remaining part of the paper is organized as follows: in Sec. 2 we present a method to obtain sparse decomposition using convex optimization with con-

straints on the distortion. Sec. 3 is on a first-order method for solving the proposed convex problem. We provide simulations in Sec. 4 and discussions in Sec. 5.

2 Convex Relaxation

In this work, we cast the MD problem into a similar form as (D.1).[1] Let $z_j \in \mathbb{R}^{M\times 1}$, $\forall j \in \mathcal{J}_J$, be the descriptions and $z_\ell = \mathrm{C}_{j\in\ell}\, z_j \in \mathbb{R}^{|\ell|M\times 1}$, $\forall \ell \in \mathcal{I}_J$, be the vector concatenation of the descriptions used in the decoding when the subset $\ell \subseteq \mathcal{J}_J$ is received. We then form the linear reconstruction functions $g_\ell(z_\ell) = D_\ell z_\ell$, $\forall \ell \in \mathcal{I}_J$, see also [12]. The dictionaries are given as $D_\ell = \mathrm{C}_{j\in\ell}\, \bar{D}_{\ell,j}$ with $D_\ell \in \mathbb{R}^{M\times|\ell|M}$, $\forall \ell \in \mathcal{I}_J$, and $\bar{D}_{\ell,j} = \rho_{\ell,j} D_j$, $\forall \ell \in \mathcal{I}_J, j \in \ell$. We choose:

- the reconstruction weight

$$\rho_{\ell,j} = \begin{cases} 1 & \text{if} \quad |\ell| = 1 \\ \frac{\sum_{i\in\ell\setminus j}\delta_i^2}{(\|\ell\|-1)\sum_{i\in\ell}\delta_i^2}, & \text{otherwise} \end{cases},$$

  in order to weight the joint reconstruction relative to the distortion bound of the individual distortions, see [15],

- D_j, $\forall j \in \mathcal{J}_J$ invertible, the reason for such will become clear in Sec. 3,

- the Euclidean norm as the measure $d(x, y) = \|x - y\|_2$.

With these choices we obtain the *standard multiple-description l_1-compression* (SMDL1C) problem

$$\begin{aligned} \text{min.} \quad & \sum_{j\in\mathcal{J}_J} \lambda_j \|W_j z_j\|_1 \\ \text{s.t.} \quad & \|D_\ell z_\ell - y\|_2 \le \delta_\ell, \qquad \forall \ell \in \mathcal{I}_J\,, \end{aligned} \tag{D.2}$$

for $\delta_\ell > 0$, $\forall \ell \in \mathcal{I}_J$, and $\lambda_j > 0, W_j \succ 0$, $\forall j \in \mathcal{J}_J$. The problem (D.2) is a second-order cone program (SOCP) [16].

For Gaussian sources with the Euclidean fidelity criterion, it has been shown that linear reconstruction functions are sufficient for achieving the MD rate-distortion function, see [17, 18] and [19] for white and colored Gaussian sources, respectively.

In (D.2) we have introduced $W_j = \mathrm{diag}(w_j)$, $\forall j \in \mathcal{I}_J$, to balance the cost of the coefficients with small and large magnitude [20]. To find w_j, the problem (D.2) is first solved with $w_j = \mathbf{1}$. Then w_j is chosen approximately inversely

[1] This work was presented in part for the case of $J = 2$ in [14].

proportional to the solution $z_j^\star$ of that problem, $w_j(i) \leftarrow {}^1/_{(|z_j^\star(i)|+\tau)}$, for the ith coordinate and with a small $\tau > 0$. The problem (D.2) is then resolved with the new weighting w_j. This reweighting scheme can be iterated a number of times. The parameter λ_j in (D.2) allows weighting of the l_1-norms in order to achieve a desired ratio $\frac{\|W_j z_j\|_1}{\|W_{j'} z_{j'}\|_1}$, $\forall j, j' \in \mathcal{J}_J$.

For the SMDL1C problem there is always a solution. Since $D_j z_j = y$ has a solution then $D_\ell z_\ell = \sum_{j\in\ell} \bar{D}_{\ell,j} z_j = \sum_{j\in\ell} \rho_{\ell,j} D_j z_j = y \sum_{j\in\ell} \rho_{\ell,j} = y$, $\forall \ell \in \mathcal{I}_J$. This implies that there exists a strictly feasible point z with $\|D_\ell z_\ell - y\|_2 = 0 < \delta_\ell$, $\forall \ell \in \mathcal{I}_J$, such that Slater's condition for strong duality holds [16].

3 A First-Order Method

We are interested in solving the SMDL1C problem for image sequences, that is, large instances involving more than 10^6 variables. First-order methods have proved efficient for large scale problems [21–23]. However, such methods require projection onto the feasible set, which might prove inefficient because the projection on a set of coupled constraints requires yet another iterative method such as alternating projection. Also, if we apply alternating projection then we will only obtain a sub-optimal projection which might generate irregularity in the first-order master method.

A problem with coupled constraints is when variable components are coupled in different constraints. To exemplify the coupled constraints, note that in the case where we let $J = 2$, we see that the constraints for $\ell = 1$ or $\ell = 2$ can easily be fulfilled by simply thresholding the smallest coefficients to zero in the transform domain D_ℓ independently. This will, however, not guarantee the joint reconstruction constraint $\|D_{\{1,2\}} z_{\{1,2\}} - y\|_2 \leq \delta_{\{1,2\}}$ which then corresponds to the coupling of the variables z_1 and z_2.

3.1 Dual Decomposition

Dual decomposition is a method to decompose coupled constraints if the objective function is decoupled [24, 25]. A dual problem of (D.2) can be represented as

$$\begin{aligned} \text{max.} \quad & -\sum_{\ell\in\mathcal{I}_J} \Big(\delta_\ell \|t_\ell\|_2 + y^T t_\ell\Big) \\ \text{s.t.} \quad & \|u_j\|_\infty \leq \lambda_j,\ \forall\, j \in \mathcal{J}_J,\ t_\ell \in \mathbb{R}^{M\times 1},\ \forall\, \ell \in \mathcal{I}_J \backslash \mathcal{J}_J \\ & t_j = -\left(D_j^{-T} W_j u_j + \sum_{\ell\in c_j(\mathcal{I}_J)\backslash j} D_j^{-T} \bar{D}_{\ell,j}^T t_\ell\right),\ \forall j \in \mathcal{J}_J, \end{aligned} \tag{D.3}$$

with optimal objective $g^\star$ and

$$c_j(\mathcal{I}) = \{\ell \,|\, \ell \in \mathcal{I}, j \in \ell\}\,.$$

The equality constraints in (D.3) are simple because t_ℓ, $\forall \ell \in \mathcal{J}_J$, are isolated on the left hand side, while the remaining variables t_ℓ, $\forall \ell \in \mathcal{I}_\mathcal{J} \backslash \mathcal{J}_\mathcal{J}$, are on the right side. We could then make a variable substitution of t_ℓ, $\forall \ell \in \mathcal{J}_J$, in the objective function. However, we choose the form (D.3) for clarity. The problem (D.3) is then decoupled in the constraints but coupled in the objective function which makes the problem (D.3) appropriate for first-order methods. Note that if the dictionaries D_j, $\forall j \in \mathcal{I}_J$ are not invertible we could not easily make a variable substitution and instead needed to handle the vector equality involving the matrix dictionaries explicitly. Indeed a difficult problem for large scale MD instances.

3.2 Primal recovery

Recovery of optimal primal variables from optimal dual variables can be accomplished if there is a unique solution to the minimization of the Lagrangian, usually in the case of a strictly convex Lagrangian [16, §5.5.5]. Define the primal variables $h_\ell = D_\ell z_\ell - y$, $\forall \ell \in \mathcal{I}_J$, and $x_j = W_j z_j$, $\forall j \in \mathcal{J}_J$, and the Lagrangian at optimal dual variables is then given as

$$\begin{aligned}\mathcal{L}(z, x, h, t^\star, u^\star, \kappa^\star) = \sum_{j \in \mathcal{J}_J} \lambda_j \|x_j\|_1 + \sum_{\ell \in \mathcal{I}_J} \kappa_\ell^\star (\|h_\ell\|_2 - \delta_\ell) \\ + \sum_{\ell \in \mathcal{I}_J} t_\ell^{*T} (D_\ell z_\ell - y - h_\ell) + \sum_{j \in \mathcal{J}_J} u_j^{*T} (W_j z_j - x_j)\,.\end{aligned}$$

However the Lagrangian associated to the problem is not strictly convex in x due to the $\|\cdot\|_1$-norm. Instead, lets consider the Karush-Kuhn-Tucker (KKT) conditions for the sub-differentiable problem (D.2) given as

$$\begin{cases} h_2(D_\ell z_\ell^\star - y)\kappa_\ell^\star - t_\ell^\star \ni 0, \quad \forall \ell \in \mathcal{I}_J \\ \kappa_\ell^\star (\|D_\ell z_\ell^\star - y\|_2 - \delta_\ell) = 0, \quad \forall \ell \in \mathcal{I}_J \qquad (\|t_\ell^\star\|_2 = \kappa_\ell^\star) \\ \sum_{\ell \in c_j(\mathcal{I}_\mathcal{J})} \bar{D}_{\ell,j}^T t_\ell^\star + W_j u_j^\star = 0, \quad \forall j \in \mathcal{J}_J \\ \|D_\ell z_\ell^\star - y\|_2 \leq \delta_\ell, \quad \forall \ell \in \mathcal{I}_J \\ \lambda_j h_1(W_j z_j^\star) - u_j^\star \ni 0, \quad \forall j \in \mathcal{J}_J \end{cases}$$

with $h_a(x) = \partial\|x\|_a$. We can rewrite the above system using $\delta_\ell > 0, \forall \ell \in \mathcal{I}_J$ and obtain the equivalent KKT optimality conditions

$$\begin{cases} \sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^\star\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T D_\ell z_\ell^\star = r_j, \quad \forall j \in \mathcal{J}_J \quad \text{(D.4.}\triangle\text{)} \\ \|D_\ell z_\ell^\star - y\|_2 \leq \delta_\ell, \quad \forall \ell \in \mathcal{I}_J \\ \lambda_j h_1(W_j z_j^\star) - u_j^\star \ni 0, \quad \forall j \in \mathcal{J}_J \end{cases} \tag{D.4}$$

where

$$r_j = -W_j u_j^\star + \sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^\star\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T y, \quad \forall j \in \mathcal{J}_J \,.$$

The equations (D.4.$\triangle$) can solved with low complexity for invertible dictionaries. However, the remaining equations are sub-differentiable and feasibility equations and are too difficulty to handle. Especially for large scale problems. Also, for a sub-optimal dual solution it is not possible to find a primal solutions that fulfills (D.4), because this implies that the dual solution is in fact an optimal dual solution. That is, for a sub-optimal dual solution we can only solve a subset of the KKT system.

Let $z^\star \in \mathcal{Z}$ be a solution to (D.4) and let $\bar{z} \in \bar{\mathcal{Z}}$ be a solution to the square system (D.4.$\triangle$). Then the following proposition shows that it is in fact possible to recover optimal primal variables in certain cases.

Proposition 3.1. *(Uniqueness) If the solution $\bar{z}$ to the linear system (D.4.$\triangle$) is unique, then $z^\star = \bar{z}$ for the SMDL1C problem.*

Proof. Since the SMDL1C problem has a solution and the system (D.4.$\triangle$) is a subsystem of (D.4) then $\emptyset \neq \mathcal{Z} \subseteq \bar{\mathcal{Z}}$. If $|\bar{\mathcal{Z}}| = 1$ then $|\mathcal{Z}| = 1$ such that $\bar{z} = z^\star$. ■

In the first-order method, from the dual sub-optimal iterates $\left(t^{(i)}, u^{(i)}\right)$, the primal iterate $z^{(i)}$ is obtained as the solution to

$$\sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^{(i)}\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T D_\ell z_\ell^{(i)} = -W_j u_j^{(i)} + \sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^{(i)}\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T y, \, \forall j \in \mathcal{J}_J.$$

The algorithm is halted if it is a primal-dual ϵ-solution

$$f(z^{(i)}) - g(t^{(i)}) \leq \epsilon, \quad z^{(i)} \in Q_\mathrm{p}, \, \left(t^{(i)}, u^{(i)}\right) \in Q_\mathrm{d} \,,$$

where Q_p and Q_d defines the primal and dual feasible set, respectively. We select $\epsilon = MJ\epsilon_\mathrm{r}$ with $\epsilon_\mathrm{r} = 10^{-3}$ to scale the accuracy ϵ in the dimensionality of the primal variables.

3.3 Complexity

The objective of the dual problem (D.3) is differentiable on $\|t_\ell\| > 0$ and sub-differentiable on $\|t_\ell\|_2 = 0$. The objective in the dual problem (D.3) is hence not smooth. A smooth function is a function with Lipschitz continuous derivatives [26]. We could then apply an algorithm such as the sub-gradient algorithm with complexity $\mathcal{O}(1/\epsilon^2)$ where ϵ is the accuracy in function value. However, it was recently proposed to make a smooth approximation and apply an optimal first-order method to the smooth problem and obtain complexity $\mathcal{O}(\frac{1}{\epsilon})$ [27]. We can not efficiently apply the algorithm in [27], since this requires projections on both the primal and dual feasible set. We will instead show how to adapt the results of [27], similar to [28], using only projection on the dual set and still achieve complexity $\mathcal{O}(\frac{1}{\epsilon})$. Consider

$$\|x\|_2 = \max_{\|v\|_2 \leq 1} \left\{ v^T x \right\}$$

and the approximation

$$\begin{aligned} \Psi_\mu(x) &= \max_{\|v\|_2 \leq 1} \left\{ v^T x - \frac{\mu}{2} \|v\|_2^2 \right\} \\ &= \begin{cases} \|x\|_2 - \mu/2, & \text{if} \quad \|x\|_2 \geq \mu \\ \frac{1}{2\mu} x^T x, & \text{otherwise} \end{cases}, \end{aligned}$$

where $\Psi_\mu(\cdot)$ is a Huber function with parameter $\mu \geq 0$. For $\mu = 0$ we have $\Psi_0(x) = \|x\|_2$. The function $\Psi_\mu(x)$ has for $\mu > 0$ the (Lipschitz continuous) derivative

$$\nabla \Psi_\mu(x) = \frac{x}{\max\{\|x\|_2, \mu\}} .$$

The dual objective is

$$g(t) = -\sum_{\ell \in \mathcal{I}_J} \left(\delta_\ell \|t_\ell\|_2 + y^T t_\ell \right)$$

and we can then form the smooth function g_μ

$$g_\mu(t) = -\sum_{\ell \in \mathcal{I}_J} \left(\delta_\ell \Psi_\mu(t_\ell) + y^T t_\ell \right).$$

The Lipschitz constant of the gradient is $L(\nabla \Psi_\mu) = \frac{1}{\mu}$ and

$$L_\mu = L(\nabla g_\mu) = \left(\sum_{\ell \in \mathcal{I}_J} \frac{\delta_\ell}{\mu} + 1 \right) = \frac{C}{\mu} + |\mathcal{I}_J| . \tag{D.5}$$

The smooth function has the approximation

$$g_\mu(t) \leq g(t) \leq g_\mu(t) + \mu C\,. \tag{D.6}$$

Hence, the parameter μ both controls the level of smoothness (D.5) and the approximation accuracy (D.6). Select $\mu = \epsilon/(2C)$ and let the ith iteration $t^{(i)}$ of a first-order method have the property

$$g_\mu^\star - g_\mu(t^{(i)}) \leq \frac{\epsilon}{2}\,,$$

where $g_\mu^\star$ is the optimal objective for the smooth problem. Then we obtain

$$g^\star - g(t^{(i)}) \leq g_\mu^\star + \mu C - g_\mu(t^{(i)}) \leq \epsilon\,.$$

By using an optimal-first order algorithm for L-smooth problems with complexity $\mathcal{O}\left(\sqrt{\frac{L}{\epsilon}}\right)$ [26], then $t^{(i)}$ can be obtained in i iterations, where

$$i = \mathcal{O}\left(\sqrt{\frac{L_\mu}{\epsilon}}\right) = \mathcal{O}\left(\sqrt{\frac{1}{\epsilon^2} + \frac{1}{\epsilon}}\right) \leq \mathcal{O}\left(\sqrt{\frac{1}{\epsilon^2}} + \sqrt{\frac{1}{\epsilon}}\right) = \mathcal{O}\left(\frac{1}{\epsilon}\right)\,.$$

4 Simulations

For the simulations we will present an example of obtaining a sparse decomposition in the presented MD framework. As the source we select the grayscale image sequence of "foreman" with height $m = 288$ pixels and width $n = 352$ pixels. We jointly process $k = 8$ consecutive frames [29] and y is formed by stacking each image and scaled such that $y \in [0;1]^M$, $M = mnk$. We select $J = 3$ and as dictionaries

D_1 : the three dimensional cosine transform,

D_2 : a two dimensional Symlet16 discrete wavelet transform with 5 levels along the dimensions associated to m,n and a one dimensional Haar discrete wavelet transform with 3 levels along the dimension associated to k,

D_3 : the three dimensional sine transform.

Let the peak signal-to-noise ratio (PSNR) measure be defined by

$$\mathrm{PSNR}(\delta) = 10\log_{10}\left(\frac{1}{\frac{1}{M}\delta^2}\right)\,.$$

As distortion constraints we select $\mathrm{PSNR}(\delta_\ell) = 30$, $\forall|\ell| = 1$, $\mathrm{PSNR}(\delta_\ell) = 33$, $\forall|\ell| = 2$ and $\mathrm{PSNR}(\delta_\ell) = 37$, $|\ell| = 3$ with $\ell \in \mathcal{I}_J$. Further we choose equal weights $\lambda_j = 1$, $\forall j \in \mathcal{J}_J$.

If the primal variables were obtained from an algorithm using projection [21] or a method employing a soft-thresholding operator [22], a sub-optimal solution will contain coefficients which are exactly zero. The primal variables are in this approach obtained as the solution to a linear system arising from sub-optimal dual variables and hence there might be many small coefficients which are not exactly zero. To handle this, the distortion requirement are changed by $\bar{\delta}_\ell = \delta_\ell - |\ell|\sigma$, $\forall \ell \in \mathcal{I}_J$, with $\sigma > 0$ when the SMDL1C problem is solved and the smallest coefficients are afterwards thresholded to zero using the slack introduced by $|\ell|\sigma$ while ensuring the original distortion constraints δ_ℓ. Let $z(r)$ be an ϵ-solution after r reweight iterations of the SMDL1C problem and set $\hat{z} = z(7)$.

4.1 Example

Define a frame extraction function $s(y, i)$ which extract the ith frame from the image sequence stacked in y. In Fig. D.2 we show a few examples of the decoded 6th frame for the subset $\ell = \{1\}$, $\ell = \{2, 3\}$ and $\ell = \{1, 2, 3\}$. This example is a large scale problem with $10 \cdot 10^6$ primal-dual variables.

4.2 Reweighting

In Fig. D.3 we report the relative cardinality of $z(r)$ as a function of the number of applied reweight iterations r. We observe in Fig. D.3 that the cardinality is significantly decreased for $r \in \{0, \cdots, 3\}$, whereupon the decrease is less distinct.

4.3 Threshold Comparison

For comparison we will obtain sparse decompositions in each basis independently using thresholding. We define the operation $z = T(D, y, \gamma)$ as thresholding the coefficients with the smallest magnitude in the basis D from the source y such that $\text{PSNR}(\|Dz - y\|_2) \approx \gamma$. We report the relative cardinalities $\text{card}(z)/M$ and PSNR measures obtained by SMDL1C in Tab. D.1 and by independent thresholding for each basis in Tab. D.2. When using SMDL1C we observe that from $|\ell| = 1$ to $|\ell| = 2$ the descriptions obtain a refinement in the range $3.2 - 4.8\,\text{dB}$. For independent thresholding the refinement is smaller, in the range $0.7 - 1.2\,\text{dB}$. This shows that the obtained refinement by the SMDL1C method is non-trivial.

The PSNR measures for thresholding to same cardinality as using SMDL1C are reported in Tab D.3. The descriptions are formed independently and refinement is there not guaranteed, which we can observe when the reconstructions at level $|\ell| = 2$ are combined to reconstruction at level $|\ell| = 3$.

The cardinalities for thresholding at PSNR $37.2\,\text{dB}$ are given in Tab. D.4. By comparing Tab. D.1 and D.4, we see that the cardinalities of SMDL1C are

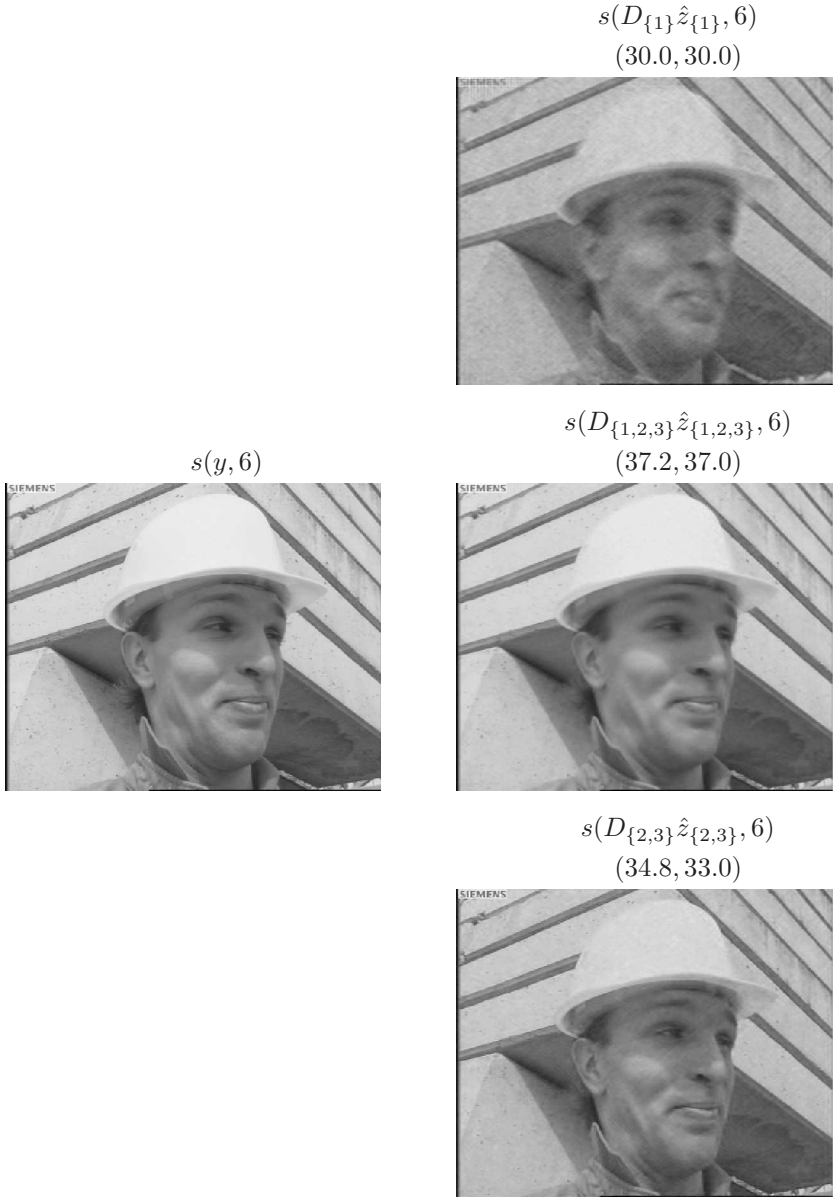


Fig. D.2: Example using "foreman" (grayscale, 288×352). The images show the 6th frame of the decoded images for $\ell = \{1\}$, $\ell = \{2, 3\}$ and $\ell = \{1, 2, 3\}$. Above the figures are the actual distortion and the distortion bounds reported using the format $(\mathrm{PSNR}(\|D_\ell \hat{z}_\ell - y\|_2), \mathrm{PSNR}(\delta_\ell))$.

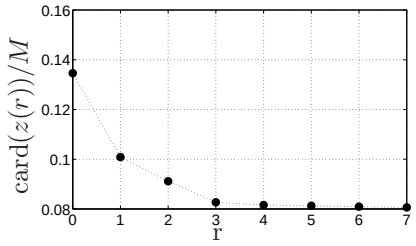


Fig. D.3: Example of reweighting an image sequence of "foreman" (grayscale, 288×352) by jointly processing $k=8$ frames.

$\mathrm{card}(z_j)/M$	$\mathrm{PSNR}(\|D_\ell z_\ell - y\|_2)$		
$j=1$	$\ell=\{1\}$	$\ell=\{1,2\}$	
0.019	30.0	34.4	
$j=2$	$\ell=\{2\}$	$\ell=\{1,3\}$	$\ell=\{1,2,3\}$
0.025	30.0	33.2	37.2
$j=3$	$\ell=\{3\}$	$\ell=\{2,3\}$	
0.037	30.0	34.8	

Table D.1: Cardinality and reconstruction PSNR for SMDL1C ($z \leftarrow \hat{z}$), with $\mathrm{card}(z)/M = 0.081$.

$\mathrm{card}(z_j)/M$	$\mathrm{PSNR}(\|D_\ell z_\ell - y\|_2)$		
$j=1$	$\ell=\{1\}$	$\ell=\{1,2\}$	
0.012	30.0	30.9	
$j=2$	$\ell=\{2\}$	$\ell=\{1,3\}$	$\ell=\{1,2,3\}$
0.006	30.0	30.7	31.3
$j=3$	$\ell=\{3\}$	$\ell=\{2,3\}$	
0.019	30.0	31.2	

Table D.2: Cardinalities and reconstruction PSNRs for thresholding ($z \leftarrow \mathrm{C}_{j\in\mathcal{J}_J}\, T(D_j, y, 30)$), with $\mathrm{card}(z)/M = 0.037$.

$\mathrm{card}(z_j)/M$	$\mathrm{PSNR}(\|D_\ell z_\ell - y\|_2)$		
$j=1$	$\ell=\{1\}$	$\ell=\{1,2\}$	
0.019	31.2	34.8	
$j=2$	$\ell=\{2\}$	$\ell=\{1,3\}$	$\ell=\{1,2,3\}$
0.025	34.7	32.5	34.5
$j=3$	$\ell=\{3\}$	$\ell=\{2,3\}$	
0.037	32.1	35.1	

Table D.3: Cardinalities and reconstruction PSNRs for thresholding to same cardinality as using SMDL1C.

smaller than that of simple tresholding at 37.2 dB. Also, by comparing Tab. D.1 and D.2, we see that the cardinalities of SMDL1C are larger than that of simple tresholding at 30.0 dB. These bounds are to be expected for non-trivial descriptions. We also note that if we used the dictionary with the smallest cardinality to achieve the requested PSNR ($j=2$), it is not possible to duplicate this description at the highest PSNR before the total cardinality exceeds that of $\hat{z}$. This exemplifies that it is not possible to simply transmit the coefficients $T(D_2, y, 37.2)$ over all channels and obtain a comparable cardinality as obtained by the SMDL1C problem.

	$j=1$	$j=2$	$j=3$
$\mathrm{card}(z_j)/M$	0.107	0.048	0.122

Table D.4: Cardinalities using thresholding at the highest obtained PSNR by SMDL1C ($z \leftarrow \bigcup_{j\in\mathcal{J}_J} T(D_j, y, 37.2)$).

5 Discussion

We presented a multiple description formulation using convex relaxation. In the case of large-scale problems we have proposed a first-order method for the dual problem. The simulations showed that the proposed multiple description formulation renders non-trivial descriptions with respect to both the cardinality and the refinement.

References

[1] S. Mallat, *A Wavelet tour of signal processing, Third Edition: The Sparse Way*. Academic Press, 2009.

[2] D. Giacobello, M. G. Christensen, J. Dahl, S. H. Jensen, and M. Moonen, "Sparse linear predictors for speech processing," in *Proc. Ann. Conf. Int. Speech Commun. Ass. (INTERSPEECH)*, Brisbane, Australia, Sep. 2008, pp. 1353–1356.

[3] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Comp.*, vol. 13, no. 4, pp. 863–882, 2001.

[4] B. A. Olshausen and D. J. Field, "Natural image statistics and efficient coding," *Network: Computation in Neural Systems*, vol. 7, no. 2, pp. 333–339, 1996.

[5] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[6] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[7] A. A. E. Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 28, no. 6, pp. 851 – 857, Nov. 1982.

[8] L. Ozarow, "On a source-coding problem with two channels and three receivers," *Bell System Technical Journal*, vol. 59, pp. 1909 – 1921, Dec. 1980.

[9] U. G. Sun, J. Liang, C. Tian, C. Tu, and T. Tran, "Multiple desciption coding with prediction compensation," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 1037–1047, 2009.

[10] H. Chan and C. Huang, "Multiple description and matching pursuit coding for video transmission over the internet," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Hong Kong, Apr. 2003, pp. 425–428.

[11] T. Nguyen and A. Zakhor, "Matching pursuits based multiple description video coding for lossy environments," in *Proc. Int. Conf. on Image Process. (ICIP)*, Sep. 2003, pp. 57–60.

[12] T. Petrisor, B. Pesquet-Popescu, and J.-C. Pesquet, "A compressed sensing approach to frame-based multiple description coding," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Honolulu, Hawaii, Apr. 2007, pp. 709–712.

[13] Y. Zhang, S. Mei, Q. Chen, and Z. Chen, "A multiple description image/video coding method by compressed sensing theory," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, Seattle, Washington, May. 2008, pp. 1830–1833.

[14] T. L. Jensen, J. Dahl, J. Østergaard, and S. H. Jensen, "A first-order method for the multiple-description l1-compression problem," Signal Processing with Adaptive Sparse Structured Representations (SPARS'09), Saint-Malo, France, Apr. 2009.

[15] S. Diggavi, N. Sloane, and V. Vaishampayan, "Asymmetric multiple description lattice vector quantizers," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 174 – 191, Jan. 2002.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[17] J. Chen, C. Tian, T. Berger, and S. S. Hemami, "Multiple description quantization via Gram-Schmidt orthogonalization," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5197 – 5217, Dec. 2006.

[18] J. Østergaard and R. Zamir, "Multiple description coding by dithered delta-sigma quantization," *IEEE Trans. Inf. Theory*, vol. 55, no. 10, pp. 4661–4675, Oct. 2009.

[19] Y. Kochman, J. Østergaard, and R. Zamir, "Noise-shaped predictive coding for multiple descriptions of a colored Gaussian source," in *Proc. IEEE Data Comp. Conf. (DCC)*, Snowbird, Utah, Mar. 2008, pp. 362 – 371.

[20] E. Candès, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, 2008.

[21] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Top. Sign. Proces.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[22] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, 2009.

[23] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algo.*, vol. 50, no. 1, pp. 67–92, 2010.

[24] G. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101 – 111, Jan.-Feb. 1960.

[25] D. P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 1995.

[26] Y. Nesterov, *Introductory Lectures on Convex Optimization.* Kluwer Academic Publishers, 2004.

[27] ——, "Smooth minimization of nonsmooth functions," *Math. Prog. Series A*, vol. 103, pp. 127–152, 2005.

[28] L. Vandenberghe, "Optimization methods for large-scale systems," Lecture Notes, 2009.

[29] J. Dahl, J. Østergaard, T. L. Jensen, and S. H. Jensen, "ℓ_1 compression of image sequences using the structural similarity index measure," in *Proc. IEEE Data Comp. Conf. (DCC)*, Snowbird, Utah, Mar. 2009, pp. 133 – 142.

Paper E

Multiple-Description l_1-Compression

T. L. Jensen, J. Østergaard, J. Dahl and S. H. Jensen

This paper is published in
IEEE Transactions on Signal Processing,
Vol. 59, No. 8, pp. 3699–3711, 2011.

The layout is revised.

Minor spelling, gramma and notation errors have been corrected.

Abstract

Multiple descriptions (MDs) is a method to obtain reliable signal transmissions on erasure channels. An MD encoder forms several descriptions of the signal and each description is independently transmitted across an erasure channel. The reconstruction quality then depends on the set of received descriptions. In this paper, we consider the design of redundant descriptions in an MD setup using l_1-minimization with Euclidean distortion constraints. In this way we are able to obtain sparse descriptions using convex optimization. The proposed method allows for an arbitrary number of descriptions and supports both symmetric and asymmetric distortion design. We show that MDs with partial overlapping information corresponds to enforcing coupled constraints in the proposed convex optimization problem. To handle the coupled constraints, we apply dual decompositions which makes first-order methods applicable and thereby admit solutions for large-scale problems, e.g., coding entire images or image sequences. We show by examples that the proposed framework generates non-trivial sparse descriptions and non-trivial refinements. We finally show that the sparse descriptions can be quantized and encoded using off-the-shelf encoders such as the set partitioning in hierarchical trees (SPIHT) encoder, however, the proposed method shows a rate-distortion loss compared to state-of-the-art image MD encoders.

1 Introduction

An important problem in signal processing is the *multiple-description* (MD) problem [1]. The MD problem is on encoding a source into multiple descriptions, which are transmitted over separate channels. The channels may occasionally break down causing description erasures, in which case only a subset of the descriptions are received. Which of the channels that are working at any given time is known by the decoder but not by the encoder. The problem is then to construct a number of descriptions, which individually provide an acceptable quality and furthermore are able to refine each other. It is important to notice the contradicting requirements associated with the MD problem; in order for the descriptions to be individually good, they must all be similar to the source and therefore, to some extend, the descriptions are also similar to each other. However, if the descriptions are the same, they cannot refine each other.

Let J be the number of channels and let $\mathcal{J}_J = \{1, \ldots, J\}$. Then $\mathcal{I}_J = \{\ell \,|\, \ell \subseteq \mathcal{J}_J,\, \ell \neq \emptyset\}$ describes the indices of the non-trivial subsets which can be received. Further, let z_j denote the jth description and define $z_\ell = \{z_j \mid j \in \ell\}$, $\forall \ell \in \mathcal{I}_J$. At the decoder, the descriptions z_ℓ, $\ell \in \mathcal{I}_J$, approximate the source y via their individual reconstructions $g_\ell(z_\ell)$ which satisfy the fidelity constraint $d(g_\ell(z_\ell), y) \leq \delta_\ell$, $\forall \ell \in \mathcal{I}_J$, with $d(\cdot, \cdot)$ denoting a distortion measure. An example

with $J = 2$ is illustrated in Fig. E.1.

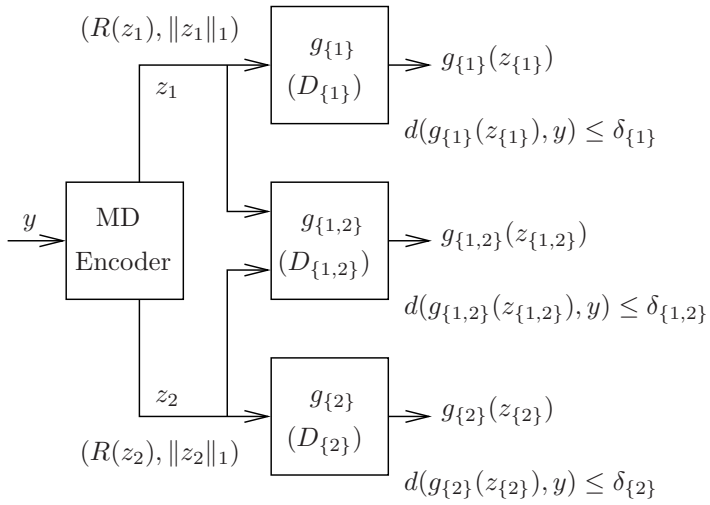


Fig. E.1: The MD (l_1-compression) problem for $J = 2$.

The traditional MD coding problem aims at characterizing the set of achievable tuples
$\big(R(z_1), R(z_2) \cdots, R(z_J), \delta_{\{1\}}, \cdots, \delta_{\{1,2\cdots,J\}}\big)$ where $R(z_j)$ denotes the minimum coding rate for description z_j, $\forall j \in \mathcal{J}_J$, required in order to approximate the source y to within the distortion fidelities δ_ℓ, $\forall \ell \in \mathcal{I}_J$ [1]. The problem is then to construct z_ℓ, $\forall \ell \in \mathcal{I}_J$, so that $R(z_j)$, $\forall j \in \mathcal{J}_J$, are minimized and the fidelity constraints are satisfied, cf., Fig. E.1. This well-known information theoretic problem remains largely unsolved. In fact, it is only completely solved for the case of two descriptions, with the squared error fidelity criterion and Gaussian sources [2].

Another direction is to form descriptions in a deterministic setting, as opposed to the traditionally MD approach [1]. Algorithms designed for video and image coding may be based on, *e.g.*, Wiener filters with prediction compensation [3], matching pursuit [4, 5] or compressed sensing [6, 7]. Recovery of the latent variables can in compressed sensing be obtained by sparsity driven methods such as l_1-minimization with known guarantees [8]. There is also results in the case of quantization [9–11].

In this paper we propose a convex problem, which can be used to obtain sparse descriptions for MD problems using l_1-minimization with Euclidean constrains on the distortion of the reconstruction. The proposed MD formulation is flexible in terms of applications (*e.g.*, speech, image and video compression),

the number of channels J as well as supporting both symmetric and asymmetric design. We show how to apply a first-order method to solve the proposed convex optimization problem using dual decomposition and smoothing [12]. Let ϵ be the desired accuracy of an approximate solution in function value, in which case the first-order method has iteration complexity $\mathcal{O}\left(\frac{1}{\epsilon}\right)$. The combination of a reasonable iteration complexity and the low complexity of a single iteration in first-order methods makes it possible to apply the proposed MD method to large scale problems such as for entire images or image sequences. The descriptions are for example represented in discrete wavelet dictionaries but arbitrary dictionaries are allowed in the original formulation. For encoding the sparse descriptions it is possible to apply state-of-the-art methods for wavelet encoding, *e.g.*, set partitioning in hierarchical trees (SPIHT) [13]. However, we are not able to obtain state-of-the-art rate-distortion descriptions by the two stage approach of first forming sparse descriptions and then encode.

The organization of the paper is as follows: we will first propose the MD l_1-compression (MDl1C) problem in Sec. 2 and then analyze and discuss important properties of the problem in Sec. 3. Then, in Sec. 4, we discuss algorithms for solving the proposed convex problem, and present an efficient first-order method. We analyze the sparse descriptions in Sec. 5 and extend the framework to encoding of MD wavelet coefficient based on well known methods and provide simulations on compression of images and image sequences in Sec. 6.

2 Problem Formulation

An interesting direction of research is in sparse estimation techniques for signal processing based on l_1-norm heuristics, where, *e.g.*, compressive sampling [8, 14] have gained much attention. The theory is by now well-established and much is known about cases where the l_1-minimization approach coincides with the solution to the otherwise intractable minimum cardinality solution, see [15] and references therein.

One way of obtaining a sparse approximation z of the source y is to solve the so-called l_1-*compression* problem

$$\begin{array}{ll} \text{minimize} & \|Wz\|_1 \\ \text{subject to} & \|Dz - y\|_2 \leq \delta, \end{array} \tag{E.1}$$

where $\delta > 0$ is a given distortion bound, $D \in \mathbb{R}^{M \times N}$ is an overcomplete dictionary ($N \geq M$), $z \in \mathbb{R}^N$ is the variable, and $y \in \mathbb{R}^M$ is the signal we wish to decompose into a sparse representation. In a standard formulation $W \in \mathbb{R}^{N \times N}$ can be selected as $W = I$. To improve the l_1-minimization approach for minimizing the cardinality it has been proposed to select $W = \text{diag}(w)$ to reduce the

cost of large coefficients [16], see also [17]. To find w, the unscaled problem (E.1) is solved first (*i.e.*, with $w = \mathbf{1}$). Then w is chosen inversely proportional to the solution z^* of that problem, and (E.1) is solved again with the new weighting w. This reweighting scheme can be iterated a number of times.

In this work, we cast the MD problem into the framework of l_1-compression (E.1).[1] Let $z_j \in \mathbb{R}^{\tilde{N}_j \times 1}$, $\forall j \in \mathcal{J}_J$, be the descriptions of length $\tilde{N}_j$. We will define a concatenation operator

$$X = \bigcirc_{i \in S} Y_i = \begin{bmatrix} Y_{S_1} \\ Y_{S_2} \\ \vdots \\ Y_{S_n} \end{bmatrix}$$

where $Y_i \in \mathbb{R}^{p_i \times q}$, $S = \{S_1, S_2, \cdots, S_n\}$ has n elements and $X \in \mathbb{R}^{\sum_{i \in S} p_i \times q}$. Then $z_\ell = \bigcirc_{j \in \ell} z_j \in \mathbb{R}^{\sum_{j \in \ell} \tilde{N}_j \times 1}$, $\forall \ell \in \mathcal{I}_J$, is the vector concatenation of the descriptions used in the decoding when the subset $\ell \subseteq \mathcal{J}_J$ is received. For simplicity we will use z_j with the meaning $z_{\{j\}}$, $j \in \mathcal{J}_J$, which also applies to other symbols with subscripted ℓ. The matrix $D_\ell \in \mathbb{R}^{M \times \sum_{j \in \ell} \tilde{N}_j}$, $\forall \ell \in \mathcal{I}_J$, is the dictionary associated with the description z_ℓ given as $D_\ell = \left(\bigcirc_{j \in \ell} \bar{D}_{\ell,j}^T \right)^T$ with $\bar{D}_{\ell,j} \in \mathbb{R}^{M \times \tilde{N}_j}$. Our idea is to form the *multiple-description l_1-compression* problem using linear reconstruction functions, *i.e.*, $g_\ell(z_\ell) = D_\ell z_\ell$ similar to [6] since it preserves convexity [20], and the Euclidean norm as the distortion measure, *i.e.*, $d(x, y) = \|x - y\|_2$.[2] Note that its possible to select other distortion measures that is convex, but we choose $\|\cdot\|_2$ since it relates to the well known peak signal-to-noise-ratio (PSNR). The definition is given below.

Definition 2.1. *An instance* $\{y, \{\delta_\ell\}_{\ell \in \mathcal{I}_J}, \{D_\ell\}_{\ell \in \mathcal{I}_J}, \{W_j\}_{j \in \mathcal{J}_J}, \{\lambda_j\}_{j \in \mathcal{J}_J}\}$ *of the MDl1C problem is defined by*

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}_J} \lambda_j \|W_j z_j\|_1 \\ \text{subject to} \quad & \|D_\ell z_\ell - y\|_2 \leq \delta_\ell, \qquad \forall \ell \in \mathcal{I}_J \,, \end{aligned} \tag{E.2}$$

for $\delta_\ell > 0$, $\forall \ell \in \mathcal{I}_J$, $\lambda_j > 0$, $\forall j \in \mathcal{J}_J$ *and* $W_j \succ 0$, $\forall j \in \mathcal{J}_J$. *For simplicity we sometime use* $f(z) = \sum_{j \in \mathcal{J}_J} \lambda_j \|W_j z_j\|_1$ *for the primal objective and* $Q_\mathrm{p} = \{z \,|\, \|D_\ell z_\ell - y\|_2 \leq \delta_\ell, \forall \ell \in \mathcal{I}_J\}$ *for the primal feasible set.*

[1] This work was presented in part [18, 19].

[2] Interestingly, in the Gaussian case and for the mean squared error fidelity criterion, it has been shown that linear reconstruction functions are sufficient for achieving the MD rate-distortion function, see [21, 22] and [23] for white and colored cases, respectively.

The problem (E.2) amounts to minimize the number of non-zero coefficients in the descriptions (using convex relaxation) under the constraint that any combination of received descriptions allows a reconstruction error smaller than some quantity. The idea is that the problem (E.2) can be used to obtain sparse coefficients which obeys certain bounds on the reconstruction error. Since it has been shown that bit rate and sparsity is almost linearly dependent [24], the problem formulation (E.2) can be used to form descriptions in a MD framework. In Sec. 6 we will discuss in detail how to encode the sparse coefficients. Note that since $|\mathcal{I}_J| = 2^J - 1$, the number of possible received combinations grows exponential in the number of channels, and thereby the number of constraints in problem (E.2).

In Definition 2.1 we have introduced $\lambda > 0$ to allow weighting of the l_1-norms in order to achieve a desired ratio $\frac{\|W_j z_j\|_1}{\|W_{j'} z_{j'}\|_1}$, $\forall j, j' \in \mathcal{J}_J$. Note that in the case where we let $\bar{D}_{j,j}$, $\forall j \in \mathcal{J}_J$, be orthogonal, we see that the constraints on the side reconstructions can easily be fulfilled by simply truncating the smallest coefficients $z_j = \bar{D}_{j,j}$, $\forall\, j \in \mathcal{J}_J$, to zero separately for the coefficients of each side description. This will, however, not guarantee the joint reconstruction constraint $\|D_\ell z_\ell - y\|_2 \leq \delta_\ell$, $\forall \ell \in \mathcal{I}_J \backslash \mathcal{J}_J$. Thus, the problem at hand is non-trivial.

In the following sections we will analyse the MDl1C problem presented in Definition 2.1 and give an algorithm to solve large scale instances of this problem.

3 Analysis of the Multiple-description l_1-Compression Problem

In this section we will review and discuss some important properties of the proposed MDl1C problem.

Definition 3.1. *(Solvable) The MDl1C problem is solvable if the problem has at least one feasible point.*

Remark (Definition 3.1) Since the MDl1C problem is always bounded below, this is the same definition as in [20].

Proposition 3.2. *(Solvable conditions) Let* $\bar{D}_{\ell,j} = \rho_{\ell,j} \bar{D}_{j,j}$, $\forall \ell \in \mathcal{I}_J$, $j \in \ell$ *with* $\rho_{\ell,j} \in \mathbb{R}$, $\sum_{j \in \ell} \rho_{\ell,j} = 1$, $\forall \ell \in \mathcal{I}_J$. *Furthermore, let* $y \in \operatorname{span}(\bar{D}_{j,j})$, $\forall j \in \mathcal{J}_J$. *Then the MDl1C problem* (E.2) *is solvable.*

Proof. There exists z_j, $\forall j \in \mathcal{J}_J$, such that $\bar{D}_{j,j} z_j = y$, $\forall j \in \mathcal{J}_J$. Then we also have that $D_\ell z_\ell = \sum_{j \in \ell} \bar{D}_{\ell,j} z_j = \sum_{j \in \ell} \rho_{\ell,j} \bar{D}_{j,j} z_j = y \sum_{j \in \ell} \rho_{\ell,j} = y$, $\forall \ell \in \mathcal{I}_J$. Hence, $z = \bigcap_{j \in \mathcal{J}_J} z_j \in Q_{\mathrm{p}}$ is a primal feasible solution and the problem (E.2) is therefore solvable. ■

One way to obtain the setup used in Proposition 3.2 is to use a standard MDl1C setup.

Definition 3.3. *(Standard MD ℓ_1-compression problem) We denote an MDl1C problem a standard MDl1C problem if*

- $\bar{D}_{j,j}$, $\forall j \in \mathcal{J}_J$, *are invertible.*
- $\rho_{\ell,j} = \begin{cases} 1 & \text{if} \quad |\ell| = 1 \\ \frac{\sum_{i\in\ell\setminus j}\delta_i^2}{(|\ell|-1)\sum_{i\in\ell}\delta_i^2}, & \text{otherwise} \end{cases}$, *to weight the contributions*
- $\bar{D}_{\ell,j} = \rho_{\ell,j} D_j$, *(combined with above $\bar{D}_{j,j} = D_j$, $\forall j \in \mathcal{J}_J$).*

Remark (Definition 3.3) In the general asymmetric case, its common to weight the reconstruction of the joint reconstructions relative to the distortion of the individual reconstruction [25, 26]. Note that in the symmetric case, $\delta_\ell = \delta_{\ell'}$, $\forall \ell, \ell' \in \mathcal{I}_J, |\ell| = |\ell'|$, we have equal weight $\rho_{\ell,j} = \rho_{\ell,i}$, $j, i \in \ell$.

Proposition 3.4. *(Strong duality) Strong duality holds for the standard MDl1C problem.*

Proof. Since $\delta_\ell > 0$, $\forall \ell \in \mathcal{I}_J$, $\text{span}(D_j) \in y$, $\forall j \in \mathcal{J}_J$, and $\sum_{j\in\ell} \rho_{\ell,j} = 1$, $\forall \ell \in \mathcal{I}_J$, there exists a strictly feasible $\|D_\ell z_\ell - y\|_2 = 0 < \delta_\ell$, $\forall \ell \in \mathcal{I}_J$, point z such that Slater's condition for strong duality holds [20]. ■

In Proposition 3.2 we assumed that $\bar{D}_{\ell,j} = \rho_{\ell,j} D_j$, $\forall \ell \in \mathcal{I}_J, j \in \ell$. We will, however, shortly discuss the case where $\bar{D}_{\ell,j} \neq \rho_{\ell,j} D_j$ for at least one pair $(\ell, j) \in \mathcal{I}_J \times \mathcal{J}_J$. The interpretation is that the dictionaries associated with the same description may not be the same in all reconstruction functions. This can be illustrated with an example where we will solve a small problem of size $M = 2$, $J = 2$, with D_j, $\forall j \in \mathcal{J}_J$, being the orthogonal discrete cosine transform. We will construct another dictionary in the central reconstruction using a rotation matrix

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

such that we solve problems on the form

$$\begin{aligned} \text{minimize} \quad & \|z_1\|_1 + \|z_2\|_1 \\ \text{subject to} \quad & \|D_1 z_1 - y\|_2 \leq \delta_1 \\ & \|D_2 z_2 - y\|_2 \leq \delta_2 \\ & \|\tfrac{1}{2}(D_1 R_\theta z_1 + D_2 z_2) - y\|_2 \leq \delta_{\{1,2\}} \end{aligned} \tag{E.3}$$

with solution $\tilde{z}^*_\theta$. By considering different θ's we obtain different central decoding functions. In Fig. E.2 we show the optimal objective $f(\tilde{z}^*_\theta)$ from solving problem (E.3). We investigate $\theta \in [-\pi/2; \pi/2]$ and only report $f(\cdot)$ and the cardinality $\text{card}(\cdot)$ if the problem (E.3) is solvable. We choose $\delta_1 = \delta_2 = \{0.2, 0.02\}$ and $\delta_{\{1,2\}} = 0.01$. Observe that for both $\delta_1 = \delta_2 = 0.2$ and $\delta_1 = \delta_2 = 0.02$, the objective $f(\cdot)$ can be reduced if we select $\theta \neq 0$, *i.e.*, if the dictionaries associated to the different decoding functions are not equal. For $\delta_1 = \delta_2 = 0.2$ the cardinality can also be reduced from 3 at $\theta = 0$ to cardinality 2 at $\theta \approx -\pi/8$. If $\|D_1 z_1 - y\|_2$ is required to be small, we would expect $|\theta|$ to be small because $D_1 z_1 \approx y$ and then $D_1 R_\theta z_1 \approx y$ for $\theta \approx 0$. Note that if $|\theta|$ is too large, then the problem is not solvable.

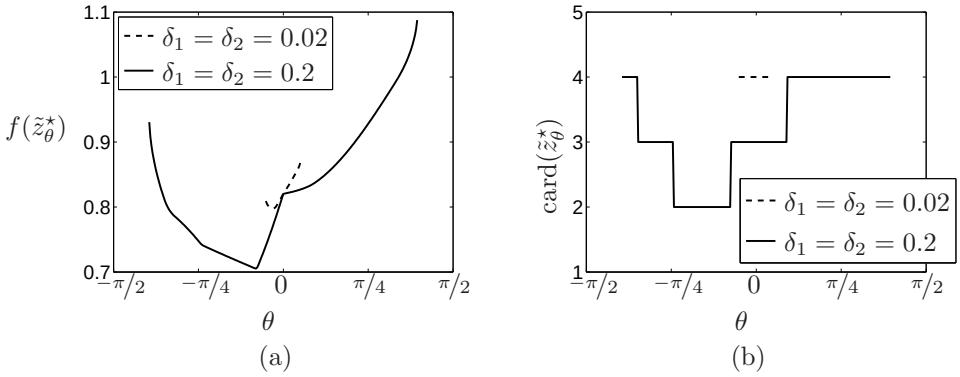


Fig. E.2: (a): optimal objective $f(\tilde{z}^*_\theta)$ and (b): the cardinality $\text{card}(\tilde{z}^*_\theta)$ from solving the problem (E.3) with $M = 2$. The distortion bounds are $\delta_1 = \delta_2 = \{0.2, 0.02\}$, $\delta_{\{1,2\}} = 0.01$ and $D_1 = D_2$: the discrete cosine transform. We only report $f(\cdot)$ and $\text{card}(\cdot)$ if the problem (E.3) is solvable.

This example illustrates that it can be useful to have different dictionaries in the decoder associated to the same description. To find such dictionaries a-priori for different applications is signal dependent, and a separate research topic, which will not be treated in this work.

4 Solving the MD l_1-Compression Problem

The MDl1C problem (E.2) can be solved using general-purpose primal-dual interior point methods. To do so, we need to solve several linear systems of equations of size $\mathcal{O}(K) \times \mathcal{O}(K)$, arising from linearizing first-order optimality conditions, with $K = M|\mathcal{I}_J| + \sum_{j \in \mathcal{J}_J} \tilde{N}_j$. This practically limits the size of the problems we can consider to small and medium size, except if the problem has a certain structure that can be used when solving the linear system of equations [27]. Another

approach is to use first-order methods [12, 28–30]. Such first-order projection methods have shown to be efficient for large scale problems [31–34]. However, it is difficult to solve the MDl1C problem efficiently because the feasible set $\|D_\ell z_\ell - y\|_2 \leq \delta_\ell$, $\forall \ell \in \mathcal{I}_J$, is an intersection of Euclidean norm balls.

We are interested in solving large-scale instances of problem (E.2), and in the following subsections 4.1 through 4.5, we will present an efficient first-order method to handle problem (E.2).

4.1 Intersecting Euclidean Norm Balls

In order to illustrate the implications of the overlapping constraints on the feasible set, consider the following simple example. Let $D_1 = D_2 = W_1 = W_2 = \lambda_1 = \lambda_2 = 1$ so that $D_1 z_1 = z_1$ and $D_2 z_2 = z_2$. From the joint constraint it may be noticed that z_1 and z_2 can be picked arbitrarily large but of different signs and yet satisfy $|\frac{1}{2}(z_1 + z_2) - y| \leq \delta_{\{1,2\}}$. However, due to the individual constraints $|z_1 - y| \leq \delta_1$ and $|z_2 - y| \leq \delta_2$, the feasible set is bounded as illustrated in Fig. E.3.

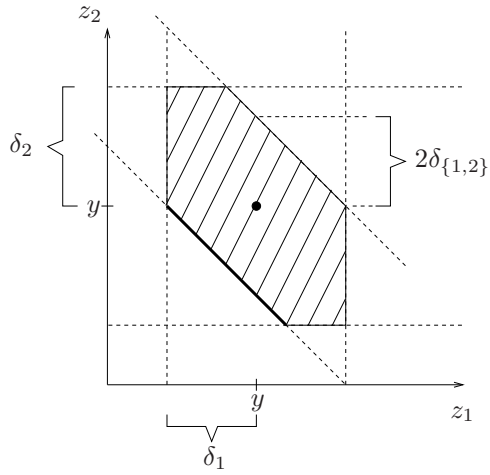


Fig. E.3: An example of the feasible set (shaded region) in $\mathbb{R}^{(1+1)\times 1}$. The thick line indicates the optimal solutions for the problem of minimizing $|z_1| + |z_2|$.

4.2 Dual Decomposition

An approach to handle problems with intersecting constraints, sometimes referred to as complicating or coupling constraints, is by dual decomposition [29, 35].

Proposition 4.1. *(Dual problem) A dual problem of the standard MDl1C problem can be represented as*

$$\begin{aligned} &\text{maximize} \quad -\sum_{\ell\in\mathcal{I}_J}\Big(\delta_\ell\|t_\ell\|_2 + y^T t_\ell\Big) \\ &\text{subject to} \quad \|u_j\|_\infty \le \lambda_j,\ \forall\, j\in\mathcal{J}_J, \quad t_\ell\in\mathbb{R}^{M\times 1},\ \forall\,\ell\in\mathcal{I}_J\backslash\mathcal{J}_J \\ &\qquad\qquad t_j = -\left(\bar{D}_{j,j}^{-T}W_j u_j + \sum_{\ell\in c_j(\mathcal{I}_J)\backslash j}\bar{D}_{j,j}^{-T}\bar{D}_{\ell,j}^T t_\ell\right),\ \forall\, j\in\mathcal{J}_J\,, \end{aligned} \tag{E.4}$$

where $g(t)$ is the dual objective and $Q_{\rm d}$ the dual feasible set.

Proof. The dual function of (E.2) is given by

$$\bar{g}(t,\kappa) = \begin{cases} \displaystyle\sum_{j\in\mathcal{J}_J}\bar{g}_j(t) - \left(\sum_{\ell\in\mathcal{I}_\mathcal{J}} t_\ell^T y + \delta_\ell\kappa_\ell\right), & \text{if} \quad \|t_\ell\|_2\le\kappa_\ell,\ \forall\ell\in\mathcal{I}_J \\ -\infty, & \text{else} \end{cases}, \tag{E.5}$$

where

$$t = \{t_\ell\}_{\ell\in\mathcal{I}_J}, \quad \kappa = \{\kappa_\ell\}_{\ell\in\mathcal{I}_J},$$

$$\bar{g}_j(t) = \inf_{z_j}\tilde{g}_j(t) = \inf_{z_j}\left\{\lambda_j\|W_j z_j\|_1 + \left(\sum_{\ell\in c_j(\mathcal{I}_J)} t_\ell^T\bar{D}_{\ell,j}\right) z_j\right\},$$

and

$$c_j(\mathcal{I}) = \{\ell\,|\,\ell\in\mathcal{I}, j\in\ell\}\,.$$

Note that the dual function is now decoupled in the functions $g_j(t)$ with the implicit constraint $\|t_\ell\|_2\le\kappa_\ell$, $\forall\ell\in\mathcal{I}_J$. Furthermore,

$$\bar{g}_j(t) = \begin{cases} 0, & \text{if} \quad \|u_j\|_\infty\le\lambda_j,\ u_j = -W^{-1}\displaystyle\sum_{\ell\in c_j(\mathcal{I}_J)}\bar{D}_{\ell,j}^T t_\ell \\ -\infty, & \text{else.} \end{cases}. \tag{E.6}$$

At this point, we change the implicit domain in (E.5) and (E.6), to explicit constraints and note that $\kappa_\ell^* = \|t_\ell^*\|_2$, $\forall\ell\in\mathcal{I}_J$, under maximization and thereby obtain the dual problem (E.4). ■

The equality constraints in (E.4) are simple because the variables t_j, $\forall j\in\mathcal{J}_J$, associated with the side descriptions, only occurs on the left hand side, while the rest of the variables t_ℓ, $\forall\ell\in\mathcal{I}_\mathcal{J}\backslash\mathcal{J}_\mathcal{J}$, associated with the joint description, are on the right side. We can then make a variable substitution of t_j, $\forall j\in\mathcal{J}_J$, in the objective, but we choose the form (E.4) for readability.

4.3 Smoothing

Since the dual problem has simple and non-intersecting constraints it is possible to efficiently apply first-order projection methods. The objective of the dual problem (E.4) is differentiable on $\|t_\ell\|_2 > 0$ and sub-differentiable on $\|t_\ell\|_2 = 0$. The objective in the dual problem (E.4) is hence not smooth.[3] We could then apply an algorithm such as the sub-gradient algorithm with complexity $\mathcal{O}(1/\epsilon^2)$ or form a smooth approximation and apply an optimal first-order method to the smooth problem and obtain complexity $\mathcal{O}(\frac{1}{\epsilon})$, as proposed in [12]. The primal feasible set has intersecting Euclidean norm ball constraints, so we cannot efficiently follow the algorithm [12], since this approach requires projections in both the primal and dual feasible set. We will next show how to adapt the results of [12], in the spirit of [36], using only projection on the dual feasible set and still achieve complexity $\mathcal{O}(\frac{1}{\epsilon})$. Consider

$$\|x\|_2 = \max_{\|v\|_2 \leq 1} \left\{ v^T x \right\} \tag{E.7}$$

and the approximation

$$\Psi_\mu(x) = \max_{\|v\|_2 \leq 1} \left\{ v^T x - \frac{\mu}{2}\|v\|_2^2 \right\} = \begin{cases} \|x\|_2 - \mu/2, & \text{if} \quad \|x\|_2 \geq \mu \\ \frac{1}{2\mu} x^T x, & \text{otherwise} \end{cases}, \tag{E.8}$$

where $\Psi_\mu(\cdot)$ is a Huber function with parameter $\mu \geq 0$. For $\mu = 0$ we have $\Psi_0(x) = \|x\|_2$. The function $\Psi_\mu(x)$ has for $\mu > 0$ the (Lipschitz continuous) derivative

$$\nabla \Psi_\mu(x) = \frac{x}{\max\{\|x\|_2, \mu\}} .$$

The dual objective is

$$g(t) = -\sum_{\ell \in \mathcal{I}_J} \left(\delta_\ell \|t_\ell\|_2 + y^T t_\ell \right)$$

and we can then form a smooth function g_μ as

$$g_\mu(t) = -\sum_{\ell \in \mathcal{I}_J} \left(\delta_\ell \Psi_\mu(t_\ell) + y^T t_\ell \right).$$

The Lipschitz constant of the gradient is $L(\nabla \Psi_\mu(x)) = \frac{1}{\mu}$ and then

$$L_\mu = L(\nabla g_\mu(t)) = \left(\sum_{\ell \in \mathcal{I}_J} \frac{\delta_\ell}{\mu} + 1 \right) = \frac{C}{\mu} + |\mathcal{I}_J| .$$

[3] A smooth function is a function with Lipschitz continuous derivatives [30].

Also, $g(t)$ can bounded as

$$g_\mu(t) \leq g(t) \leq g_\mu(t) + \mu C\,.$$

Now, fix $\mu = \frac{\epsilon}{2C}$ and let the ith iteration $t^{(i)}$ of an algorithm have the property

$$g_\mu^* - g_\mu(t^{(i)}) \leq \frac{\epsilon}{2}\,. \tag{E.9}$$

Then it follows that

$$g^* - g(t^{(i)}) \leq g_\mu^* + \mu C - g_\mu(t^{(i)}) \leq \epsilon\,. \tag{E.10}$$

By using an optimal-first order algorithm for L-smooth problems with complexity $\mathcal{O}\left(\sqrt{\frac{L}{\epsilon}}\right)$ [30], then $t^{(i)}$ can be obtained in

$$\begin{aligned} i \quad &= \mathcal{O}\Big(\sqrt{\tfrac{L_\mu}{\epsilon}}\Big) = \mathcal{O}\Big(\sqrt{\tfrac{1}{\mu\epsilon} + \tfrac{1}{\epsilon}}\Big) = \mathcal{O}\Big(\sqrt{\tfrac{1}{\epsilon^2} + \tfrac{1}{\epsilon}}\Big) \\ &\leq \mathcal{O}\Big(\sqrt{\tfrac{1}{\epsilon^2}} + \sqrt{\tfrac{1}{\epsilon}}\Big) = \mathcal{O}\big(\tfrac{1}{\epsilon}\big) + \mathcal{O}\Big(\sqrt{\tfrac{1}{\epsilon}}\Big) = \mathcal{O}\big(\tfrac{1}{\epsilon}\big) \end{aligned} \tag{E.11}$$

iterations.[4]

4.4 Recovering primal variables from dual variables

The primal variables can be recovered as a minimizer z_j of $\tilde{g}_j(t)$, see [20, §5.5.5]. But since $\|\cdot\|_1$ is not strictly convex there will in general be more than one minimizer. We will instead consider a different approach.

The Karush-Kuhn-Tucker (KKT) optimality conditions for the convex problem (E.2) are

$$\begin{cases} h_2(D_\ell z_\ell^* - y)\|t_\ell^*\|_2 - t_\ell^* \in 0, \quad \forall \ell \in \mathcal{I}_J \\ \kappa_\ell^*(\|D_\ell z_\ell^* - y\|_2 - \delta_\ell) = 0, \quad \forall \ell \in \mathcal{I}_J \qquad (\|t_\ell^*\|_2 = \kappa_\ell^*) \\ \displaystyle\sum_{\ell \in c_j(\mathcal{I}_\mathcal{J})} D_{\ell,j}^T t_\ell^* + W_j u_j^* = 0, \quad \forall j \in \mathcal{J}_J \\ \|D_\ell z_\ell^* - y\|_2 \leq \delta_\ell, \quad \forall \ell \in \mathcal{I}_J \\ \lambda_j h_1(W_j z_j^*) - u_j^* \in 0, \quad \forall j \in \mathcal{J}_J \end{cases} \tag{E.12}$$

with $h_a(x) = \partial\|x\|_a$. We have for $\delta_\ell > 0$, $\forall \ell \in \mathcal{I}_J$,

$$\begin{cases} h_2(D_\ell z_\ell^* - y)\|t_\ell^*\|_2 - t_\ell^* \in 0 \\ \|t_\ell^*\|_2(\|D_\ell z_\ell^* - y\|_2 - \delta_\ell) = 0 \end{cases} \Leftrightarrow t_\ell^* = \frac{\|t_\ell^*\|_2}{\delta_\ell}(D_\ell z_\ell^* - y) \text{ if } \|t_\ell^*\|_2 > 0 \tag{E.13}$$

[4] See *e.g.*, [37] for a definition of the big-O notation .

for all $\ell \in \mathcal{I}_J$. The system

$$\begin{cases} h_2(D_\ell z_\ell^* - y)\|t_\ell^*\|_2 - t_\ell^* \in 0, \quad \forall \ell \in \mathcal{I}_J \\ \kappa_\ell^*(\|D_\ell z_\ell^* - y\|_2 - \delta_\ell) = 0, \quad \forall \ell \in \mathcal{I}_J \qquad (\|t_\ell^*\|_2 = \kappa_\ell^*) \\ \sum_{\ell \in c_j(\mathcal{I}_\mathcal{J})} D_{\ell,j}^T t_\ell^* + W_j u_j^* = 0, \quad \forall j \in \mathcal{J}_J \end{cases} \tag{E.14}$$

is then equivalent to

$$\sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^*\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T D_\ell z_\ell^* = -W_j u_j^* + \sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^*\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T y, \quad \forall j \in \mathcal{J}_J \,.$$

We can then obtain the equivalent KKT optimality conditions

$$\begin{cases} \sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^*\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T D_\ell z_\ell^* = -W_j u_j^* + \sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^*\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T y, \quad \forall j \in \mathcal{J}_J \quad \text{(E.15.}\triangle\text{)} \\ \|D_\ell z_\ell^* - y\|_2 \le \delta_\ell, \quad \forall \ell \in \mathcal{I}_J \\ \lambda_j h_1(W_j z_j^*) - u_j^* \in 0, \quad \forall j \in \mathcal{J}_J \,. \end{cases} \tag{E.15}$$

Let $z^* \in \mathcal{Z}$ be a solution to (E.15) and let $\bar{z} \in \bar{\mathcal{Z}}$ be a solution to (E.15.$\triangle$).

Proposition 4.2. *(Uniqueness) If the solution $\bar{z}$ to the linear system (E.15.$\triangle$) is unique and there exist a solution z^* to problem* (E.1)*, then $z^* = \bar{z}$.*

Proof. We have from the assumption and the system (E.15) that $\emptyset \neq \mathcal{Z} \subseteq \bar{\mathcal{Z}}$. If $|\bar{\mathcal{Z}}| = 1$ then $|\mathcal{Z}| = 1$ such that $\bar{z} = z^*$. ■

Proposition 4.2 explains when it is interesting to solve the primal problem by the dual problem and then recover the primal variables by (E.15.$\triangle$). The reason why we will focus on (E.15.$\triangle$) is that the remaining equations in the system (E.15) are sub-differentiable and feasibility equations. These are difficult to handle - especially for large scale problems. On the other hand, the system (E.15.$\triangle$) is linear in z and can easily be solved for invertible D_j, $\forall j \in \mathcal{J}_J$.

However, first we will analyze the implication of $\kappa_\ell^* = 0$. Let

$$\Omega_J = \{\ell \,|\, \kappa_\ell^* = \|t_\ell^*\|_2 = 0, \ell \in \mathcal{I}_J\},$$

with $\Omega_J \subseteq \mathcal{I}_J$. Then, solving the original primal problem (E.2) is equivalent to solving [20]

$$\begin{aligned} \text{minimize} \quad & \sum_{j \in \mathcal{J}_J} \lambda_j \|W_j z_j\|_1 \\ \text{subject to} \quad & \|D_\ell z_\ell - y\|_2 \le \delta_\ell, \qquad \forall \ell \in \mathcal{I}_J \backslash \Omega_J \,, \end{aligned} \tag{E.16}$$

where we can now remove constraints which are not strongly active. Similarly, if there is an $i \in \mathcal{J}_J$, such that

$$c_i(\mathcal{I}_J) \subseteq \Omega_J \,,$$

then this corresponds to minimization over an unconstrained z_i. Since by definition $\lambda_i > 0$ and $W_i \succ 0$ then $z_i^* = 0$. Solving the original primal problem (E.2) is, hence, equivalent to solving [20]

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{j\in\mathcal{J}_J\setminus i} \lambda_j \|W_j z_j\|_1 & \\ \text{subject to} & \|D_\ell z_\ell - y\|_2 \leq \delta_\ell, & \forall \ell \in \mathcal{I}_J\setminus\Omega_J \\ & z_i = 0\,. & \end{array} \tag{E.17}$$

Definition 4.3. *(Trivial instance) We will call an instance* $\{y, \{\delta_\ell\}_{\ell\in\mathcal{I}_J}, \{D_\ell\}_{\ell\in\mathcal{I}_J}, \{W_j\}_{j\in\mathcal{J}_J}, \{\lambda_j\}_{j\in\mathcal{J}_J}\}$ *of the MDl1C problem a trivial instance if it can be reformulated as an MDl1C problem without coupled constraints.*

The reason why we call these trivial instances is because they do not include coupled constraints and therefore do not include the trade-off normally associated with MD problems. All trivial instances can be solved straightforwardly by a first-order primal method because they do not include any coupled constraints, see [38].

Proposition 4.4. *We have* $z^* = \bar{z}$ *for all non-trivial instances of the standard MDl1C problem with* $J = 2$.

Proof. Let us represent the system (E.15.$\triangle$) by $\tilde{D}_J z^* = \tilde{u}$ and consider the determinant of this system to analyze under which conditions there is a unique solution. By factorizing $\tilde{D}_J$ and using the multiplicative map of determinants $\det(AB) = \det(A)\det(B)$, $\det(A^T) = \det(A)$, $\det(\alpha A) = \alpha^h \det(A)$ for $A \in \mathrm{R}^{h\times h}$ we obtain the determinant

$$\det(\tilde{D}_J) = \left(\prod_{j\in\mathcal{J}_J} \det(D_j)^2\right) \det\left(\left[\sum_{\substack{\ell\in\mathcal{I}_J \\ i,j\in\ell}} \rho_{\ell,j}\rho_{\ell,i}\frac{\kappa_\ell^*}{\delta_\ell}\right]_{i=1,\cdots,J;j=1,\cdots,J}\right)^M, \tag{E.18}$$

for the standard MDl1C problem. For our example with $J = 2$ we have

$$\begin{aligned} \det(\tilde{D}_2) \;=\; \det(D_1)^2\det(D_2)^2\Bigg(&\frac{\kappa^\star_{\{1\}}\kappa^\star_{\{2\}}\rho^2_{\{1\},1}\rho^2_{\{2\},2}}{\delta_{\{1\}}\delta_{\{2\}}} + \frac{\kappa^\star_{\{1\}}\kappa^\star_{\{1,2\}}\rho^2_{\{1\},1}\rho^2_{\{1,2\},2}}{\delta_{\{1\}}\delta_{\{1,2\}}} \\ &+\frac{\kappa^\star_{\{2\}}\kappa^\star_{\{1,2\}}\rho^2_{\{2\},2}\rho^2_{\{1,2\},2}}{\delta_{\{2\}}\delta_{\{1,2\}}}\Bigg)^M. \end{aligned}$$

Since $\rho_{\ell,j} > 0, \forall \ell \in \mathcal{I}_J, j \in \ell$, $\delta_\ell > 0, \forall \ell \in \mathcal{I}_J$ and $\det(D_j) \neq 0, \forall j \in \mathcal{J}_J$, the condition $\det(\tilde{D}_2) = 0$ is determined by which ℓ, $\kappa_\ell^* = 0$. Let

$$O_2 = \{\Omega_2 \,|\, \det(\tilde{D}_2) = 0\}\,.$$

Then O_2 is given as

$$O_2 = \{\{\{1\},\{2\}\},\{\{1\},\{2\},\{1,2\}\},\{\{1\},\{1,2\}\},\{\{2\},\{1,2\}\}\}\,.$$

Let us consider all the cases:

- $\{\{1\},\{2\}\}$. No coupled constraints, which implies a trivial instance.
- $\{\{1\},\{1,2\}\}$ or $\{\{2\},\{1,2\}\}$. Corresponds to $z_1^* = 0$ or $z_2^* = 0$. The primal problem can be solved directly over z_2 or z_1 with no coupled constraints, which implies a trivial instance.
- $\{\{1\},\{2\},\{1,2\}\}$. Corresponds to an instance with $z_1^* = 0$ and $z_2^* = 0$ and no coupled constraints, which implies a trivial instance.

■

Remark (Proposition 4.4): All the cases for $J = 2$ can be seen as two descriptions transmitted as one description over one channel. It is not easy to analyze O_J for $J \geq 3$ in all cases and compare them to the definition of trivial instances. However, we will make the following partial description on the number of active constraints to ensure recovery of optimal primal variables.

Proposition 4.5. *For a standard MDl1C problem, if*

i) all side constraints are strongly active, $\kappa_j^* > 0, \forall j \in \mathcal{J}_J$, then $z^* = \bar{z}$

ii) there are no strongly active constraints $\kappa_\ell^* = 0, \forall \ell \in \mathcal{I}_J$, then $z^* = 0$.

Proof. *i)* From (E.13) we have $t_\ell^* = \frac{\|t_\ell^*\|_2}{\delta_\ell}(D_\ell z_\ell^* - y)$, $\forall \ell \in \mathcal{J}_J$, which gives a unique solution to $z^* = \mathop{\mathrm{C}}_{j\in\mathcal{J}_J} D_j^{-1}\left(t_j^* \frac{\delta_j}{\|t_j^*\|_2} + y\right)$. Since a subsystem ($\mathcal{J}_J \subseteq \mathcal{I}_J$) of (E.15.$\triangle$) has exactly one point, then $|\bar{\mathcal{Z}}| \leq 1$. A standard MDl1C problem is solvable such that $|\bar{\mathcal{Z}}| \geq |\mathcal{Z}| \geq 1$. Hence $|\bar{\mathcal{Z}}| = 1$ and from Proposition 4.2 we then have $z^* = \bar{z}$.

ii) If $\kappa_\ell^* = \|t_\ell^*\|_2 = 0, \forall \ell \in \mathcal{I}_J$, then $g(t^*) = 0$ and $f(z^*) = 0$ according to strong duality. From the definition, $W_j \succ 0, \forall j \in \mathcal{J}_J$ and $\lambda_j > 0, \forall \mathcal{J}_J$, then $f(z^*) = 0 \Leftrightarrow z^* = 0$.

■

Remark (Proposition 4.5): It is always possible to make all the inactive side distortion constraints strongly active by adjusting δ_j, $j \in \mathcal{J}_J$, without significantly changing the original formulation. With this approach we can always recover the primal variables as $z^* = \bar{z}$.

4.5 Stopping Conditions

Since we implement a primal-dual first-order method and the problem has strong duality, a primal-dual stopping criteria is interesting. From the dual iterates $\left(t^{(i)}, u^{(i)}\right)$ we obtain the primal iterate $z^{(i)}$ as the solution to

$$\sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^{(i)}\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T D_\ell z_\ell^{(i)} = -W_j u_j^{(i)} + \sum_{\ell \in c_j(\mathcal{I}_J)} \frac{\|t_\ell^{(i)}\|_2}{\delta_\ell} \bar{D}_{\ell,j}^T y, \quad \forall j \in \mathcal{J}_J \,.$$

We then stop the first-order method at iteration i if

$$f(z^{(i)}) - g(t^{(i)})) \leq \epsilon, \quad z^{(i)} \in Q_{\mathrm{p}}, \left(t^{(i)}, u^{(i)}\right) \in Q_{\mathrm{d}} \,.$$

To ensure scalability in the dimensions of the problem, we select $\epsilon = JM\epsilon_{\mathrm{r}}$, where for example we may choose to solve the problem to medium accuracy, *e.g.*, $\epsilon_{\mathrm{r}} = 10^{-3}$.

5 Analyzing the Sparse Descriptions

In this section we will use an image example and analyze the sparse descriptions. In particular, we show that it is possible to obtain a sparse representation which has a lower cardinality for the same PSNR requirement, or better PSNR for same cardinality, using the MDl1C approach compared to the simple approach of thresholding.

For images, let y be the column major wise stacked version of a two-dimensional image of dimension $m \times n$, $M = mn$. The images are normalized such that $y \in [0; 1]^M$ and the PSNR is

$$\mathrm{PSNR}(\delta) = 10 \log_{10} \left(\frac{1}{\frac{1}{M}\delta^2} \right) .$$

We define $D = \{D_\ell\}_{\ell \in \mathcal{I}_J}$, $\delta = \{\delta_\ell\}_{\ell \in \mathcal{I}_J}$, $\lambda = \{\lambda_j\}_{j \in \mathcal{J}_J}$. We will denote $\bar{z} = \Phi_D(y, \delta, \lambda)$ an ϵ-suboptimal solution of the problem (E.2) with $\bar{z} = \{z_j\}_{j \in \mathcal{J}_J}$ after 4 reweight iterations. Note that in the single channel case $J = 1$ we will obtain the problem (E.1). We will also define the function $\bar{z} = T_{D_j}(y, \delta)$ as the thresholding function of the smallest coefficients of $D_j^{-1} y$ such that $\mathrm{PSNR}(\|D_j T_{D_j}(y, \delta) - y\|_2) \approx \delta$.

We now select the two channel case $J = 2$, y the Pirate standard image (Grayscale, 512×512) and as dictionaries D_1^{-1}: a 2-dimensional orthogonal Symlet16 discrete wavelet transform (DWT) with 7 levels, and D_2^{-1}: a 2-dimensional biorthogonal CDF 9/7 DWT with 7 levels. The results are reported in Table E.1, where we for clarity will refer to different approaches using

the numbering (1)-(4). First, we obtain $\bar{z} = \Phi_D(y, \delta, \lambda)$ (1) and then apply thresholding to the same signal such that the side PSNRs are the same (2). Notice that due to the independent thresholding, the refinement $\ell = \{1, 2\}$ is not much better than the individual descriptions. Considering the same setup, but where we select $\tilde{\delta}_j$ such that the cardinality of each description is the same $\text{card}((\Phi_D(y, \delta, \lambda))_j) = \text{card}(T_{D_j}(y, \tilde{\delta}_j))$ (3). In this case, we have a better side PSNR, but the refinement is still poor and the central distortion not as good as in the case of the MDl1C approach. Finally, if we performed thresholding to achieve the same PSNR on the side distortion as the central distortion (4) we see that we need an excessive cardinality. We conclude that the MDl1C framework is able to generate non-trivial descriptions in a MD framework with respect to both the cardinality of the descriptions and the refinement.

ID	Method	$\text{card}(\bar{z}_j)/N$		$\text{PSNR}(\|D_\ell \bar{z}_\ell - y\|_2)$		
		$j = 1$	$j = 2$	$\ell = \{1\}$	$\ell = \{2\}$	$\ell = \{1, 2\}$
(1)	$\bar{z} = \Phi_D(y, \delta, \lambda)$	0.058	0.059	27.0	27.0	33.1
(2)	$\bar{z}_j = T_{D_j}(y, \delta_j)$	0.030	0.026	27.0	27.0	27.9
(3)	$\bar{z}_j = T_{D_j}(y, \tilde{\delta}_j)$	0.058	0.059	29.3	30.0	30.8
(4)	$\bar{z}_j = T_{D_j}(y, \delta_{\{1,2\}})$	0.128	0.112	33.1	33.1	34.4

Table E.1: Comparison between MDl1C (1), thresholding to same side PSNR (2) thresholding to same cardinality (3) and thresholding to achieve the central distortion on each side channel (4).

With the same setup as used in Table E.1, we also investigate the first-order iteration complexity for obtaining a solution to the MDl1C problem, including 4 reweight iterations. Each reweight iteration has the worst-case iteration complexity $\mathcal{O}\big(\frac{1}{\epsilon}\big)$ given in (E.11) which results in an overall complexity of $\mathcal{O}\big(\frac{1}{\epsilon}\big)$. The results are shown in Fig. E.4. In general, we obtain an empirical complexity slightly (but not significantly) better than the theoretical worst-case iteration complexity $\mathcal{O}\big(\frac{1}{\epsilon}\big)$. For obtaining an $\epsilon = \epsilon_{\text{r}} JM$-suboptimal solution with $\epsilon_{\text{r}} = 10^{-3}$ and 4 reweight iterations requires approximately 700 first-order iterations.

6 Simulation and Encoding of Sparse Descriptions

In this section we will consider an application of the proposed scheme, where the sparse descriptions are encoded to adapt the sparse MD framework to a rate-distortion MD framework.

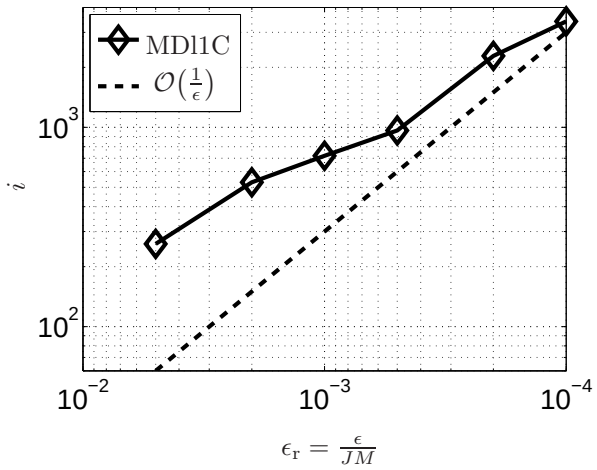


Fig. E.4: Number of first-order iterations i including 4 reweight iterations as a function of of the relative accuracy ϵ_r. We also plot the complexity function $\mathcal{O}(\frac{1}{\epsilon})$ for comparison.

A state-of-the-art method for encoding images is the SPIHT encoder [13], which efficiently uses the tree structure of the DWT. We then denote the encoding of the coefficients $\bar{z} = D^{-1}y$ as $\hat{z} = \Gamma_D(\bar{z}, y, \delta)$, such that $\|D\hat{z} - y\|_2 \leq \delta$. The encoder applies baseline SPIHT encoding. We add half a quantization level to all the significant wavelets transform coefficients when the stopping criteria $\|D\hat{z} - y\|_2 \leq \delta$ is evaluated, see [39] or [40, Chapt. 6]. The quantized non-zero coefficients and their locations are further entropy coded using the arithmetic coder [41].

In order to illustrate the behaviour of encoding we will also obtain coefficients from solving a series of reweighted l_1-compression problems

$$\begin{array}{ll} \text{minimize} & \|Wz\|_1 \\ \text{subject to} & \|Dz - y\|_2 \leq \delta\,. \end{array}$$

An ϵ-optimal solution from the above problem is denoted $\bar{z} = \Phi_D(y, \delta, 1)$ (the same notation in the case of $J = 1$).

The encoder can be used in two different ways $\hat{z} = \Gamma_D(D^{-1}y, y, \delta)$ or $\hat{z} = \Gamma_D(\Phi_D(y, \delta - \gamma, 1), y, \delta)$. When encoding the coefficients $\bar{z} = \Phi_D(y, \delta - \gamma, 1)$, this corresponds to letting SPIHT encode the (reconstructed) image $D\bar{z} \approx y$. Note that we have introduced a modification of the distortion requirement with the parameter γ. This is done because $\bar{z} = \Phi_D(y, \delta, 1)$ is on the boundary of the ball $\|D\bar{z} - y\|_2 = \delta$ unless $\bar{z} = 0$. However, quantization slightly degrades the reconstruction quality and it is therefore difficult to ensures $\|D\hat{z} - y\|_2 \leq \delta$

without requiring that the input $\bar{z}$ to the encoder ensure $\|D\bar{z} - y\|_2 \leq \delta - \gamma$, $\gamma > 0$. We use $\gamma = 0.05\delta$. It is possible to use any encoder which is based on encoding the coefficients associated to a linear reconstruction function and SPIHT is such an encoder.

In Fig. E.5 we illustrate the results from encoding coefficients obtained as $\hat{z} = \Gamma_D(D^{-1}y, y, \delta)$ or $\hat{z} = \Gamma_D(\Phi_D(y, \delta - \gamma, 1), y, \delta)$. As test images we use Lena and Boat (Grayscale, 512×512) and we select D^{-1} as a Symlet16 pyramid wavelet transform with 7 levels.

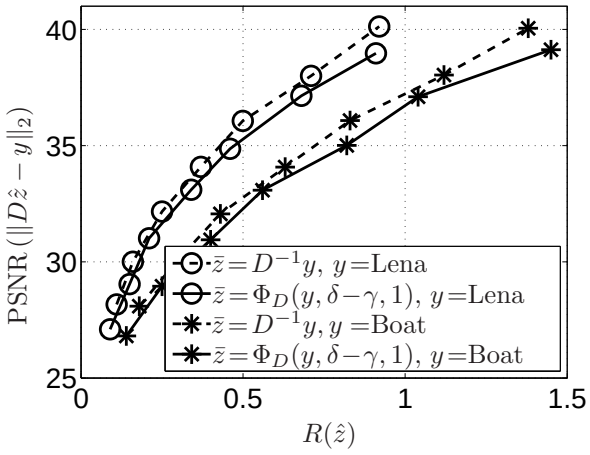
For the simulations we choose different δ's and report the PSNR, the corresponding cardinality and the rate $R(\hat{z})$ [$^{\text{bits}}/_{\text{pixel}}$]. From Fig. E.5(a) we can see that for the same rate the reconstruction using the l_1-minimization approach Φ_D shows a smaller PSNR than with the standard approach. This is to be expected, since for an orthogonal transform, SPIHT is designed to minimize the Euclidean distortion $\|\hat{z} - \bar{z}\|_2$ to the input vector $\bar{z} = D^{-1}y$ [13], and $\|\hat{z} - \bar{z}\|_2$ is also our quality criteria. For $\bar{z} = \Phi_D(y, \delta - \gamma, 1)$, which implies $\bar{z} \neq D^{-1}y$ in general, the design argumentation is slightly distorted by the modified input but the quality criteria remains the same. Further, by first forming a sparse coefficient vector using convex relaxation technique and later encode is suboptimal which further add to the loss. We also notice from Fig. E.5(b) that the cardinality and bit rate behaves linearly in this setup for the l_1-minimization approach as also observed in other sparse decompositions, see *e.g.*, [24].

6.1 Encoding for Multiple Descriptions

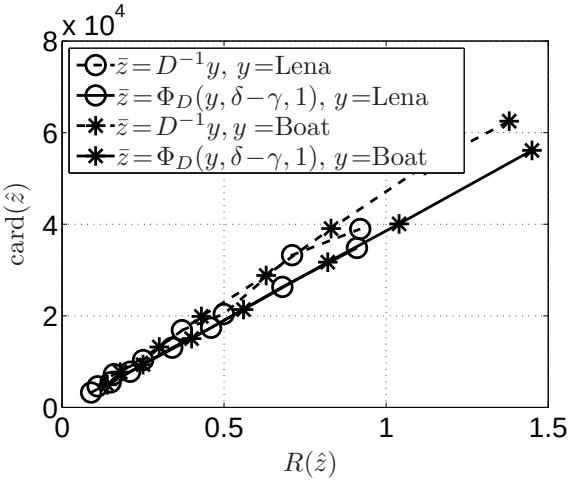
We use the following approach when applying encoding for multiple descriptions, shown in Fig. E.6. First the sparse coefficients vectors are formed using the function $\Phi_D(y, \delta - \gamma, \lambda)$ with $\gamma = \{\gamma_j\}_{j \in \mathcal{J}_J}$ and $\gamma_j = 0.05 \min_{i \in \mathcal{J}_J} \delta_i,\ \forall j \in \mathcal{J}_J$. That is, we aim for at least a 5% better reconstruction in the optimization stage which we use to allow for a loss in the encoding stage. Each description is independently coded using $\Gamma_{D_j}(D_j^{-1}\bar{z}_j, y, \delta_j - \tilde{\gamma}_j)$ to generate the encoded description vectors $\hat{z}_j$ with the rate $R(\hat{z}_j)$. The parameter $\tilde{\gamma}_j$ will be discussed shortly. From the received descriptions ℓ, we then apply the decoding function $g_\ell(\hat{z}_\ell) = D_\ell \hat{z}_\ell$.

For the encoding function Γ_{D_j} it is easy to ensure $\|D_j\hat{z}_j - y\|_2 \leq \delta_j$, $\forall j \in \mathcal{J}_J$, since we can encode each description independently until the side distortion is satisfied as we previously did in the example with $J = 1$. It is, however, more complicated to ensure that the coupled constraints are satisfied without requiring an excessive large rate. Setting $R(\hat{z}_j)$ large, we can always satisfy the distortion requirement, but by locating more appropriate points on the rate-distortion function of $\hat{z}_j$, we can achieve a better rate-distortion trade-off. To handle this problem we adjust $\tilde{\gamma}_j$ by applying the following algorithm:

- Encode the descriptions and independently ensure that $\|D_j\hat{z}_j - y\|_2 \leq$



(a)



(b)

Fig. E.5: Encoding of the images Lena and Boat, (Grayscale, 512×512) using $\hat{z} = \Gamma_D(\bar{z}, y, \delta)$ with different distortion requirement δ. In (a) PSNR versus rate and (b) cardinality versus rate.

δ_j, $\forall j \in \mathcal{J}_J$.

- Check all the coupled distortion requirement $\|D_\ell \hat{z}_\ell - y\|_2 \leq \delta_\ell$, $\forall \ell \in \mathcal{I}_J / \mathcal{J}_J$.
  - For the first distortion requirement not fulfilled, check the first-order approximation of the slope $\mathrm{PSNR}(\|D_j \hat{z}_j - y\|_2)/R(\hat{z}_j)$ for $j \in \ell$. Increase the rate (decrease $\tilde{\gamma}_j$) of the description j with the largest slope using bisection.
  - If all distortion requirements of the encoded descriptions are satisfied then decrease the rate (increase $\tilde{\gamma}_j$) for the description $j \in \mathcal{J}_J$ with the smallest slope $\mathrm{PSNR}(\|D_j \hat{z}_j - y\|_2)/R(\hat{z}_j)$ using bisection.

The purpose of the above algorithm is to find a stable point of the Lagrange rate-distortion function. The process of adjusting the description rate using the description with highest or smallest slope respectively is also applied in [42]. The difference is that the above algorithm target distortion constraint (feasibility $\hat{z} \in Q_p$) while [42] targets a rate constraint.

6.2 MD Image Encoding

We perform comparison with state-of-the-art algorithms, specifically MDLT-PC [3, 43] and RD-MDC [42, 44] for the two channel case $J = 2$ and y the Pirate image. We adjust the quantization levels such that we achieve a fixed rate and plot the (average) side and central distortion of the schemes under comparison. We also plot the single description performance of Γ_{D_1} and Γ_{D_2} as the distortion obtained at full rate and at half the rate and associate this to

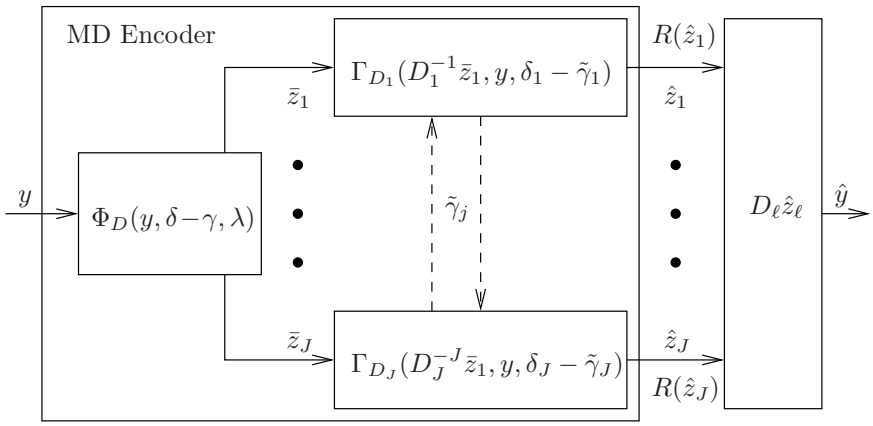


Fig. E.6: Encoding of sparse coefficients in a MD setup.

the central (horizontal line) and side distortion (vertical line), respectively. This corresponds to the extreme MD setup where there is no requirement for the side or central distortion, in which case a single description encoder is sufficient.

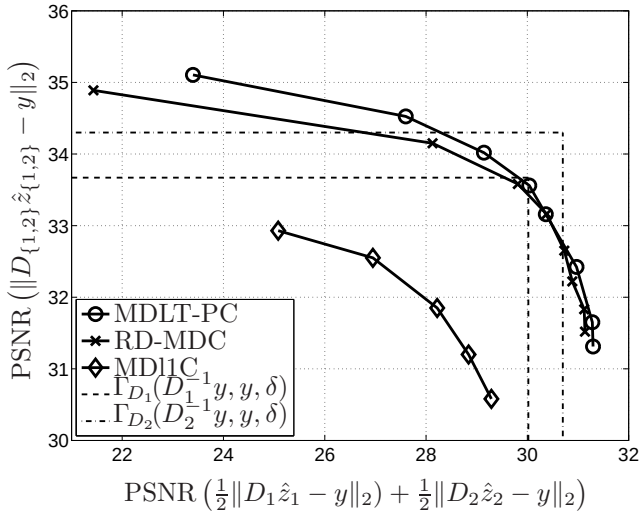
In Fig. E.7 (a) and (b) we see that MDLT-PC and RD-MDC are able to obtain better single description PSNR than that of the SPIHT encoder using either Γ_{D_1} or Γ_{D_2}. Further, we observe that MDl1C shows larger distortion at same rate, but behaves according to the single description bounds of the SPIHT encoder (Γ_{D_1} and Γ_{D_2}), however with a gap. This gap is due to the suboptimality of the MDl1C approach exemplified in Fig. E.5.

We also show an example which demonstrates the flexibility of the proposed method. To this end we select $J = 3$, apply non-symmetric distortion requirement $\delta_\ell \neq \delta_{\ell'}$ for at least some $|\ell| = |\ell'|$ with $\ell, \ell' \in \mathcal{I}_J$, non-symmetric weights $\lambda_j \neq \lambda_{j'}$ for at least some $j, j' \in \mathcal{J}_J$ and both orthogonal and biorthogonal dictionaries. The results are shown in Fig. E.8 where we obtain rates $R(\hat{z}_1) = 0.32$, $R(\hat{z}_2) = 0.52$ and $R(\hat{z}_3) = 0.58$ such that $R(\hat{z}) = 1.42$. For comparison, if we encode the same image in a single description setup with the coder Γ_D using a biorthogonal Cohen-Daubechies-Feauveau (CDF) 9/7 DWT with 7 levels as dictionary D we obtain the distortion measure $\mathrm{PSNR}(\|D\hat{z}'-y\|_2) \approx 27.0$ or $\mathrm{PSNR}(\|D\hat{z}' - y\|_2) \approx 34.0$ at the rates $R(\hat{z}') = 0.25$ or $R(\hat{z}') = 0.81$, respectively. This example is a large scale problem with $M \times (|\mathcal{I}_J| + J + J) \approx 3.4 \cdot 10^6$ primal-dual variables. The encoding process required $v = 13$ iterations and the SPIHT encoder was then applied $J+(v-1) = 15$ times since in the first iteration we need to encode all J descriptions and in the remaining iterations it is only neccessary to encode the single description j for which $\tilde{\gamma}_j$ was modified in the previous iteration.

6.3 MD Image Sequence Encoding

To show the flexibility of the proposed framework, we give an image sequence example. An image sequence can be seen as a three dimensional signal. If we join k consequent frames in a single block we obtain a windowed three dimensional signal with dimension $m \times n \times k$ with $m \times n$ being the frame dimensions of the video. For image sequences we then form y as a column major wise stacked version of a each two-dimension frame with $M = mnk$. For encoding, we apply 3D SPIHT [45]. To comply to the 3D SPIHT framework we also form the dictionaries as three dimensional DWTs. As dictionaries we select $D^{-1}_{\{1\}}$: a 3 level orthogonal Haar DWT along the (temporal) dimension associated with k and a 2-dimensional orthogonal Symlet16 DWT with 5 levels along the (spatial) dimensions associated with m, n, and $D^{-1}_{\{2\}}$: a 3 level orthogonal Haar DWT along the (temporal) dimension associated with k and a 2-dimensional orthogonal Daub8 DWT with 5 levels along the (spatial) dimensions associated with m, n.

(a) $R(\hat{z}) = 1$



(b) $R(\hat{z}) = 0.25$

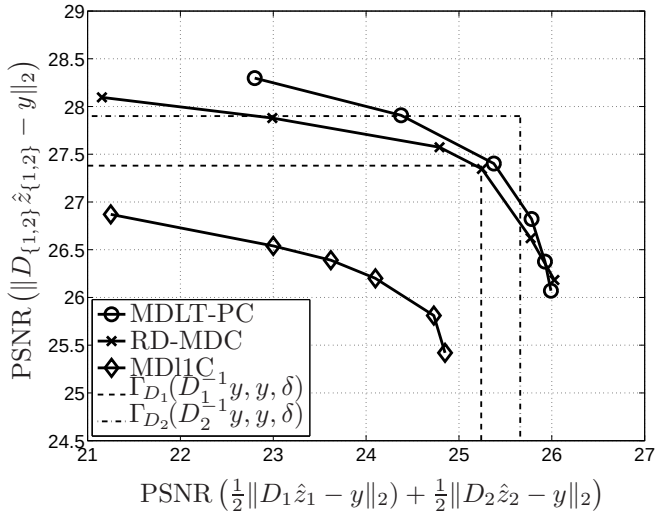


Fig. E.7: Comparison between different MD methods for the image Pirate (Grayscale, 512×512) at: (a) rate $R(\hat{z}) = 1$ and (b) rate $R(\hat{z}) = 0.25$. The plot shows central distortion versus (average) side distortion. The single description performance using Γ_{D_1} and Γ_{D_2} are also shown as the distortion at full rate and at half the rate which are associated to respectively the central distortion (horizontal lines) and side distortion (vertical lines). As dictionaries we use $D_{\{1\}}^{-1}$: a 2-dimensional orthogonal Symlet16 DWT with 7 levels, and $D_{\{2\}}^{-1}$: a 2-dimensional biorthogonal CDF 9/7 DWT with 7 levels.

Fig. E.8: Encoding the image Barbara (Grayscale, 512×512). As dictionaries we use $D_{\{1\}}^{-1}$: a 2-dimensional orthogonal Symlet8 DWT with 7 levels, $D_{\{2\}}^{-1}$: a 2-dimensional orthogonal Symlet16 DWT with 7 levels, and $D_{\{3\}}^{-1}$: a 2-dimensional biorthogonal CDF 9/7 DWT with 7 levels. We have $\lambda_1 = 1.5, \lambda_2 = 1.4$ and $\lambda_3 = 1.0$. The distortion requirements and actual distortions are given above the individual images using the notation $(\mathrm{PSNR}(\|D_\ell \hat{z}_\ell - y\|_2), \mathrm{PSNR}(\delta_\ell))$. The resulting rates are respectively $R(\hat{z}_1) = 0.32$, $R(\hat{z}_2) = 0.52$, $R(\hat{z}_3) = 0.58$ such that $R(\hat{z}) = 1.42$.

Fig. E.9 shows an example for an image sequence where we have defined a frame extraction function $s(y, \bar{k})$ which takes the $\bar{k}$th frame from the image sequence stacked in y. For this example we obtain the rates $R(\hat{z}_1) = 0.10$ and $R(\hat{z}_2) = 0.12$ such that $R(\hat{z}) = 0.22$. For comparison, if we had encoded the same image with the coder Γ_D using $D_{\{2\}} = D$ as dictionary we obtain the distortion measure $\mathrm{PSNR}(\|D\hat{z}' - y\|_2) \approx 29.3$ or $\mathrm{PSNR}(\|D\hat{z}' - y\|_2) \approx 34.0$ at the rates $R(\hat{z}') = 0.05$ or $R(\hat{z}') = 0.17$, respectively. This example is a large scale problem with $M \times (|\mathcal{I}_J| + J + J) \approx 5.7 \cdot 10^6$ primal-dual variables. It is expected that the comparison between the MDl1C method and state-of-the-art MD video coder will render similar results as in Fig. E.7, that is, overall determined by the singe channel encoder and a gap introduced by the suboptimal approach of forming sparse descriptions using convex relaxation. Comparison between 3D SPIHT and standard video encoding schemes are given in, *e.g.*, [45, 46].

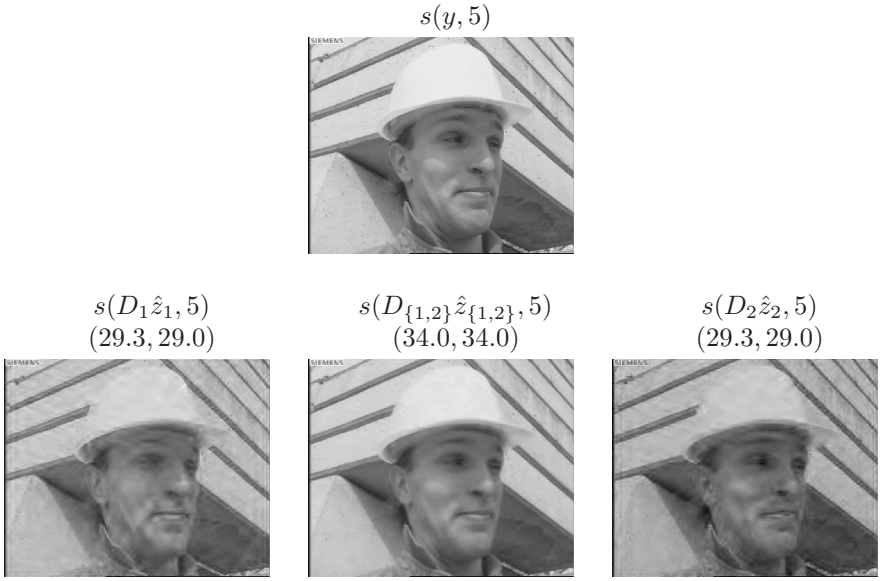


Fig. E.9: Encoding the image sequence Foreman (Grayscale, 288×352). We jointly process $k = 8$ consecutive frames and let $\lambda_1 = \lambda_2 = 1.0$. The dictionaries are $D_{\{1\}}^{-1}$: a 3 level orthogonal Haar DWT along the dimension associated with k and a 2-dimensional orthogonal Symlet16 DWT with 5 levels along the dimensions associated with m, n, and $D_{\{2\}}^{-1}$: a 3 level orthogonal Haar DWT along the dimension associated with k and a 2-dimensional orthogonal Daub8 DWT with 5 levels along the dimensions associated with m, n The distortion requirement and actual distortion are given above the individual images using the notation $(\mathrm{PSNR}(\|D_\ell\hat{z}_\ell - y\|_2), \mathrm{PSNR}(\delta_\ell)))$. The resulting rates are respectively $R(\hat{z}_1) = 0.10$ and $R(\hat{z}_2) = 0.12$ such that $R(\hat{z}) = 0.22$.

7 Conclusion

We have shown how to use efficient first-order convex optimization techniques in a multiple description framework in order to form sparse descriptions, which satisfies a set of individual and joint distortion constraints. The proposed convex formulation allows for non-symmetric distortions, non-symmetric l_1-measures, different dictionaries and an arbitrary number of descriptions. We analyzed the sparse descriptions and concluded that the sparse descriptions were non-trivial. When encoding the sparse coefficients and comparing with state-of-the-art methods it was not possible to achieve the same rate-distortion performance. On the other hand, the proposed method allow for a more flexible formulation and provides an algorithm for applying encoding in sparse signal processing. Efficient encoding of sparse coefficients is generally an open research topic.

References

[1] A. A. E. Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inf. Theory*, vol. 28, no. 6, pp. 851 – 857, Nov. 1982.

[2] L. Ozarow, "On a source-coding problem with two channels and three receivers," *Bell System Technical Journal*, vol. 59, pp. 1909 – 1921, Dec. 1980.

[3] U. G. Sun, J. Liang, C. Tian, C. Tu, and T. Tran, "Multiple desciption coding with prediction compensation," *IEEE Trans. Image Process*, vol. 18, no. 5, pp. 1037–1047, May 2009.

[4] H. Chan and C. Huang, "Multiple description and matching pursuit coding for video transmission over the internet," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Hong Kong, Apr. 2003, pp. 425–428.

[5] T. Nguyen and A. Zakhor, "Matching pursuits based multiple description video coding for lossy environments," in *Proc. Int. Conf. of Image Process. (ICIP)*, Sep. 2003, pp. 57–60.

[6] T. Petrisor, B. Pesquet-Popescu, and J.-C. Pesquet, "A compressed sensing approach to frame-based multiple description coding," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Honolulu, Hawaii, Apr. 2007, pp. 709–712.

[7] Y. Zhang, S. Mei, Q. Chen, and Z. Chen, "A multiple description image/video coding method by compressed sensing theory," in *IEEE Int. Symp. on Circuits and Systems (ISCAS)*, Seattle, Washington, May. 2008, pp. 1830–1833.

[8] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[9] P. Boufounos and R. Baraniuk, "1-bit compressive sensing," in *Proc. 42nd annual Conf. on Inform. Sci. and Syst. (CISS)*, Princeton, New Jersey, Mar. 2008, pp. 16 – 21.

[10] J. Z. Sun and V. K. Goyal, "Optimal quantization of random measurements in compressed sensing," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Seoul, Korea, Jun.-Jul 2009, pp. 6–10.

[11] A. Zymnis, S. Boyd, and E. Candés, "Compressed sensing with quantized measurements," *Signal Processing Letters*, vol. 17, no. 2, pp. 149–152, Feb. 2010.

[12] Y. Nesterov, "Smooth minimization of nonsmooth functions," *Math. Prog. Series A*, vol. 103, pp. 127–152, 2005.

[13] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[14] D. Donoho, "Compressed sensing," *IEEE Trans. on Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[15] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21 – 30, Mar. 2008.

[16] E. Candès, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *The Journal of Fourier Analysis and Applications, Special Issue on Sparsity*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[17] M. Davies and R. Gribonval, "Restricted isometry constants where lp sparse recovery can fail for $0<p\leq 1$," *IEEE Trans. on Inf. Theory*, vol. 55, no. 5, pp. 2203–2214, May 2009.

[18] T. L. Jensen, J. Dahl, J. Østergaard, and S. H. Jensen, "A first-order method for the multiple-description l1-compression problem," in *Proc. of Signal Processing with Adaptive Sparse Structured Representations (SPARS'09), Saint-Malo, France*, Apr. 2009, Available at: http://hal.inria.fr/inria-00369483/PDF/77.pdf.

[19] T. L. Jensen, J. Østergaard, J. Dahl, and S. H. Jensen, "Multiple descriptions using sparse decompositions," in *Proc. European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, Aug. 2010, pp. 110–114.

[20] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[21] J. Chen, C. Tian, T. Berger, and S. S. Hemami, "Multiple description quantization via Gram-Schmidt orthogonalization," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5197 – 5217, Dec. 2006.

[22] J. Østergaard and R. Zamir, "Multiple description coding by dithered delta-sigma quantization," *IEEE Trans. Inf. Theory*, vol. 55, no. 10, pp. 4661–4675, Oct. 2009.

[23] Y. Kochman, J. Østergaard, and R. Zamir, "Noise-shaped predictive coding for multiple descriptions of a colored Gaussian source," in *Proc. IEEE Data Comp. Conf. (DCC)*, Snowbird, Utah, Mar. 2008, pp. 362 – 371.

[24] H. Zhihai and S. K. Mitra, "A linear source model and a unified rate control algorithm for DCT video coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 970–982, Nov. 2002.

[25] S. Diggavi, N. Sloane, and V. Vaishampayan, "Asymmetric multiple description lattice vector quantizers," *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 174 – 191, Jan. 2002.

[26] J. Østergaard, R. Heusdens, and J. Jensen, "n-channel asymmetric entropy-constrained multiple-description lattice vector quantization," *IEEE Trans. Inf. Theory*, vol. 56, no. 12, pp. 6354–6375, Dec. 2010.

[27] S. J. Wright, *Primal-Dual Interior-Point Methods*. SIAM, 1997.

[28] J. Barzilai and J. Borwein, "Two point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, Jan. 1988.

[29] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.

[30] Y. Nesterov, *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

[31] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[32] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imag. Sci.*, vol. 4, no. 1, pp. 1–39, 2011.

[33] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. on Imaging Sciences*, vol. 2, pp. 183–202, Mar. 2009.

[34] J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Algo.*, vol. 53, no. 1, pp. 67–92, Jan. 2010.

[35] G. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101 – 111, Jan.-Feb. 1960.

[36] L. Vandenberghe, "Optimization methods for large-scale systems," Lecture Notes, 2009.

[37] R. Graham, D. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 1994.

[38] J. Dahl, J. Østergaard, T. L. Jensen, and S. H. Jensen, "An efficient first-order method for ℓ_1 compression of images," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, Taipei, Taiwan, Apr. 2009, pp. 1009–1012.

[39] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2009.

[40] E. K. Rao and P. Yip, *The Transform and Data Compression Handbook*. CRC Press, 2001.

[41] A. Said, "FastAC: Arithmetic coding implementation," Public available software for arithmetic coding, 2004.

[42] T. Tillo, M. Grangetto, and G. Olmo, "Multiple description image coding based on lagrangian rate allocation," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 673–683, Mar. 2007.

[43] MDLT-PC, Source code, Available at: http://www.ensc.sfu.ca/~jiel/MDLTPC.html.

[44] RD-MDC, Source code, Available at: http://www.telematica.polito.it/sas-ipl/.

[45] B.-J. Kim and W. A. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," in *Proc. IEEE Data Comp. Conf. (DCC)*, Snowbird, Utah, Mar. 1997, pp. 251–260.

[46] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical tress (3-D SPIHT)," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, pp. 1374–1387, Dec. 2000.