Tomographic Image Reconstruction Using Training Images with Matrix and Tensor Formulations

Soltani, Sara; Hansen, Per Christian; Andersen, Martin Skovgaard

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Soltani, S., Hansen, P. C., & Andersen, M. S. (2015). Tomographic Image Reconstruction Using Training Images with Matrix and Tensor Formulations. Kgs. Lyngby: Technical University of Denmark (DTU). (DTU Compute PHD-2015; No. 387).

Tomographic Image Reconstruction Using Training Images

with Matrix and Tensor Formulations

Sara Soltani

Kongens Lyngby 2015

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

"In the sunset of dissolution, everything is illuminated by the aura of nostalgia."
–*Milan Kundera,* The Unbearable Lightness Of Being

In memory of
Alireza 1983-2006
and
Razieh 1959-2015

Summary (English)

Reducing X-ray exposure while maintaining the image quality is a major challenge in computed tomography (CT); since the imperfect data produced from the few view and/or low intensity projections results in low-quality images that are suffering from severe artifacts when using conventional reconstruction methods. Incorporating a priori information about the solution is a necessity to improve the reconstruction. For example, Total Variation (TV) regularization method – assuming a piecewise constant image model – has been shown to allow reducing X-ray exposure significantly, while maintaining the image resolution compared to a classical reconstruction method such as Filtered Back Projection (FBP).

Some priors for the tomographic reconstruction take the form of cross-section images of similar objects, providing a set of the so-called training images, that hold the key to the structural information about the solution. The training images must be reliable and application-specific. This PhD project aims at providing a mathematical and computational framework for the use of training sets as non-parametric priors for the solution in tomographic image reconstruction. Through an unsupervised machine learning technique (here, the dictionary learning), prototype elements from the training images are extracted and then incorporated in the tomographic reconstruction problem both with matrix and tensor representations of the training images.

First, an algorithm for the tomographic image reconstruction using training images, where the training images are represented as vectors in a training matrix, is described. The dictionary learning problem is formulated as a regularized non-negative matrix factorization in order to compute a nonnegative dictionary. Then a tomographic solution with a sparse representation in the dictionary is obtained through a convex optimization formulation. Computational experiments clarify the choice and interplay of the model parameters and the regularization parameters. Furthermore, the assumptions in the tomographic problem formulation are analyzed. The sensitivity and robustness of the reconstruction to variations of the scale and rotation in the training images is investigated and algorithms to estimate the correct relative scale and orientation of the unknown image to the training images are suggested.

Then, a third-order tensor representation for the training images images is used. The dictionary and image reconstruction problem are reformulated using the tensor representation. The dictionary learning problem is presented as a nonnegative tensor factorization problem with sparsity constraints and the reconstruction problem is formulated in a convex optimization framework by looking for a solution with a sparse representation in the tensor dictionary. Numerical results show considering a tensor formulation over a matrix formulation significantly reduces the approximation error by the dictionary as well as leads to very sparse representations of both the training images and the reconstructions.

Further computational experiments show that in few-projection and low-dose settings our algorithm is while (not surprisingly) being superior to the classical reconstruction methods, is competitive with (or even better of) the TV regularization and tends to include more texture and sharper edges in the reconstructed images.

The focus of the thesis is the study of mathematical and algorithmic prospectives and thus the training images and tomographic scenarios are mostly simulation based. More studies are however needed for implementing the proposed algorithm in a routine use for clinical applications and materials testing.

Summary (Danish)

I forbindelse med brugen af “computed tomography” eller computer-tomografi (CT) er det en stor udfordring af opnå høj billedkvalitet når man reducerer mængden af Røntgenstråling, idet de traditionelle rekonstruktionsalgoritmer giver billeder af lav kvalitet når man har få eller støjfyldte data. Det er nødvendigt at udnytte yderligere viden om objektet for at kunne beregne en pålidelig rekonstruktion. Et eksempel på en metode der tillader dette er TV (“total variation”) som beregner stykkevis konstante rekonstruktioner - denne metoder tillader at man reducerer Røntgendosen betydeligt. Som a priori viden for tomografisk rekonstruktion kan man i visse tilfælde bruge træningsbilleder, dvs. tværsnitsbilleder af objekter af samme type som dét der måles på og som indeholder information om objektets struktur. Træningsbillederne skal være pålidelige og specifikke for objektet. Målet med dette ph.d.-projekt er at give en matematisk og beregningsvenlig formulering af, hvorledes træningsbilleder bruges om ikke-parametrisk prior for tomografisk billedrekonstruktion. Ved hjælp af teknikker fra “unsupervised machine learning” (specifikt, “dictionary learning”) udtrækkes prototype-elementer fra træningsbillederne således at de kan bruges i rekonstruktionen; der betrages både matrix- og tensor-formuleringer af dette problem. Først beskrives en algoritme til tomografisk rekonstruktion baseret på træningsbilleder, i hvilken billederne er repræsenteret som vektorer i en trænings-matrix. “Dictionary learning” problemet formuleres hér som en regulariseret ikke-negativ matrix-faktorisering med det formål at beregne et ikke-negativt “dictionary”. Derefter beregnes en løsning med sparse repræsentation i dette “dictionary” vha. en konveks optimerings-formulering. Computereksperimenter klarlægger valget og sammenhængen af model- og regulariseringsparametrene samt betydningen af de valgte antagelser. Endvidere undersøges følsomheden over for variationer i træningsbilledernes geometriske skalering og rotation, og der beskrives metoder til at bestemme disse parametre i træningsbillederne. Dernæst beskrives en tilsvarende algoritme hvor træningsbillederne repræsenteres i en tensor, som benyttes i både “dictionary”- og rekonstruktionsproblemet. “Dictionary learning” formuleres nu som en ikke-negativ tensorfaktorisering med sparsitet, mens rekonstruktionsproblemet formulerings som et konvekst optimeringsproblem hvor løsningen har en sparse repræsentation i tensor-“dictionary”. Computer-eksperimenter viser at brugen af tensorer reducerer approximationsfejlen og giver mere sparse repræsentationer af træningsbillederne og rekonstruktionen, sammenlignet med den først algoritme. Yderligere computer-eksperimenter viser at i måle-situationer med få data eller lav Røntgenstråling er de nye algoritmer bedre end de klassiske metoder, og de er konkurrencedygtige med TV-regularisering idet de kan give billeder med mere tekstur og skarpere kanter. Fokus i denne afhandling er studiet af de matematiske og beregningsmæssige aspekter, og arbejdet er derfor baseret på computersimuleringer. Fremadrettet skal metoderne testes på konkrete anvendelser, fx inden for materialevidenskab.

Preface

This thesis was prepared at DTU Compute in fulfillment of the requirements for acquiring a PhD degree at the Technical University of Denmark (DTU). The work was carried out between September 2012 and August 2015 in the Section for Scientific Computing, Department of Applied Mathematics and Computer Science (formerly Department of Informatics and Mathematical Modeling), DTU, under supervision of Professor Per Christian Hansen and Assistant Professor Martin S. Andersen. Some part of the work was done during two research stays — in total one month — in 2014 at the Department of Mathematics, Tufts University, MA, USA, under supervision of Professor Misha E. Kilmer.

This work was part of the project HD-Tomo funded by Advanced Grant No. 291405 from the European Research Council.

Lyngby, 31-August-2015

Sara Soltani

Sara Soltani

Acknowledgements

I would like to thank my supervisor Per Christian Hansen for introducing me to inverse problems and tomography and for teaching me the true meaning of research, many hours of encouraging discussions, and many comments and suggestions to improve the work. I want to also thank my co-supervisor Martin Skovgaard Andersen for sharing his knowledge in various aspects of optimization and for his contributions to this work. I am also grateful to Misha Elena Kilmer at Tufts University for sharing her knowledge on tensors and many inspiring insights on applications of tensor formulations in imaging problems.

I would also like to thank Anders Bjorholm Dahl, Samuli Siltanen and Sabine Van Huffel for accepting to be part of the evaluation committee. I would also like to express my gratitude to Yiqiu Dong, Jürgen Frikel and Kim Knudsen for their helpful insights and comments along the way. I am very thankful to Jakob Sauer Jørgensen, Federica Sciacchitano, Mikhail Romanov and my roomate Sazuan Nazrah Mohd. Azam for friendly conversations and the great times. A special thank to all the members (and former members) of the HD-tomo project and Scientific Computing Section, who have been both great friends and colleagues. It was a pleasure to attend the regular HD-tomo meetings and listen to great talks and learn a lot from all of my colleagues and visiting scholars. I am thankful to the Department of Mathematics at Tufts University for hosting me during my two short visits in 2014. I am grateful to the IT support team at DTU compute for providing and maintaining amazing IT facilities and for their immediate help when it was needed.

I want to thank every single person who contributed to the Öresundsbron and Öresundståg project and who is working everyday to make it possible for people

like me to commute between Sweden and Denmark.

I acknowledge all of the support and encouragements from my parents Marzieh and Davoud. Finally, I would like to thank my husband Hossein for all his love, patience and understanding during these last three years.

List of Abbreviations

ADMM	Alternating Direction Method of Multipliers
ART	Algebraic Reconstruction Technique
BP	Basis Pursuit
BPDN	Basis Pursuit De-Noising
CP	CANDECOMP/PARAFAC decomposition
CST	Central Slice Theorem
CT	Computed Tomography
DC	Discrepancy principle
DFT	Discrete Fourier Transform
FBP	Filtered Back Projection
FFT	Fast Fourier Transform
IRLS	Iterative Re-weighted Least Squares
ISA	Iterative Shrinkage Algorithm
KKT	Karush-Kuhn-Tucker conditions, i.e., the first-order necessary conditions for optimality
MAE	Mean Approximation Error
MOD	Method of Optimal Directions
MRI	Magnetic Resonance Imaging
NCP	Normalized Cumulative Periodgram

NMF	Nonnegative Matrix Factorization
NNSC	Nonnegative Sparse Coding
NP-hard	Non-deterministic Polynomial-time hard
NTF	Nonnegative Tensor Factorization
OMP	Orthogonal Matching Pursuit
PET	Positron Emission Tomography
SC	Sparse Coding
SIFT	Scale-Invariant Feature Transform
SIRT	Simultaneous Iterative Reconstruction Technique
SSIM	Structural Similarity Index Measure
TV	Total Variation

List of Symbols

A	System matrix: the forward tomography model
C	Polyhedral cone in which the representation of each block in the unknown image lies in (matrix dictionary)
D	Matrix dictionary
D_i^{fd}	Finite-difference approximation of the gradient at each component i
H	Matrix of representation coefficients
I_0	Initial X-ray intensity at the source
I_{L}	X-ray intensity measured by the detector
L	The line in which X-ray moves along
L	A matrix such that Lz for any z is a vector with finite-difference approximations
M_X	Maximum absolute value of the Radon transform of an image X
N_{p}	Total number of projections/views in one tomographic experiment
N_{r}	Number of rays in each projection of a tomographic experiment
P_{C}	Projected representation/approximation of the tomographic image blocks in the cone (matrix dictionary)

P_{G}	Projected representation/approximation of the tomographic image blocks in the cone (tensor dictionary)
RE	Relative reconstruction error
$\mathrm{R}_\theta f(\text{ţ})$	Radon transform of the 2D function f in polar coordinates
$\widehat{\mathrm{R}_\theta f}$	Fourier transform of the Radon transform
S	Sinogram matrix
SSIM	Structural similarity index of two images
TV	Discrete total variation
U, V	Auxiliary variables in the ADMM method (matrix dictionary)
W	Global matrix dictionary for the tomographic unknown image
X_j, $\texttt{squeeze}(\overrightarrow{\mathcal{X}_j})$	Non-overlapping patches in the unknown image
Y	Training data matrix
$A^{(k)}$, $\mathcal{A}(:,:,k)$	The kth frontal slice of the third-order tensor $\mathcal{A}$
$\overrightarrow{\mathcal{A}_j}$, $\mathcal{A}(:,j,:)$	The jth lateral slice of the third-order tensor $\mathcal{A}$
$\mathcal{C}$	Representation tensor coefficient in the global dictionary for the tomographic unknown image
C	Polyhedral cone in which the representation of each block in the unknown image lies in (matrix dictionary)
$\mathcal{D}$	third-order Tensor dictionary
D	The convex set in which the tensor dictionary belongs to
$\text{Đ}(\text{Đ}_\infty, \text{Đ}_2)$	The convex set in which the matrix dictionary belongs to
G	Polyhedral cone in which the representation of each block in the unknown image lies in (tensor dictionary)
$\mathcal{H}$	Third-order tensor of representation coefficients
$\mathcal{I}$	The identity tensor
$\mathcal{K}$	Continuous forward operator of an inverse problem
$\mathscr{L}_{\text{dic}}$	Factorization approximation loss function
$\mathscr{L}_{\text{rec}}$	Data fidelity loss function
$\mathscr{L}_{\text{reg}}$	Regularization penalty function
$\mathcal{L}_\rho$	The augmented Lagrangian objective function in the ADMM method
$\mathit{\Lambda}$, $\bar{\mathit{\Lambda}}$	Lagrange multipliers in the ADMM method

P	Projection operator
$\mathcal{P}\mathrm{R}_{\theta} f(\mathfrak{t})$	Filter in the filter back projection method
$\Phi_{\mathrm{IP}}(\cdot)$	Penalty function imposing prior on the tomographic unknown image
$\Phi_{\mathrm{SP}}(\cdot)$	Penalty function imposing sparsity on the representation coefficient for the unknown image
$\Phi_{\mathrm{dic}}(\cdot)$	The penalty function imposing prior on the dictionary
$\Phi_{\mathrm{rep}}(\cdot)$	The penalty function imposing prior on the representation coefficient
$\mathit{\Phi}$	A known dictionary of arbitrary size
$\mathscr{M}_j$	Mean of the jth column/angle vector in the sinogram matrix
$\mathcal{S}$	Soft thresholding operator
Θ	The direction with most linear trends along it in an image
$\mathcal{U}$, $\mathcal{V}$	Auxiliary variables in the ADMM method (tensor dictionary)
$\mathcal{X}$	A tensor of all non-overlapping patches in the unknown image
$\mathcal{Y}$	Tensor of training patches
b	Vector of tomographic observed noisy data
`circ`	A block circulant matrix
d_j	A vector of one dictionary element
diag	A diagonal matrix
dom	Domain of a function
$f(\cdot)$	Continuous attenuation coefficients/variables of the object/model in a/an tomography/inverse problem
f	Vector of discrete attenuation coefficients of the object in tomography
`fft`	Fast Fourier Transform
`fold`, `unfold`	`unfold`($\mathcal{A}$) takes a $l \times m \times n$ tensor and stacks the frontal slices of the tensor $\mathcal{A}$and returns a block $ln \times m$ matrix, whereas the `fold` command undoes the operation
g	Continuous observed data in an inverse problem
h_j	Vector of representation coefficients for the training image j
k_0	Maximum number of non-zero elements to represent a training image

m	Number of rows in the model matrix equals total number of tomographic measurements
$\lVert \cdot \rVert_{\max}$	$= \lVert \mathrm{vec}(\cdot) \rVert_\infty$
n	Number of columns in the model matrix equals total number of pixels in the unknown image
p	Number of pixels in each column of a training patch
prox	Proximal operator
q	Number of non-overlapping blocks or patches in the unknown tomographic image
r	Number of pixels in each row of a training patch
s	Number of dictionary basis elements equals the number of columns/lateral slices in the dictionary
spark	The smallest number of columns in a matrix that are linearly-dependent
`squeeze`	Reshape a $l \times 1 \times n$ tensor into a $l \times n$ matrix
$\lVert \cdot \rVert_{\mathrm{sum}}$	$= \lVert \mathrm{vec}(\cdot) \rVert_1$
t	Total number of training data/patches
t-product	Third-order tensor multiplication
`trace`	Sum of elements on the main diagonal of a matrix
$\mathfrak{t}$	Line parameter in the Radon transform
tubal fiber	A $1 \times 1 \times n$ tensor
u	Response/observed signal
v_j	Variance of the sinogram matrix for the jth projection/column
`vec`	Transform a matrix/tensor into a vector
x	Vector of unknown parameters/image
y_j	Vector of training image j
$\lVert \cdot \rVert_*$	Nuclear norm: Sum of singular values of a matrix
α	Representation coefficient in the global dictionary for the tomographic unknown image
β	The regularization parameter which balances the data fitting term and the regularization induced by the dictionary
γ	Sparsity level of the representation vector
η	The scale factor

δ	Regularization parameter that controls the block artifacts
ϵ	Error/noise of the representation/forward approximation and the observed data
ε	Tolerance in the stopping criteria for the ADMM method
θ	Angular parameter in the Radon transform
ϑ	The total number of pixels along the boundaries of the patches in the reconstruction image
κ	Condition number of a matrix
λ	Regularization parameter that controls the sparsity of the representation matrix/tensor
$\tilde{\lambda}$	Upper bound for the regularization parameter in the dictionary learning problem
λ_{Tikh}	Tikhonov regularization parameter
λ_{TV}	Total variation regularization parameter
μ	Regularization parameter that controls the sparsity of the representation coefficient
ξ	Total number of pixels in each training patch
ρ	Penalty parameter in the ADMM method
τ	Regularization parameter that controls the sparsity of the representation coefficient of the reconstruction
$\varphi(\mathcal{C})$	Regularization function on the tensor representation coefficient tensor $\mathcal{C}$
$\psi(\cdot)$	Penalty function to reduce block artifacts in the tomographic reconstruction
ω	Representation coefficient of a signal in a known dictionary

Contents

Chapter 1

Introduction and Motivation

Computed tomography (CT), or tomographic imaging, is a technique to construct an image of the interior of an object from measurements obtained by sending X-rays through the object and recording the damping of each ray. CT was developed in the late 1960's and the early 1970's by Godfrey N. Hounsfield of EMI Laboratories, England and Allan M. Cormack of Tufts University, Massachusetts, USA. The Nobel Prize in Physiology or Medicine 1979 was awarded jointly to Cormack and Hounsfield "for the development of computer assisted tomography". CT is still as exciting as the beginning of its development and nowadays is routinely used as a nondestructive testing method in medical imaging, materials science and many other applications. CT is an inverse problem [85], i.e., the problem of estimating the attenuation coefficients of an object, given the observed data and the tomographic system geometry.

One of the main challenges in CT is image reconstruction from imperfect sampling data due to low-dose measurements and/or with projections at few views or with a limited-angle (e.g. due to measurement time or dose constraints). These limited-data scenarios lead to the so called ill-posed problems. In these circumstances the classic methods of CT, such as filtered back projection [68] and algebraic reconstruction techniques [47], are often incapable of producing satisfactory reconstructions, because they fail to incorporate adequate prior information [10].

To overcome these difficulties and improve the reconstruction and *regularize* the solution of an ill-posed problem, it is necessary to introduce and incorporate a priori information into the mathematical reconstruction formulation that can compensate for the lack of data. The prior information can be available in various forms, such as training images, constraints, edge information, statistical priors, etc. It is challenging to design mathematical methods that incorporate the prior information in an optimal way.

A popular prior is that the image is composed of homogeneous regions separated by sharp boundaries leading to Total Variation (TV) regularization methods [69, 113]. TV regularization can be very well suited for edge-preserving imaging problems in low dose and/or few-view data sets. A drawback of the TV methods is that these methods produce images whose pixel values are clustered into regions with somewhat constant intensity with sharp boundaries [105], which results in over-smoothed textural images.

A different approach is to use prior information in form of a carefully chosen set of images referred to as "training images". Training images used as the prior for tomographic reconstruction can be slice pictures or high-accuracy reconstructions of similar specimens which incorporate the important features of the desired solution. Obviously, such images must be reliable and relevant to the specific application. From the statistical point of view, the target image is a realization of an unknown distribution and the training images are the representatives or realizations of such a distribution. The training images should thus be tightly related to the reconstruction problem under study.

The features of training images are obviously not in form of mathematical formulations. *"Dictionary learning"* is an unsupervised learning method to extract the prototype features of training images. The dictionary learning problem is to find "basis elements" and sparse representations of the training signals/images. That is, we want to write the input images/signals as a weighted linear combination of a small number of (unknown) basis (dictionary) elements. The dictionary learning was introduced by Olshausen and Field in 1996 [88]. One can extract and represent prior information from the training images by forming a dictionary that sparsely encodes the information. This approach appears to be very suited for incorporating priors such as image texture that are otherwise difficult to formulate.

What completes this picture comes with the sparse representation theory; the sparse reconstruction problem seeks an approximate representation of a signal/image using a linear combination of a few known dictionary elements. Sparse reconstruction of signals and images has acquired a great deal of interest and has been extensively studied in the last few years, see, e.g., [14, 20, 31, 110]. In the classical framework, for the sparse reconstruction problem, the dictionaries are fixed and predefined. Examples of such dictionaries are Fourier, Curvelets and Wavelets bases. For instance, Wavelets are used in a Bayesian regularization formulation [66] and Curvelets are used for sparse regularization in X-ray computed tomography [115]. Methods using learned dictionaries are computationally more expensive than using precomputed dictionaries in solving inverse problem regularization with sparsity constraints; but they perform better in promoting sparsity while fitting the measurement data, because the dictionary is tailored to the statistics of the solution and optimized for the training set. This concept can be very useful for tomographic reconstruction in general. If the unknown image is sparse in a specific dictionary, the remaining task is to find the representation coefficients that reconstruct the original image from the given measurement data.

The use of dictionary learning in tomographic imaging has been a hot topic in the last few years. Two different approaches have been emerging, either constructing the dictionary from the intermediate solutions in an iterative joint learning-reconstruction algorithm [21, 56, 73, 74, 94], or constructing the dictionary from training images in a separate step before the reconstruction [34, 84, 109, 116, 121]. The simultaneous learning and reconstruction is a non-convex optimization problem. Furthermore, it violates the fundamental principle of inverse problems that a data-independent prior must be incorporated in the problem formulation to eliminate unrealistic solutions that may fit the data.

In this thesis we focus on formulating a regularized tomographic reconstruction problem that incorporates the available information in terms of training images at hand. We first construct a dictionary from training images and then solve the reconstruction problem using the dictionary as a prior to regularize the problem, in a convex optimization framework, via computing a solution that has a sparse representation in the dictionary.

The input images in the aforementioned methods are rearranged as vectors in a matrix. By vectorizing images, the inherent spatial coherences and the original structures would be lost in the second dimension; however, the images themselves can be more naturally represented as a multidimensional array, called a *"tensor"*. Recent work by Kilmer et al. [63] sets up a new theoretical linear algebra framework for tensors. A new method based on [63] for the dictionary learning problem and its application in tomographic image reconstruction using third order tensor formulations is developed. This approach differs from previous approaches in that, first a third-order tensor representation is used for our images and then the dictionary learning problem is reformulated using the tensor formulation. The reconstruction problem is similarly formulated by looking for a solution with a sparse representation in the tensor dictionary. It is shown that it is possible to represent repeated features compactly in the dictionary by using such tensor formulations.

We seek to use realistic simulations with noisy data, we avoid committing "*inverse crime*", i.e., the target image is not contained in the training set. We perform a careful study of the sensitivity of the reconstruction to different parameters in the reconstruction problem and the dictionary. To the best of our knowledge, no previous comprehensive study has investigated and explored the influence of the learned dictionary structure and dictionary parameters in CT.

1.1 Contributions of the Thesis

The aim of this thesis is to provide a theoretical, methodological, and computational framework for the use of training images as priors for the solutions in tomographic reconstruction. The main content of this PhD thesis is based on the collection of two papers [99, 100] and one technical report [98] with the same author as this thesis. The thesis contributions fall in three major categories:

- An algorithmic framework for using the training images as the prior information in image reconstruction is developed: from a set of training images, a regularized non-negative matrix factorization is used to form a dictionary that captures the desired features; then a reconstruction with a sparse representation in this dictionary is computed in terms of a convex optimization problem. A careful study of how to compute a dictionary based on the Alternating Direction Method of Multipliers (ADMM) and how the dictionary parameters influences the reconstruction quality is performed. Simulations show that for textural images, this approach is superior to other methods used for limited-data situations.

  Relevant paper:

  S. Soltani, M. S. Andersen, P. C. Hansen, "*Tomographic Image Reconstruction using Training Images*", submitted, 2015.

  http://arxiv.org/abs/1503.01993

- An empirical study to evaluate the influence of the algorithm and design parameters in our problem formulation, as well as sensitivity to scale and rotation – with focus on robustness – is performed. Algorithms to estimate the correct relative scale and rotation of the unknown image to the training images are presented.

  Relevant paper:

  S. Soltani, *"Studies of Sensitivity in the Dictionary Learning Approach to Computed Tomography: Simplifying the Reconstruction Problem, Rotation, and Scale"*, DTU Compute Technical Report 2015-4, July 2, 2015.

http://orbit.dtu.dk/fedora/objects/orbit:140904/datastreams/file_112138797/content

- The advantage of using a tensor formulation of the problem are demonstrated, that is more natural than the standard matrix formulation when working with images. The problems of dictionary learning in the context of a regularized nonnegative tensor factorization; and the tomographic image reconstruction in a convex optimization framework with a tensor formulation are presented. It is also shown that using such tensor formulations leads to much sparser representations because tensors better allow for identifying spatial coherence in the training images.

  Relevant paper:

  S. Soltani, M. E. Kilmer, P. C. Hansen, *"A Tensor-Based Dictionary Learning Approach to Tomographic Image Reconstruction"*, submitted, 2015.

  http://arxiv.org/abs/1506.04954

1.2 Outline

This thesis is organized as follows.

We first establish backgrounds and fundamentals of this thesis and describe basic definitions in Chapters 2 and 3. Chapter 2 provides the reader with background knowledge of inverse problems and tomographic image reconstruction. In Chapter 3 the stage for the image reconstruction problem using dictionaries is set; providing an overview of the dictionary learning and sparse reconstruction methods and briefly introducing a generic formulation of such a reconstruction problem.

Chapters 4, 5 and 6 are dedicated to the main contributions of this thesis. The Chapters 4 and 5 use the matrix formulation and Chapter 6 use the tensor formulation of our algorithm. In Chapter 4 an algorithm for tomographic image reconstruction where prior knowledge about the solution is available in the form of training images is described. In Chapter 5 the problem formulation assumptions from Chapter 4 is investigated in more details; furthermore, we study the sensitivity of the reconstruction towards changes in scale and rotation and present algorithms to determine the correct scale and rotation from the measurement tomographic data. In Chapter 6 we describe the tomographic image reconstruction using the training images in a tensor formulation. Tensor dictionary learning problem and the corresponding regularized image reconstruction problem in tensor formulation are discussed in 6. The implementation details of the Alternating Direction Method of Multipliers (ADMM) to compute the matrix and tensor dictionaries are also given in Chapters 4 and 6.

Finally in Chapter 7 we discuss the obtained results and suggest possible future directions.

Chapter 2

Inverse Problems and Regularization

In this chapter we will briefly introduce inverse problems with both computational and theoretical prospective where we present the discrete inverse problems in the context of imaging problems. Furthermore, we give a brief overview of the tomographic image reconstruction problem which belongs to the class of discrete inverse problems.

2.1 Discrete Inverse Problems

One can say that a direct problem is a problem which consists of computing the consequences of given causes; then, the corresponding inverse problem consists of finding the unknown causes of known consequences. The definition of a direct-inverse pair must be based on well-established physical laws. In other words the forward problem is to compute the output, given a physical system and the input to the system. The inverse problem in a continuous setting is to compute the input given the two other quantities [46, §1]. The objective of an inverse problem is to find the best model function of parameters f such that

$$\mathcal{K}(f) = g, \tag{2.1}$$

where $\mathcal{K}$ is the forward operator that describes the explicit relationship between model parameters f and the observed data g (i.e., the governing physics).

The inverse problems are often ill-posed. Hadamard [43] gave the definition of a well posed inverse problem:

Well-posed problem

- **Existence:** The problem must have a solution.
- **Uniqueness:** The solution must be unique.
- **Stability:** The solution must depend continuously on the data.

If the problem violates one of the well-posedness conditions, it is said to be ill-posed.

The general discrete inverse problem obtain from discretization of the continuous formulation (2.1), often takes the form of a linear least square problem

$$\min_x \|Ax - b\|_2, \quad A \in \mathbb{R}^{m\times n},\ x \in \mathbb{R}^n,\ b \in \mathbb{R}^m, \tag{2.2}$$

where $m \neq n$, x is the vector of unknown parameters, A is the forward system matrix and b is the observed—often noisy—data.

The (2-norm) condition number of the matrix A is given by:

$$\kappa(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)},$$

where $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ are maximal and minimal singular values of A respectively. The condition number of the inverse problem can be quantified. A linear system with a low condition number is said to be well-conditioned, while a linear system with a high condition number is said to be ill-conditioned. The measure for the ill-posedness of the discrete problem is the decay rate of the singular values. For very large condition numbers, small data perturbations can lead to large reconstruction errors and the least squares solution is far from being a stable solution.

For an ill-posed linear system, the minimization of the data fitting term is ill-posed and must be regularized. Regularization refers to formulating further assumptions in the discrete inverse problem (2.2) in order to obtain a unique and stable solution. We can achieve this by including an additional regularization term in the object function. The generic regularized problem can hence be defined as

$$\min_{x} \mathscr{L}_{\text{rec}}(Ax, b) + \mathscr{L}_{\text{reg}}(x), \tag{2.3}$$

where the data fidelity is measured by the loss function $\mathscr{L}_{\text{rec}}$— often the quadratic 2-norm –and regularization is imposed via penalty function $\mathscr{L}_{\text{reg}}$.

2.1.1 Tikhonov Regularization

Tikhonov regularization method is the most well-known regularization method and has been introduced by Tikhonov in 1977 [108]. Tikhonov's method explicitly incorporate the regularity of the solution in the formulation of the problem. The Tikhonov solution solves the problem

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda_{\text{Tikh}}^2 \|x\|_2^2, \tag{2.4}$$

where $\lambda_{\text{Tikh}} > 0$ is the Tikhonov regularization parameter which controls the weighting between the fidelity measure and the regularity of the solution. The behavior of the Tikhonov regularization method is studied using the SVD analysis in [45, §3]. For a review on algorithms for finding the solution to the Tikhonov regularization problem and corresponding examples please see [45, 46, 102].

The solution to the regularized solution depend on the choice of regularization parameter. One of the well known methods for finding a suitable value for λ_{Tikh} is the L-curve method [44]. Some more automated methods for selecting the regularization parameter has been suggested in the literature, see [45] for a review.

2.1.2 Total Variation Regularization

The total variation (TV) has been originally introduced in image processing by Rudin, Osher and Fatemi [97], as a regularizing criterion for solving inverse problems. It has proved to be quite efficient for regularizing images by requiring the images to have sharp edges and without smoothing the boundaries of the objects [105].

In this model, the prior is formulated such that the solution is sparse in the gradient domain. In order to enforce regularization and obtain a unique and stable solution, the reconstruction image x can be defined as the solution of:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda_{\text{TV}} \text{TV}(x), \tag{2.5}$$

where $\lambda_{\mathrm{TV}} > 0$ is the TV regularization parameter. The discrete TV for the 1D signal is given by:

$$\mathrm{TV}(x) = \sum_{1 \le i \le n-1} |D_i^{\mathrm{fd}} x|, \quad \text{where } |D_i^{\mathrm{fd}} x| = |x_{i+1} - x_i|, \tag{2.6}$$

i.e., D_i^{fd} is the finite-difference approximation of the derivative at the ith point.

In image denoising problems TV regularization method tends to work well on images with regions of constant intensity where it produces a sparse gradient magnitude.

There are other ways to define discrete TVs by means of finite differences, with more symmetric schemes (with 3, 4 or 8 neighbors), or absolute values (1-norm) in higher dimension, but (2.6) is the simplest case which can be efficiently solved by means of the fast Chambolle method [18].

For example, for higher dimension arrays the isotropic TV is defined by

$$\mathrm{TV}(z) = \sum_{1 \le i \le n_z} \|D_i^{\mathrm{fd}} z\|_2, \tag{2.7}$$

whereas the anisotropic TV is given by

$$\mathrm{TV}(z) = \sum_{1 \le i \le n_z} \|D_i^{\mathrm{fd}} z\|_1, \tag{2.8}$$

where n_z is the total number of elements in the high-dimensional array z. The matrix D_i^{fd} computes a finite-difference approximation of the gradient at each pixel/voxel. The TV formulation (2.5) falls into the class of convex optimization problems. A first order method for large-scale convex TV regularization problems is implemented in [59].

2.2 Tomographic Image Reconstruction

Tomography entails the reconstruction of an image from object's interior where the projection data from several different directions is given. Tomography has found widespread application in many scientific fields, including but not limited to medicine, material science, physics, and geoscience.

Computed tomography (CT) is affiliated with X-ray photons transmitted in straight lines through the object of interest along several projection lines. While

X-ray CT may be the most familiar application of tomography, several competing methods, such as magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound, and nuclear-medicine nowadays exist. CT is an interesting model case for inverse problems and many mathematical aspects of CT have been extensively studied and are now well understood, see e.g., [15, 35, 42].

In this thesis we focus on the X-ray CT formulation and since the term "tomography" is often associated with X-ray CT, throughout this thesis we use the term tomography to denote the X-ray CT problem.

In tomographic imaging the projection data is measured by the number of X-ray photons transmitted through the object along individual projection lines while the goal is to compute the linear attenuation coefficient in the slice being imaged. Examples of CT scanner geometries often used to collect projection data is illustrated in Fig. 2.1.

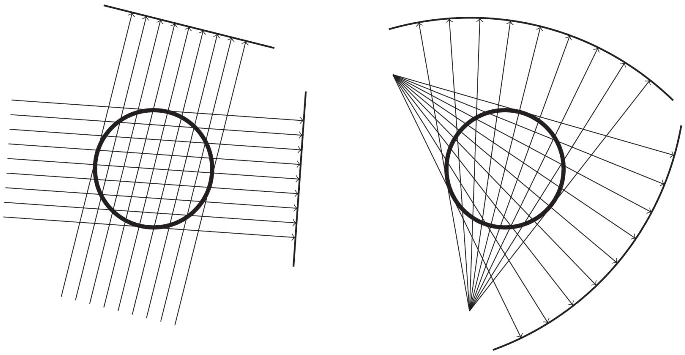


Figure 2.1: Left: the parallel-beam CT geometry with equidistant angles between the detector elements and equidistant X-ray source spacing. Right: the fan-beam CT geometry. A fan-beam sampling unit consists of an X-ray source and a detector array mounted on the same rotation disk. This figure is from [60].

The spatial distribution of the attenuating components of the object that produce the projection data is not known a priori. The X-ray attenuation in the object's components primarily depends on the it's density. X-ray attenuation in tissue can be described by Lambert-Beer's law, see e.g., [15, §2.3.1]. If $f(X)$ is the attenuation coefficient at the spatial position $X = (x_1, x_2)$ in the 2D object, L is the line in which X-rays moves along, I_0 is the initial X-ray intensity and

I_{L} is the intensity when exiting the object, known from measurements, then

$$I_{\mathrm{L}} = I_0 \exp\Big(- \int_{\mathrm{L}} f(X)dx \Big).$$

The Lambert-Beer's law states that the number of photons, decreases exponentially while running through an object along the incident direction. This attenuation is due to absorption and scattering. By taking the logarithm of the Lambert-Beer's law we obtain:

$$\int_{\mathrm{L}} f(X)dx = \log I_0 - \log I_{\mathrm{L}}. \tag{2.9}$$

The right hand side of (2.9) is known and the left-hand side consists of an integral of the unknown function f along a straight line L. This is the tomography's inverse problem. It can be shown that ideally the quantity I_{L}, i.e., the photon count, is a constant multiple of a Poisson-distributed random variable [15, §2.3.1], however in presence of other errors such as beam-hardening and scattering, in practice, it is common to assume a Gaussian noise-model. The Gaussian noise model is accurate when the photon count is large enough [102].

2.2.1 Continuous Tomographic Data

The Radon transform and its inverse provide the mathematical basis for reconstructing tomographic images from measured projection. An object can be perfectly reconstructed from a full set of projections [93].

Let $f(X) = f(x_1, x_2)$ be a continuous function on $\mathbb{R}^2$. The Radon transform is a function defined on the space of straight lines L in $\mathbb{R}^2$ by the line integral along each such line

$$\mathrm{R}f(\mathrm{L}) = \int_{\mathrm{L}} f(X)dx.$$

The Radon transform of the function f for the two-dimensional variable $X = (x_1, x_2)$ in the polar coordinates can be written as

$$\mathrm{R}_\theta f(\text{ţ}) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x_1, x_2)\delta(x_1 \cos\theta + x_2 \sin\theta - \text{ţ})dx_1 dx_2, \tag{2.10}$$

where the angle $\theta \in [0, \pi)$ and the parameter $\text{ţ} \in \mathbb{R}$ define a line such that $\text{ţ} = x_1 \cos\theta + x_2 \sin\theta$. The function $\delta(\cdot)$ is the Dirac delta function.

The 2D Radon transform is a 2D function of θ and ţ called the sinogram, which gives the intensity values in the coordinate system of variables $(\theta, \text{ţ})$. The function $\mathrm{R}_\theta f(\mathfrak{t})$ is often referred to as a sinogram because the Radon transform of an off-center point source is a sinusoid. When the discrete values of θ and t are used, then the sinogram is represented by a matrix.We denote the sinogram by the matrix S.

The task of tomographic reconstruction is to find $f(x_1, x_2)$ given knowledge of $\mathrm{R}_\theta f(\mathfrak{t})$. The solution to the tomographic reconstruction is based on the central slice theorem (CST) [15, §5.3]. To briefly explain, CST results in the idealized reconstruction algorithm for tomographic imaging:

Filter Back Projection Algorithm

- Measure the Radon transform of f:

$$\mathrm{R}_\theta f(\mathfrak{t}) = \log\left(\frac{I_0}{I_\mathrm{L}}\right).$$

Note that I_L is a function of θ and $\mathfrak{t}$.

- Calculate the Fourier transform of $\mathrm{R}_\theta f$ with $\mathfrak{t}$ as the independent variable and with fixed θ, denoted by $\widehat{\mathrm{R}_\theta f}$.

- Denote $\hat{f}$ as the Fourier transform of f such that:

$$\widehat{\mathrm{R}_\theta f} = \hat{f}.$$

- Reconstruct f from the Radon inversion formula:

$$f(X) = \frac{1}{(2\pi)^2} \int_0^\pi \int_{-\infty}^{\infty} \widehat{\mathrm{R}_\theta f}(\mathsf{r}) e^{i\mathsf{r}(x_1 \cos\theta + x_2 \sin\theta)} |\mathsf{r}| d\mathsf{r} d\theta,$$

where i is the imaginary unit.

We denote

$$\mathcal{P}\mathrm{R}_\theta f(\mathfrak{t}) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \widehat{\mathrm{R}_\theta f}(\mathsf{r}) e^{i\mathsf{r}(x_1 \cos\theta + x_2 \sin\theta)} |\mathsf{r}| d\mathsf{r}$$

as a filter. Then:

$$f(X) = \frac{1}{2\pi} \int_0^\pi \mathcal{P}\mathrm{R}_\theta f(x_1 \cos\theta + x_2 \sin\theta) d\theta.$$

Hence, $f(X)$ can be obtained, by definition, as the backprojection of the filter $\mathcal{P}\mathrm{R}_\theta f$. For more detailed description of the filter back projection algorithm we refer the reader to [15, §5] and [85, §2.3].

An implementation of the filtered back projection method can be found as the MATLAB function `iradon.m` from the Image Processing Toolbox. Filter Back projection (FBP) method is fast and require a low memory to compute a solution. However FBP needs a complete projection data i.e., projections in $[0, \pi)$ to compute accurate reconstructed images. Furthermore the FBP formulation does not allow incorporating prior information on the reconstruction.

2.2.2 Discrete Tomographic Data

The discrete tomographic reconstruction model relies on a discrete representation of both the image to be reconstructed and the measurement data. These methods allow integration of prior information in the reconstructing process, as well as the flexible use of diverse linear algebra and optimization techniques.

In the discrete model the image $f(X)$ is represented by the vector f, obtained by dividing the object space into $n = M \times N$ pixels, with elements f_j in a finite basis of n square summable functions $h_j(X)$.

$$f = \sum_{j=1}^{n} f_j h_j(X).$$

Here we choose pixel expansion functions for $h_j(X)$, $j = 1, \ldots, n$. The angular variable is sampled with equidistant steps in the half circle:

$$\theta_l = \frac{l-1}{N_\mathrm{p}}, \quad 1 \leq l \leq N_\mathrm{p}.$$

The linear parameter ţ is also sampled uniformly over a suitable interval:

$$\text{ţ}_k = -T + \frac{2(k-1)}{N_\mathrm{r}}, \quad 1 \leq k \leq N_\mathrm{r},$$

where $T > 0$. The number of rays in each projection is given by N_r and the total number of projections is N_p.

Let $m = N_\mathrm{r} N_\mathrm{p}$, then the measurement b_i of the line integral of f over the line L_i is approximated by

$$b_i \approx \sum_{j=1}^{n} \Big(\int_{L_i} h_j(X) dx_i \Big) f_j, \quad i = 1, \ldots, m.$$

where dx_i denotes the one-dimensional Lebesgue measure along the line L_i. The measurement b_i is equal to $\log\big(\frac{I_0}{I_{L_i}}\big) + \text{noise}$. Hence:

$$b_i \approx \sum_{j=1}^{n} a_{ij} f_j, \quad i = 1, \ldots, m,$$

where a_{ij} is the distance that L_i travels in the jth pixel. An example of a discretized object and a given projection line is depicted in Fig. 2.2.

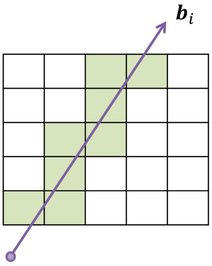


Figure 2.2: A 5×5 example of a discrete image. The length of each pixel side is 1. An attenuation variable f_j is appointed inside each pixel. Only pixels that intersect the line L_i are included in the measurement associated with this line.

By setting up the matrix $A = (a_{ij})$, $i = 1, \ldots, m$ and $j = 1, \ldots, n$, we obtain the linear system of equations $Af \approx b$. In general we consider the discrete linear model as follows:

$$Af + \epsilon = b \tag{2.11}$$

where the vector $\epsilon \in \mathbb{R}^m$ models the measurement noise error.

The sinogram S of the discrete model is given by reshaping the measurement vector as a matrix, where column indices correspond to discrete values of θ while row indices correspond to discrete values of ţ.

Let us consider the Shepp-Logan phantom test image at discretization with 125× 125 pixels generated with `phantom.m` function in MATLAB. We use the function `paralleltomo.m` from the MATLAB package AIR Tools [47] to compute the matrix A with a parallel beam geometry. The `paralleltomo.m` automatically choose $N_\mathrm{p} = 180$ projection and $N_\mathrm{r} = 177$ rays per projection. The Shepp-Logan test image and it's corresponding 177 × 180 singoram obtained by the forward computation Af are given in Fig. 2.3.

2.2.3 Algebraic Reconstruction Techniques

Discretization of the tomographic problem leads to a large, sparse and ill-conditioned system of equations. Iterative regularization methods for computing stable regularized solutions to discrete inverse problems have been widely

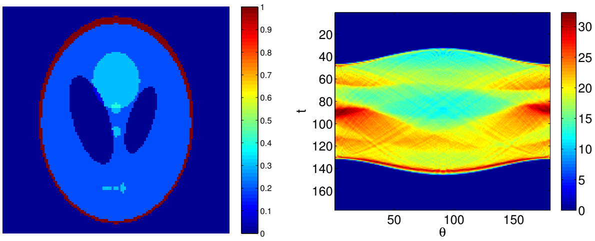


Figure 2.3: Left: The 125×125 Shepp-Logan phantom. Right: The sinogram of tomographic measurements, the column indices correspond to discrete values of θ and row indices correspond to discrete values of ţ.

used in imaging problems, [9, 46] as well as tomographic reconstruction [42, 51]. There are many variants of these iterative methods, which rely on matrix-vector multiplications and therefore are well suited for large-scale problems.

These methods often exhibit fast initial convergence towards the desired solution. The number of iterations plays the role of a regularization parameter because these iterative methods have semi-convergence behavior in the presence of noisy measurement data. When the number of iterations increases, the iterates first approach the unknown image and then diverge from the regularized solution and converge to the least squares solution.

One classical method that is routinely used for tomographic imaging problems is the Kaczmarz method also known as the algebraic reconstruction technique (ART) [40]. The ART and Kaczmarz methods are alternatively called row-action methods, the reason is that they access the matrix A one row or one block at a time. The iteration k involves a sweep through the rows of A, i.e., a_i^T for $i = 1, \ldots, m$, in the following update of the iteration vector:

$$f^{[k^{(i)}]} = f^{[k^{(i-1)}]} + \lambda^{[k]} \frac{b_i - a_i^{\mathrm{T}} f^{[k^{(i-1)}]}}{\|a_i\|_2^2} a_i, \quad i = 1, \ldots, m, \; k = 1, \ldots \tag{2.12}$$

where k is the number of iteration and $\lambda^{[k]}$ is a relaxation parameter such that $0 < \lambda^{[k]} < 2$. The superscripts $[k^{(i)}]$ and $[k^{(i-1)}]$ refer to the ith and $(i-1)$th row value at the iteration number k. If the linear system (2.11) is consistent then the iteration (2.12) converges to a solution $f^\star$, and if $f^{[0]}$ is a linear combinations of column vectors in A^{T}, then $f^\star$ is the solution of minimum 2-norm. If the system is inconsistent then every subsequence associated with a_i converges, but not necessarily to a least squares solution. For proof please see [30].

Another class of iterative methods commonly used in tomography are called simultaneous iterative reconstruction techniques (SIRT). These methods are “simultaneous” in the sense that all the equations are used at the same time in one iteration and involve matrix-vector products, and is given in general form by

$$f^{[k]} = f^{[k]} + \lambda^{[k]} T A^{\mathrm{T}} M (b - A f^{[k]}), \tag{2.13}$$

where T and M are symmetric positive matrices. Different methods depend on the choice of these matrices. A convergence scheme is depicted in [47].

With the semi-convergence properties of the iterative methods we need a reliable stopping criteria that can stop the iterations at the right point. Several stopping criteria are available in literature such as Discrepancy principle (DC) and Normalized Cumulative Periodogram (NCP), see e.g., [45]. For each iterative method, a number of strategies are also available for choosing the relaxation parameter $\lambda^{[k]}$. For a review we refer the reader to [47].

The ART and SIRT methods are well suited for modern computer architectures. ART has the faster convergence during the semi-convergence phase comparing to the SIRT method, however recent block versions of these methods, based on partitioning the linear system, are able to combine the fast semi-convergence of ART with the better multi-core properties of SIRT [104].

Several MATLAB packages with implementations of several algebraic iterative reconstruction methods for the tomographic imaging problems are available. ASTRA [90, 57] is a MATLAB package with GPU acceleration and interfaces to Python, for 2D and 3D tomography. The AIR Tools package [47] was developed for MATLAB, including 2D reconstruction test problems, and techniques.

2.3 The Need for the “Right” Priors

Let us recall from the introduction Chapter 1 that an interesting challenge for image reconstruction in tomography arises from the insufficient sampling data with projection data at few views with the uncertain noisy data.

Consider the Shepp-Logan phantom image shown in Fig. 2.3 discretized on a 125×125 pixel grid as the given exact image x^{exact}. We fix the number of rays per view at $N_{\mathrm{r}} = 177$ and use an angular range of a full $180°$. The number of views is limited to $N_{\mathrm{p}} = 25$, The tomographic measurement model, $A = (a_{ij})$ with $i = 1, \ldots, m$ and $j = 1, \ldots, n$, where m is the number of tomographic measurements and n is the total number of pixels in the target reconstruction solution (x), is obtained by a parallel beam tomographic simulation from the MATLAB Toolbox AIR Tools [47].

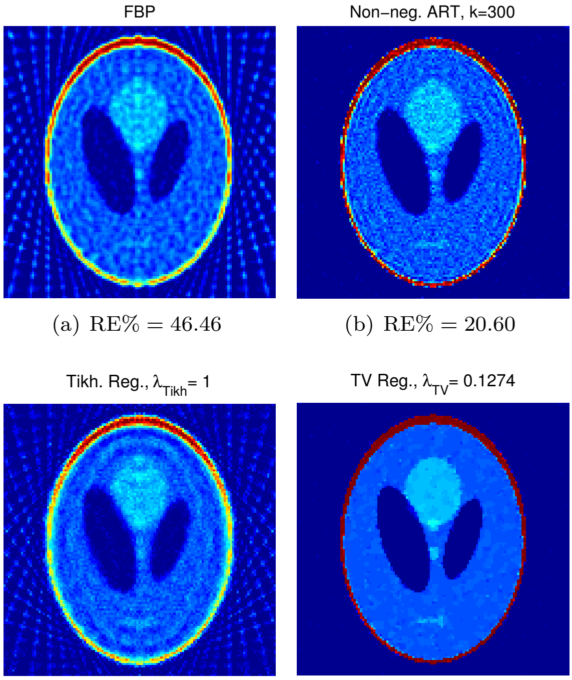
We evaluate the image reconstruction with several reconstruction methods; this is done with noisy data, the exact data is generated with forward approximation $b^{\text{exact}} = Ax^{\text{exact}}$ and 1% relative Gaussian noise is added to the exact data to compute the noisy data. The total number of rows in A is given by $m = N_{\text{p}}N_{\text{r}} = 4425$ and the number of columns is given by $n = 125^2 = 15625$. The reconstruction quality is evaluated by computing the relative error:

$$\text{RE} = \frac{\|x^{\text{exact}} - x\|_2}{\|x^{\text{exact}}\|_2} \tag{2.14}$$

The reconstructed solutions are illustrated in Fig. 2.4 where the tomographic problem is solved by means of FBP, ART, Tikhonov regularization and the TV formulation. The MATLAB toolboxes TVReg [59] and AIR Tools package [47] are used for obtaining the TV and ART solutions. In these methods the regularization parameters were chosen such that they were "optimal" in the sense that they result in a solution with the least relative error.

The FBP, Tikhonov, and ART methods fail to produce desirable reconstructed images in this low-dose tomographic problem, however the reconstruction error for the TV solution is significantly smaller than that for other solutions. The matrix A is ill-conditioned, and rank deficient, due to the ill-posedness of the underlying inverse problem and therefore the solution is very sensitive to noise in the data b. For this reason, this simple least squares approach and FBP fail to produce a meaningful solution, and we must use regularization to incorporate prior information about the solution. TV regularization methods can maintain reconstruction quality or even generate better results than the classical Filter back projection methods in low dose and/or few-view data sets. This leads to a significant reduction of radiation exposure in CT [10, 69, 101, 113].

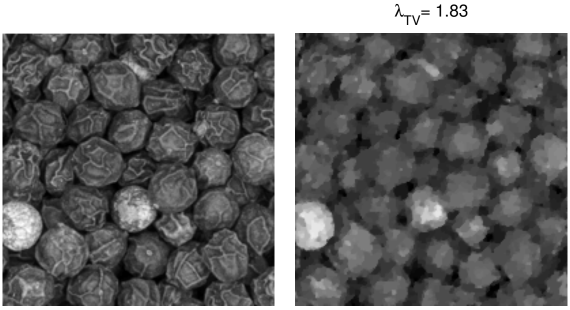
Image textures can generally be found in natural images and images of various materials. Image texture gives us information about the spatial and structural features and various intensities in an image. For example tomographic techniques in material science allows one to obtain the images of the interior of a material in a non-destructive way, collect data about the micro-structural characterization of materials and better understand the main physical phenomena that occur during the forming or the use of a material. To quantify how well textures of an image can be preserved in a tomographic reconstruction process in a low-dose scenario, we consider a test image with textural features. A true image of peppers with clear textures with 200×200 resolution is assumed to be given (in Fig. 2.5). The true image is gray and scaled in the interval $[0, 1]$. We solve the reconstruction problem using the exact image given in Fig. 2.5. For this we use $N_{\text{p}} = 25$ projections, $N_{\text{r}} = 283$ rays per projection, and 1% noise. This problem is highly underdetermined with $m = 7,075$ measurements and $n = 40,000$ unknowns. In Fig. 2.5 the reconstruction, computed by TV



(a) RE% = 46.46 (b) RE% = 20.60

(c) RE% = 43.07 (d) RE% = 4.95

Figure 2.4: Comparison of the best solutions computed by different reconstruction methods. Top: left: filter back projection, and right: algebraic reconstruction technique solutions. Bottom: left: Tikhonov, and right: TV regularization solutions. RE denotes the relative reconstruction error (2.14).



(a) x^{exact} (b) TV, RE% = 21.37

Figure 2.5: Left: The 200 × 200 peppers test image. Right: The TV regularization solution with 25 projections and 1% noise. RE denotes the reconstruction error.

regularization, is illustrated where it can be clearly seen that the TV method fails to capture the textures of peppers.

TV regularization is very well suited for edge-preserving image processing or reconstruction problems; however the main drawback of the TV is that it results in images where the pixel values are clustered into piecewise constant regions [105] and also in presence of noise TV tends to over-smooth the textures for natural images. Since the TV constraint penalizes the image gradient, and is not capable of distinguishing structural details from noise. Another drawback is that the TV problem (4.11) tends to produce reconstructions whose intensities are incorrect [105].

Our ultimate goal is to incorporate priors in order to preserve edges and details in the image. The prior information that is needed for image reconstruction may be available in so called “training images” that characterize the geometrical or visual features of the property of interest, e.g., from pictures of specimens or from high-accuracy reconstructions. The goal of this work is to formulate a variational framework for solving tomographic reconstruction problems in which a priori information is available as such training images.

Chapter 3

Dictionary Learning and Sparse Representation

Finding "good" representations of training data has been the topic of a large amount of research. Unsupervised learning involves learning from unlabeled training sets of data where no specific order or information of the training data is available. The problem of unsupervised learning is that of trying to find hidden structures in unlabeled data and/or blind source separation – the separation of a set of source/basis signals/images from a set of mixed signals/images – using feature extraction techniques for dimensionality reduction; e.g., singular value decomposition, k-means clustering, principal component analysis, independent component analysis, non-negative matrix factorization and many more. For a comprehensive overview of the unsupervised learning methods and applications, at the time of its publication, we refer to [50, §14].

"Dictionary learning" is an unsupervised learning method to learn and extract various features of a(n) signal/image. The dictionary is learned from training data, i.e., a large database of images with an arbitrary size. Dictionary learning is a way to summarize and represent a large number of training images/signals into fewer elements and, at the same time, compensate for noise or other errors in these images/signals; hence the learned dictionary is robust to irrelevant features. The goal of dictionary learning is to represent input signals/images, represented as vectors, approximately as a weighted linear combination of a

small number – introducing sparsity – of (unknown) “basis vectors”. These basis vectors thus capture high-level patterns in the training data. These basis vectors are called the “elements of the dictionary”. To obtain a sparse representation, the dictionaries are typically overcomplete, i.e., dictionaries have more basis functions than it is actually necessary to span the solution space. Such “*sparse coding*” of natural images was introduced by Olshausen and Field in 1996 [88].

One is often interested in approximating an image as a linear combination of a few (aka sparse) elements of the dictionary. Once an unknown signal is sparse in a specific dictionary, the main challenge is to find the representation coefficients that reconstruct the original full signal from the given data. We should note that when a signal is said to be sparse in an engineering sense, it means that the signal is compressible, i.e. it can be expressed either with a small number of dictionary elements or with significantly decaying expansion coefficients.

While learning the dictionary has proven to be critical to achieve (or improve upon) the quality of existing methods and results, effectively solving the corresponding optimization problem is a significant computational challenge, particularly in the context of the large scale datasets involved in image processing tasks, that may include millions of training samples.

In Section 3.1 we describe the dictionary learning problem formulation. In Section 3.2 we will briefly introduce the background on reconstruction methods with a sparse representation and we formulate our general framework to solve the tomographic reconstruction problem using learned dictionary priors in Section 3.3.

3.1 The Dictionary Learning Problem

The term dictionary learning refers to methods of inferring, given a data matrix Y, an overcomplete dictionary that will perform good at sparsely encoding the data in Y i.e., modeling data matrix as sparse linear combinations of the dictionary. A dictionary learning problem can be formulated as follows:

Given a data matrix $Y = [y_1, y_2, \ldots, y_t] \in \mathbb{R}^{\xi \times t}$ and a number of entries s find two matrices $D = [d_1, d_2, \ldots, d_s] \in \mathbb{R}^{\xi \times s}$ and $H \in \mathbb{R}^{s \times t}$, which factorize Y as well as possible, that is: $Y \approx DH$ or in other words $Y = DH + E$, where the matrix $E \in \mathbb{R}^{\xi \times t}$ represents approximation error. This standard generative model assumes that the factorization error is distributed as a zero-mean Gaussian distribution with covariance $\sigma^2 I$. The problem of learning a basis set (dictionary) can be formulated as a matrix factorization problem.

A generic dictionary learning problem takes the form:

$$\min_{D,H} \quad \mathscr{L}_{\text{dic}}(Y, DH) + \Phi_{\text{dic}}(D) + \Phi_{\text{rep}}(H). \tag{3.1}$$

Here, the misfit of the factorization approximation is measured by the loss function $\mathscr{L}_{\text{dic}}$, while the priors on the dictionary D and the representation matrix H are taken into account by the regularization functions Φ_{dic} and Φ_{rep} respectively.

Considering nonnegativity constraints on the elements of D and H or imposing sparsity constraint on matrix H are widely used methods in unsupervised learning for decomposing multivariate data into non-negative or/and sparse components. In this section we briefly describe the motivation behind this type of data representation. We note that the Bayesian methods presented in, e.g., [67, 122] based on maximum likelihood and maximum a-posteriori probability which are designed for training data corrupted by an additive noise and/or being incomplete; are not of our interest in this work.

We present the standard sparse coding, non-negative matrix factorization, and nonnegative sparse coding problem as the well-known dictionary learning formulations and try to explain their relations.

3.1.1 Sparse Coding

The term "Sparse Coding (SC)" comes from the classic paper [88] in which it is shown that a coding strategy that maximizes the sparseness is sufficient to account for capturing natural image features.

In sparse coding the problem of discovering the underlying dictionary is often formulated in terms of vector representations, i.e., each input vector y_j is characterized and represented using basis vectors $[d_1, d_2, \dots, d_s]$ and a sparse vector of weights or "coefficients" $h_j \in \mathbb{R}^s$, $j = 1, \dots, t$ such that $y_j \approx Dh_j$. In its simplest form, the sparsity of the coefficients h_j is measured by its cardinality number. The cardinality is sometimes called the l_0 pseudo-norm, although the cardinality function is not a norm. The cardinality is denoted by $\|\cdot\|_0$, then:

$$\|h_j\|_0 = \#\big\{i \in \{1, \dots, \xi\} \,:\, h_{i,j} \neq 0\big\}, \quad \forall\, j = 1, \dots, t.$$

Commonly an optimization problem of the following form is considered.

$$\min_{D,\{h_j\}_{j=1}^t} \sum_{j=1}^{t} \|y_j - Dh_j\|_2^2, \quad \text{s.t. } \|h_j\|_0 \leq k_0,\ j = 1, \dots, t, \tag{3.2}$$

where every representation has at most k_0 non-zero entries. Let the *spark* of a matrix D be defined as the smallest number of columns from D that are linearly-dependent. In terms of uniqueness, for the case when $y_j = Dh_j$, if D exists such that h_j's for $j = 1, \dots, t$ are representable using at most $k_0 < spark(D)/2$ atoms; then, up to re-scaling and permutation of the columns, D is the unique dictionary that achieves this sparsity for all the elements in the training database [31, §12.2.1].

A local minimum to the problem posed in equation (3.2) can be approximated iteratively, first minimizing over h_j's with D fixed, and then minimizing over D with h_j's fixed. Dictionary learning algorithms for (3.2) based on such strategy has been proposed for instance in Method of Optimal Directions (MOD) by Engan et al. [33] and K-SVD by Aharon et al [1]. Recent modifications and improvements of the MOD and K-SVD dictionary learning algorithms are also proposed, e.g., in [82, 103, 118].

However minimizing $\|\cdot\|_0$ is known to be a NP-hard problem [110], instead commonly in the formulation it is replaced by the l_1 regularization, leading to a convex relaxation of the sparse coding problem (3.2) when the dictionary D is fixed. Therefore to favor sparse coefficients, the sparsity prior for each coefficient h_j is defined as $\|h_j\|_1$. The l_1-norm regularization is known to produce sparse coefficients and can be robust to irrelevant features [87].

To prevent the dictionary D from having arbitrarily small values (which would lead to arbitrarily large values of in h_j), or vise versa, one can introduce constraints on the l_2-norm of the matrix columns

$$Đ \equiv \left\{ D \in \mathbb{R}^{\xi \times s} \mid \|d_j\|_2^2 \leq \xi, \ \forall j \in \{1, \dots, s\} \right\}.$$

Then the search for a sparse code can be formulated as an optimization problem by constructing the following cost function to be minimized:

$$\min_{D \in Đ, h_j \in \mathbb{R}^s} \frac{1}{2} \|y_j - Dh_j\|_2^2 + \lambda \|h_j\|_1, \quad \text{for } j = 1, \dots, t, \tag{3.3}$$

The emphasis on minimizing the sparsity induced on the elements h_j, is controlled by the regularization parameter $\lambda \geq 0$. Similar to the problem (3.2) the non-convex optimization problem (3.3) is commonly solved alternatively for D and h_j for all $j = 1, \dots, t$. The h_j updating step is a sparse linear problem −which we will describe in the next section− and the D update is a norm constrained least squares problem. Most recent methods for solving these types of problems are based on coordinate descent (gradient methods) with soft thresholding [78]. For examples of sparse coding algorithms based on problem formulation (3.3), we refer to [71] and the online dictionary learning method [79].

Using the definition of the l_1-norm and the Frobenius norm of a matrix, lead the problem definition in (3.3), to a more general representation. More specifically, given a training set of t signals $Y = [y_1, \ldots, y_t]$ in $\mathbb{R}^{\xi \times t}$, one looks for a dictionary matrix D in such that each signal y_j admits a sparse decomposition in D:

$$\min_{D \in \text{Ð}, H \in \mathbb{R}^{s \times t}} \frac{1}{2} \|Y - DH\|_{\text{F}}^2 + \lambda \sum_{i,j} |H_{i,j}| \tag{3.4}$$

Note that the problem formulation (3.4) is proper since the columns of the representation matrix H are independent and separable. Note that the problem (3.4) is an example of the generic problem formulation (3.1). The optimization problem (3.4) is not jointly convex in (D, H) and hence there is no guarantee to obtain the global minimum, but it is convex with respect to each variable D or H when the other is fixed.

3.1.2 Nonnegative Matrix Factorization

Many real world data are nonnegative and the corresponding basis elements have a physical meaning only when nonnegative. Lee and Seung in [70] proposed the notion of non-negative matrix factorization (NMF), as a way to find a set of basis functions for representing nonnegative data. It is shown in [70] that the basis vectors displayed as images, appear as a collection of parts and localized features, so one can say that NMF leads to a parts-based representation. NMF only allow additive not subtractive combinations, where a zero-value represents the absence and a positive number represents the presence of the basis component in the representation.

In principle NMF seeks to decompose a non-negative matrix. Given a nonnegative matrix $Y \in \mathbb{R}_+^{\xi \times t}$, NMF searches for non-negative factors D and H that approximate Y (i.e., $Y \approx DH$) where all the entries of D and H are nonnegative. The NMF problem is commonly reformulated as the following optimization problem:

$$\min_{D \in \mathbb{R}^{\xi \times s}, H \in \mathbb{R}^{s \times t}} \frac{1}{2} \|Y - DH\|_{\text{F}}^2 \quad \text{s.t. } D \geq 0 \text{ and } H \geq 0, \tag{3.5}$$

where D is a basis matrix and H is a coefficient matrix. The matrices D and H are forced to have non-negative entries, which lead to sparse representation [28]. We note that even in situations where $Y = DH$ holds exactly, the decomposition is not be unique [28].

A natural way of optimizing the cost function in the non-convex optimization problem (3.5) is to alternate the minimization between D and H, fixing one and optimizing with respect to the other. Numerous methods are proposed in literature for solving the non-negative matrix factorization problem. One can mention iterative multiplicative algorithms, the alternating least squares algorithms and projected gradient methods. A comprehensive overview, at the time of its publication, of non-negative matrix factorizations and applications exists [24]. Projected gradient approaches are better suited in solving the overcomplete non-negative matrix factorization problems (i.e., $\xi < s \ll t$) [119].

3.1.3 Nonnegative Sparse Coding

In the standard sparse coding, the data is described as a combination of elementary features involving both positive and negative elements. The fact is that features can cancel each other out. Moreover as mentioned in the previous section solutions obtained by NMF algorithms may not be unique, and it is often necessary to impose additional constraints such as sparsity. Furthermore, matrix factorization methods with non-negativity and sparsity constraints usually lead to estimation of the dictionary elements with specific structures and physical interpretations, in contrast to other dictionary learning methods [53].

It is clear, however, that with inducing both sparsity and non-negativity constraints some of the explained variance in the data may decrease. In other words, there is a trade-off between the two goals of interpretability, promoting sparsity and data/statistical fidelity.

For these reasons we prefer to consider the dictionary learning problem which takes the form of non-negative sparse coding [53] of a non-negative data matrix Y:

$$\min_{D,H} \quad \frac{1}{2}\|Y - DH\|_{\mathrm{F}}^2 + \lambda \sum_{i,j} |h_{i,j}| \qquad \text{s.t.} \qquad D \in \text{Đ}, \; H \in \mathbb{R}_+^{s\times t}, \tag{3.6}$$

where the set Đ is convex and $\lambda \geq 0$ is a regularization parameter that controls the sparsity-inducing penalty $\sum_{i,j} |h_{i,j}|$.

A nonnegative dictionary D with s elements refers to a collection of basis image “carrying image features” and a nonnegative H represents conic combinations of dictionary elements when approximating a nonnegative data matrix Y. A sparse H refers to the approximation of training images with a small number of dictionary elements.

A projected gradient descent algorithm for NMF with sparseness constraints− or the nonnegative sparse coding (NNSC) problem− is introduced in [53]. The problem NNSC is currently solved with projected gradient methods from bound-constrained optimization problems [86].

In Chapters 4 and 6 we present an algorithm based on the alternating direction method of multipliers (ADMM) for solving the dictionary learning problem of the form (3.6) and a third-order tensor formulation of such problem.

3.2 Sparse Solution of Linear Inverse Problem

In sparse approximation problem, the goal is to find an approximate representation of a response signal(data) using a linear combination of a few known basis elements from fewer measurements than is required for reconstructing the original signal. In other words consider $\varPhi \in \mathbb{R}^{\xi \times s}$, a known dictionary with $\xi < s$ and a response signal u, the generic discrete inverse problem of finding the representation vector ω is considered by:

$$\text{Find sparse } \omega \quad \text{such that } \varPhi\omega = u,$$

where $\omega \in \mathbb{R}^s$ and $u \in \mathbb{R}^\xi$. In such formulation, the problem is underdetermined, $\xi < s$, thus admits an infinite number of solutions. A way of solving this ill-posed problem is constraining the possible solutions with prior information, here by exploiting sparsity. Recall that a signal ω is sparse if there are a few nonzeros among the possible entries in ω and a simple sparsity measure of the vector ω is defined by the l_0 pseudo-norm. The basic problem of finding a maximally sparse representation of an observed signal u is given by

$$(P_0): \quad \min_{\omega \in \mathbb{R}^s} \|\omega\|_0 \quad \text{subject to } \varPhi\omega = u. \tag{3.7}$$

In practice, signals tend to be weakly sparse or compressible when only a few of their entries have a large magnitude, while most of them are close to zero, rather than being zero. Mathematically speaking, a compressible signal u is sparse in $\varPhi$, if the sorted coefficients in decreasing magnitude have a fast decay; i.e., most of coefficients ω vanish but a few.

The optimization problem (P_0)(3.7) in literature is referred to the “Matching Pursuit” problem. The *spark* gives a simple criterion for uniqueness of sparse solutions in (P_0). If a system of linear equations $\varPhi\omega = u$ has a solution obeying $\|\omega\|_0 < spark(\varPhi)/2$, this solution is necessarily the sparsest possible (for a proof see [14]).

One can consider a natural variation of the problem (P_0), and allow a small discrepancy between $\varPhi\omega$ and u with some error tolerance $\epsilon \geq 0$ [26]. This is the case when the measurement signal u is contaminated by noise

$$(P_0^\epsilon): \quad \min_{\omega \in \mathbb{R}^s} \|\omega\|_0 \quad \text{subject to } \|\Phi\omega - u\|_2 \leq \epsilon. \tag{3.8}$$

Sparse regularization is a popular class of priors to model natural signals and images. Given a predefined complete basis functions (e.g., Discrete Cosine Transform) or an overcomplete dictionary (e.g., Wavelets, or learned dictionaries), we are interested in an efficient encoding of the data, in the sense of sparseness, i.e., to use as few dictionary components as possible in our representation. The sparsity inducing norms perform model selection as well as regularization.

3.2.1 Algorithmic Approaches

The problem (P_0)(3.7) and (P_0^ϵ)(3.8) being non-convex and NP-hard, a straightforward approach to solve them seems intractable. There are at least five major classes of computational techniques for solving sparse approximation problems, we list them from the Tropp and Wright review paper in [110].

1. **Brute force**: Exhaustive combinatorial search through all possible support sets which is plausible only for small-scale problems.
2. **Greedy pursuit**: Iteratively refine a sparse solution by successively identifying one or more components that yield the greatest improvement in quality [83].
3. **Convex relaxation**: Replace the combinatorial problem with a convex optimization problem. Solve the convex program with algorithms that exploit the problem structure [20, 76].
4. **Bayesian framework**: Assume a prior distribution for the unknown coefficients that favors sparsity and develop a maximum a posteriori estimator that incorporates the observation [91].
5. **Nonconvex optimization**: Relax the l_0 problem to a related nonconvex problem and attempt to identify a stationary point [19, 41].

A basic suboptimal greedy sequential solver for (P_0)(3.7) and (P_0^ϵ)(3.8) is known as “Orthogonal Matching Pursuit algorithm” (OMP) [92]. The OMP algorithm iteratively generates for the signal u and the dictionary Φ, a sorted list of indexes and scalars which are the sub-optimal solution to the problem of sparse signal representation and yields a substantial improvements in approximating the signal. Many related greedy pursuit algorithms have been proposed in literature, please see, e.g., [31, §3.1].

Since Bayesian framework and non-convex optimization methods do not currently offer theoretical guarantees [110], we only focus on convex optimization formulations to obtain solutions to the sparse approximation problems. It is not clear where a convex relaxation formulation is preferable to a greedy algorithm technique however convex relaxation algorithms are more effective in a wider variety of settings, such as in presence of heavy noise in the measurement signal [110].

3.2.2 Convex Relaxation Methods

Recall that the l_1 norm is the closest convex relaxation to the l_0 pseudo norm function. The convex form of (P_0) (3.7) also known as "Basis Pursuit" (BP) [20], which is the solution having the smallest l_1 norm of coefficients, is given by

$$(P_1): \quad \min_{\omega \in \mathbb{R}^s} \|\omega\|_1 \quad \text{subject to } \Phi\omega = u. \tag{3.9}$$

By emperical observation, in many cases (P_1) successfully finds the sparsest representation [31, §3.2].

An equivalent representation of (P_0^ϵ)(3.8) is given by

$$(P_1^\epsilon): \quad \min_{\omega \in \mathbb{R}^s} \|\omega\|_1 \quad \text{subject to } \|\Phi\omega - u\|_2 \leq \epsilon, \tag{3.10}$$

where ϵ is an estimate of the noise level in the data. Some authors refer to (P_1^ϵ) (3.10) as the "Basis Pursuit Denoising" (BPDN).

Another variant of the BP problem known as the "Lasso", which specified by Tibshirani [107] is as follows:

$$\min_{\omega \in \mathbb{R}^s} \|\Phi\omega - u\|_2 \quad \text{subject to } \|\omega\|_1 \leq \gamma, \tag{3.11}$$

where the parameter $\gamma > 0$ controls the sparsity level of the representation ω. We can also use a parameter $\mu > 0$ to balance the twin objectives in (P_1^ϵ) and Lasso problems of minimizing both error and sparsity and obtain:

$$\min_{\omega \in \mathbb{R}^s} \frac{1}{2}\|\Phi\omega - u\|_2^2 + \mu\|\omega\|_1. \tag{3.12}$$

For appropriate parameter choices of ϵ, μ, and γ, the solutions of BPDN (3.10), relaxed Lagrangian formulation (3.12), and Lasso (5.1), coincide, and these problems are in some sense equivalent. However, except for special cases – such as Φ orthogonal – the parameters that make these problems equivalent cannot be known a priori [112].

The Basis Pursuit problem formulations (3.10), (5.1) and (3.12) being convex, they can be solved by means of convex optimization techniques. The interior-point method was first used in solving the BP problem [20], and simple iterative algorithms such as “iteratively re-weighted least squares” (IRLS) were developed to solve the relaxed Lagrangian problem (3.12) [31, §5.3]. The paper by Figueiredo et al. [36] proposes gradient projection algorithms for the bound-constrained quadratic programming formulation of the Lagrangian relaxation problem (3.12).

In general the interior-point methods are not as efficient as the gradient methods with very sparse solutions. In recent years, a new efficient family of optimization techniques called the “Iterative Shrinkage Algorithms” (ISA) based on the classical Donoho-Johnson shrinkage method [27] have been developed. This class of methods can be viewed as an extension to the classical gradient algorithm. For an extensive list and description of such methods we refer to [31, §6]. Most of such methods e.g., FISTA [5] are concerned with the unconstrained Lagrangian problem formulation.

There are fewer methods specially adapted to Lasso (5.1) and BPDN (3.10), SPGL1 [112] is a solver specifically designed for BPDN (3.10) and Lasso (5.1). SPGL1 can efficiently handle large scale problems, the issue is that currently it cannot handle variations in their mathematical formulations. NESTA [7] can efficiently deal with variations of the objective functional (3.10), but it has limitations due to requirement of inverting $\varPhi\varPhi^*$, where $\varPhi^*$ is the conjugate transpose of $\varPhi$.

Becker, Candès and Grant in [8] have developed a framework for solving a variety of convex cone problems including BP, BPDN and Lasso, and variations of these problem formulations, using optimal first-order methods. TFOCS is a library (MATLAB Toolbox) based on [8, 6] designed to facilitate the construction of the first-order methods which handles a variety of Basis Pursuit problem formulations. Hence, we will use TFOCS to solve the convex optimization sparse approximation problem in our image reconstruction step which we will describe later.

3.3 Application to Tomographic Reconstruction

Recall that in tomography a noisy measurement signal b is measured as the response of sending physical signals (e.g., waves, particles, currents) through an object of interest. The discrete tomographic model is represented by an m by n matrix A, representing the forward projection model. Considering an unknown $M \times N$ image x, with $n = MN$ as a vector of absorption coefficients for pixels/voxels of the image of interest; yields the linear inverse problem: $b \approx Ax$ where $x \in \mathbb{R}^n$. Our work is concerned with underdetermined problems where $m < n$, and the need for regularization is even more pronounced.

Generally the image x is not sparse but the situation changes when we know that x has a sparse representation in terms of a known basis W, i.e., we can find a solution to the problem

$$\text{Find sparse } \alpha \quad \text{such that } AW\alpha \approx b,$$

where $x = W\alpha$. The simplest dictionary, the identity matrix is a naive dictionary where no prior about the image is incorporated in the representation of the solution. Here we are interested in using a global learned dictionary W for the image x.

Since the observation b is always in presence of error, it is natural to consider the following problem formulation to allow some error tolerance $\epsilon \geq 0$:

$$\min_{\alpha \in \mathbb{R}^{\varrho}} \|\alpha\|_1 \quad \text{subject to } \|AW\alpha - b\|_2 \leq \epsilon, \tag{3.13}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $W \in \mathbb{R}^{n \times \varrho}$ is an overcomplete dictionary (i.e., $n \leq \varrho$), ϵ is a small positive constant and $\alpha \in \mathbb{R}^{\varrho}$ is the unknown variable. The number of dictionary elements (ϱ) is arbitrary here. We can then reconstruct x from the solution α as $x^\star := W\alpha^\star$. In other words, the solution to (3.13) should be a linear combination of a small number of "elements" from the dictionary. The formulation (3.13) is a simplistic model where no other assumptions are made on the image x or representation α.

Consider the case when W is an orthogonal complete basis (i.e., $n = \varrho$), then such problems as (3.13) correspond to a so-called *synthesis* regularization because one can assumes the sparsity of the coefficients α that synthesize the signal $x = W\alpha$. We should also here refer to the *analysis* problem:

$$\min_{x \in \mathbb{R}^n} \|W^{\mathrm{T}}x\|_1 \quad \text{subject to } \|Ax - b\|_2 \leq \epsilon, \tag{3.14}$$

In the analysis problem we are looking for an x such that $Ax \approx b$ and $W^{\mathrm{T}}x$ is sparse. In the synthesis problem, we seek a solution of the form $x = W\alpha$ such that $Ax \approx b$ and α is sparse. In a synthesis prior, the generative vector α is sparse in the dictionary W whereas in analysis prior, the correlation between the signal x and the dictionary W is sparse. Our problem formulation falls into a synthesis prior model.

Dealing with an overcomplete learned dictionary W, in a very generic formulation α solves the problem

$$\min_{\alpha} \quad \mathscr{L}_{\text{rec}}(AW\alpha, b) + \Phi_{\text{SP}}(\alpha) + \Phi_{\text{IP}}(W\alpha), \tag{3.15}$$

where the data fidelity is measured by the loss function $\mathscr{L}_{\text{rec}}$ – often the l_2-norm – and regularization is imposed via penalty functions. Specifically, the function Φ_{SP} enforces the sparsity prior on α, often formulated in terms of the sparsity inducing norm ($\| \cdot \|_1$), while the function Φ_{IP} enforces the Image Prior.

Note how the generic problem (3.15) is related to the formulations (2.3) and (3.12). In the next chapter we describe one of many ways to efficiently implement such a scheme.

Chapter 4

Tomographic Image Reconstruction Using Dictionary Priors

Finding low-dimensional representations of given images in a well chosen basis set is intuitively useful for image reconstruction: suppose that we have at hand a dictionary which is good at representing a class of images, i.e., the images admit sparse representations over the dictionary. Then, one hopes that a sparse approximation of the reconstruction solution with the given dictionary significantly reduces the amount of noise without losing important information and can also compensate for the lack of data. Experiments have shown that such a model with sparse coding is very effective in many applications.

In 2006, Elad and Ahron [32] address the image denoising problem using a process that combines dictionary learning and reconstruction. They use a dictionary trained from a noise-free image using the K-SVD algorithm [1] combined with an adaptive dictionary trained on patches of the noisy image. It is shown in [32] that both dictionaries perform very well in the denoising process. Since then, the dictionary learning approach has been explored in areas such image denoising [22, 72, 96], image deblurring [75], image restoration [82] and image classification [80]. The dictionary learning approach in tomographic imaging is likewise beginning to emerge recently, e.g., X-ray tomography [34, 116], spectral computed tomography [121], magnetic resonance imaging (MRI) [56, 94], ultrasound tomography [109], electron tomography [74], positron emission tomography (PET) [21] and phase-contrast tomography [84].

Most of these works use K-SVD to learn the dictionary (except [116] that uses an online dictionary learning method [79]), All of these works use the methods to regularize the reconstruction by means of a penalty that the reconstruction should be close to the subspace spanned by the dictionary images. While all these methods perform better than classical reconstruction methods, they show no significant improvement over the TV-regularized approach.

As mentioned in the introduction Chapter 1, some works ([21, 56, 73, 74, 94]) use a joint formulation that combines the dictionary learning problem and the reconstruction problem into one optimization problem, i.e., the dictionary is learned from the given noisy data. This corresponds to a "bootstrap" situation where one creates the prior as part of the solution process and it is unclear how the properties of the dictionary influence the computed reconstruction. Our work is different: we use a prior that is already available in the form of a set of training images, and we use this prior to regularize the reconstruction problem. To do this, we use a two-stage algorithm where we first compute the dictionary from the given training images, and then we use the dictionary to compute the reconstruction. Our two-stage algorithm is inspired by the work in [34] and, to some extent, [116]. However, the algorithm in [34] is tested on a simple tomography setup with no noise in the data and in [116] the dictionary is trained from an image reconstructed by a high-dose X-ray exposure and then used to reconstruct the same image with fewer X-ray projections.

We utilize the dictionary in a different way than the mentioned works, using non-overlapping blocks of the image (that we will describe in details in Section 4.3) which reduces the number of unknowns in the reconstruction problem.

Recall from Section 3.3 that the proposed framework for dictionary-based tomographic reconstruction consists of two conceptual steps: (i) computing a dictionary (using techniques from machine learning) from the training images, and (ii) computing a reconstruction composed of images from the dictionary. Our goal is to incorporate prior information e.g., about texture from a set of training images. We focus on formulating and finding a learned dictionary W from the training images and solving the tomography problem such that $x = W\alpha$ is a sparse linear combination of the dictionary elements (the columns of W). We build on ideas from sparse approximation theory [14, 31, 110].

Our reconstruction reconstruction scheme is depicted in Fig. 4.1 which we will describe in Sections 4.1 and 4.3 in details.

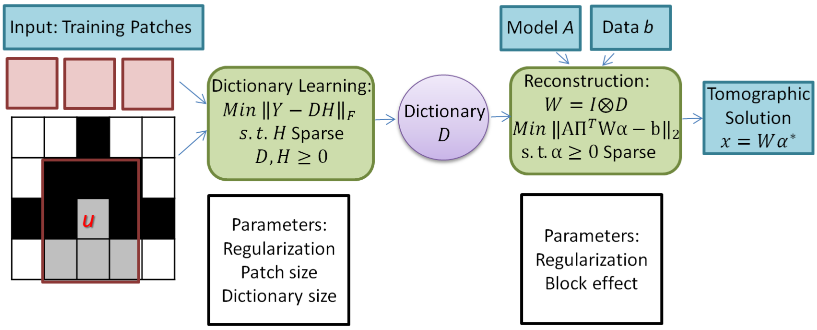


Figure 4.1: The reconstruction scheme.

The main contributions of this chapter are:

- A two-stage reconstruction framework is presented; first the dictionary learning problem is formulated as a nonnegative sparse coding problem, and then a reconstruction that is sparse with respect to the learned dictionary is computed, where, the reconstruction problem is formulated as a convex optimization problem.

- An algorithm based on the ADMM method is implemented to approximate a learned dictionary.

- The influence of the parameters of the dictionary on the reconstruction is empirically studied.

- The proposed method is compared with TV and classical reconstruction methods for solving the few-view/limited-angle tomographic problems for images that resemble texture. It is shown that in few-projection low-dose settings our algorithm while being superior to the classical reconstruction method and competitive with total variation regularization, tends to include more texture and more correct edges.

In this chapter we use the following notations, where A is an arbitrary matrix:

$$\|A\|_\mathrm{F} = \left(\sum_{ij} A_{ij}^2\right)^{1/2}, \quad \|A\|_\mathrm{sum} = \sum_{ij} |A_{ij}|, \quad \|A\|_\mathrm{max} = \max_{ij} |A_{ij}|.$$

A vector $g \in \mathbb{R}^n$ is a subgradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at $x \in \mathrm{dom} f$ if

$$f(z) \geq f(x) + g^\mathrm{T}(z - x) \quad \forall z \in \mathrm{dom} f.$$

If f is convex and differentiable then its gradient at x is the subgradient, and a subgradient can exist even when f is not differentiable at x. The subdifferential $\partial f(x)$ of f at x is the set of all subgradients:

$$\partial f(x) = \{g \mid g^{\mathrm{T}}(z-x) \leq f(z) - f(x), \forall z \in \mathrm{dom} f\}.$$

A set C is called a cone if for every $x \in \mathsf{C}$ and $\theta \geq 0$, we have $\theta x \in \mathsf{C}$. A set C is a convex cone if it is convex and a cone.

4.1 The Dictionary Learning Formulation

The dictionary should comprise all the important features of the desired solution. The number of training images should be large enough to ensure that all image features are represented, and the dictionary should preferably be overdetermined to ensure that one can sparsely realize the desired reconstructions. Using training images of the same size as the image to be reconstructed would require a huge number of training images and lead to an enormous dictionary. The dictionary based methods process training images patch by patch. The dictionary is able to capture local image features effectively because of analyzing training images in a patch-based nature. Therefore we must use patches of smaller size taken from the training images to train a patch dictionary D, and then built the global dictionary W from the found D.

We extract training patches of size $p \times r$ from our training image/images. Let the matrix $Y \in \mathbb{R}^{\xi \times t}$ consist of t training image patches arranged as vectors of length $\xi = pr$. Then a dictionary D can be computed by means of the generic dictionary learning problem (3.1), where $D \in \mathbb{R}^{\xi \times s}$ is the dictionary of s dictionary image patches, and $H \in \mathbb{R}^{s \times t}$ contains information about the approximation of each of the training image patches.

Dictionary learning problems of the form (3.1) are generally non-convex optimization problems because of the bilinear term DH where both D and H are variables. Applying a convergent iterative optimization method therefore does not guarantee that we find a global minimum (only a local stationary point). To obtain a good dictionary, we must be careful when choosing the loss functions $\mathscr{L}_{\mathrm{dic}}$ and the penalties Φ_{dic} and Φ_{rep} on D and H, and we must also pay attention to implementation issues such as the starting point; see Section 4.2 for details.

As mentioned in Section 3.1.3, a non-negative matrix factorization (NMF) has the ability to extract meaningful factors [70], and with non-negative elements in D its columns represent a basis of images. Similarly, having non-negative elements in H corresponds to each training image being represented as a conic combination of dictionary images, and the representation itself is therefore nonnegative. Additionally, NMF often works well in combination with sparsity constraints [53] which in our application translates to training image patches being represented as a conic combination of a small number of dictionary elements (basis images).

The dictionary learning problem that we will use henceforth takes the form of nonnegative sparse coding [53] of a nonnegative data matrix Y:

$$\min_{D,H} \quad \frac{1}{2}\|Y - DH\|_{\mathrm{F}}^2 + \lambda \|H\|_{\mathrm{sum}} \qquad \text{s.t.} \qquad D \in Đ,\ H \in \mathbb{R}_+^{s\times t}, \tag{4.1}$$

where the set Đ is compact and convex and $\lambda \geq 0$ is a regularization parameter that controls the sparsity-inducing penalty $\|H\|_{\mathrm{sum}}$. In our approach we affect sparsity implicitly through l_1-norm regularization and via the regularization parameter λ. This problem is an instance of the more general formulation (3.1) if we define

$$\mathscr{L}_{\mathrm{dic}}(Y, DH) = \frac{1}{2}\|Y - DH\|_{\mathrm{F}}^2$$

and

$$\Phi_{\mathrm{dic}}(D) = I_{Đ}(D), \qquad \Phi_{\mathrm{rep}}(H) = I_{\mathbb{R}_+^{s\times t}}(H) + \lambda\|H\|_{\mathrm{sum}}\ ,$$

where I_{Z} denotes the indicator function of a set Z. Note that the loss function is invariant under a scaling $D \mapsto \zeta D$ and $H \mapsto \zeta^{-1}H$ for $\zeta > 0$. Thus, letting $\zeta \to \infty$ implies that $\Phi_{\mathrm{rep}}(\zeta^{-1}H) \to 0$ and $\|\zeta D\| \to \infty$ if D is nonzero. This means that Đ must be compact to ensure that the problem has well-defined minima. Here we will consider two different definitions of the set Đ, namely

$$Đ_\infty \equiv \{D \in \mathbb{R}_+^{\xi\times s} \,|\, \|d_j\|_\infty \leq 1\} \quad \text{and} \quad Đ_2 \equiv \{D \in \mathbb{R}_+^{\xi\times s} \,|\, \|d_j\|_2 \leq \sqrt{\xi}\}.$$

The set $Đ_\infty$ corresponds to box constraints, and $Đ_2$ is a spherical sector of the 2-norm ball with radius $\sqrt{\xi}$. As we will see in the Section 4.4, the use of $Đ_\infty$ as a prior gives rise to binary-looking images (corresponding to the vertices of $Đ_\infty$) whereas $Đ_2$ gives rise to more "natural looking" images.

We use the ADMM method (see e.g. [11]) to compute an approximate local minimizer of (4.1). Learning the dictionary with an ADMM method has the advantages that it is less dependent on the initial dictionary, and it changes the initial dictionary drastically during the first few steps. At the same time the updates are cheap to compute, making the method suited for large-scale problems. The implementation details are given in the next section.

4.2 The Dictionary Learning Algorithm

The dictionary learning problem (4.1) being non-convex, it is too costly to solve it globally. We will therefore optimize locally by applying the ADMM method [11] to the following reformulation of (4.1)

$$\begin{array}{ll} \text{minimize}_{D,H} & \frac{1}{2}\left\|Y-UV\right\|_{\mathrm{F}}^{2}+\lambda\left\|H\right\|_{\mathrm{sum}}+I_{\mathbb{R}_{+}^{s\times t}}(H)+I_{\text{Đ}}(D) \\ \text{subject to} & D=U,\;\; H=V, \end{array} \tag{4.2}$$

where $U\in\mathbb{R}^{\xi\times s}$ and $V\in\mathbb{R}^{s\times t}$ are auxiliary variables that are introduced in order to make the ADMM-updates separable and hence cheap. The augmented Lagrangian associated with (4.2) can be expressed as

$$\begin{aligned} \mathcal{L}_{\rho}(D,H,U,V,\mathit{\Lambda},\mathit{\Lambda}) = \frac{1}{2}&\|Y-UV\|_{\mathrm{F}}^{2}+\lambda\left\|H\right\|_{\mathrm{sum}}+I_{\mathbb{R}_{+}^{s\times t}}(H)+I_{\text{Đ}}(D) \\ &+\mathrm{Tr}(\mathit{\Lambda}^{\mathrm{T}}(D-U))+\mathrm{Tr}(\bar{\mathit{\Lambda}}^{\mathrm{T}}(H-V)) \\ &+\frac{\rho}{2}\|D-U\|_{\mathrm{F}}^{2}+\frac{\rho}{2}\|H-V\|_{\mathrm{F}}^{2} \end{aligned} \tag{4.3}$$

where $\mathit{\Lambda}\in\mathbb{R}^{\xi\times s}$ and $\bar{\mathit{\Lambda}}\in\mathbb{R}^{s\times t}$ are Lagrange multipliers, and ρ is a positive penalty parameter which can be chosen fixed prior to the learning process. If we partition the variables into two blocks (D,V) and (H,U) and apply ADMM to (4.2), we obtain an algorithm where each iteration involves the following three steps: (i) minimize L_ρ jointly over D and V; (ii) minimize L_ρ jointly over H and U; and (iii) update the dual variables $\mathit{\Lambda}$ and $\bar{\mathit{\Lambda}}$ by taking a gradient-ascent step. Since L_ρ is separable in D and V, step (i) can be expressed as two separate updates

$$D_{k+1}=\min_{D\in\text{Đ}}L_{\rho}(D,H_k,U_k,V_k,\mathit{\Lambda}_k,\bar{\mathit{\Lambda}}_k)=\mathsf{P}_{\text{Đ}}(U_k-\rho^{-1}\mathit{\Lambda}_k) \tag{4.4a}$$

$$\begin{aligned} V_{k+1}&=\min_{V}L_{\rho}(D_k,H_k,U_k,V,\mathit{\Lambda}_k,\bar{\mathit{\Lambda}}_k) \\ &=(U_k^{\mathrm{T}}U_k+\rho I)^{-1}(U_k^{\mathrm{T}}Y+\bar{\mathit{\Lambda}}_k+\rho H_k) \end{aligned} \tag{4.4b}$$

where $P_{\text{Đ}}(\cdot)$ is the projection onto the set Đ. Similarly, L_ρ is also separable in H and U, so step (ii) can be written as

$$\begin{aligned} H_{k+1}&=\min_{H\in\mathbb{R}_{+}^{s\times t}}L_{\rho}(D_{k+1},H,U_k,V_{k+1},\mathit{\Lambda}_k,\bar{\mathit{\Lambda}}_k) \\ &=\mathsf{P}_{\mathbb{R}_{+}^{s\times t}}(\mathcal{S}_{\lambda/\rho}(V_{k+1}-\rho^{-1}\bar{\mathit{\Lambda}}_k)) \end{aligned} \tag{4.4c}$$

$$\begin{aligned} U_{k+1}&=\min_{U}L_{\rho}(D_{k+1},H_k,U,V_{k+1},\mathit{\Lambda}_k,\bar{\mathit{\Lambda}}_k) \\ &=(YV_{k+1}^{\mathrm{T}}+\mathit{\Lambda}_k+\rho D_{k+1})(V_{k+1}V_{k+1}^{\mathrm{T}}+\rho I)^{-1} \end{aligned} \tag{4.4d}$$

where $\mathcal{S}_{\lambda/\rho}$ denotes an entrywise soft-thresholding operator, and $\mathsf{P}_{\mathbb{R}_+^{s\times t}}(\cdot)$ is the projection onto the non-negative orthant. Finally, the dual variable updates in step (iii) are given by

$$\varLambda_{k+1} = \varLambda_k + \rho(D_{k+1} - U_{k+1}) \tag{4.4e}$$
$$\bar{\varLambda}_{k+1} = \bar{\varLambda}_k + \rho(H_{k+1} - V_{k+1}). \tag{4.4f}$$

The projection onto the set Đ_∞ is an element-wise projection onto the interval $[0,1]$ and hence easy to compute. However, the projection onto Đ_2 does not have a closed form solution, so we compute it iteratively using Dykstra's alternating projection algorithm [12]. The iterative scheme which approximates the projection onto the set Đ_2 is given in Algorithm 2.

Algorithm 1 Dykstra's Projection Algorithm

```
Input: The vector $\bar{u}_j$, $\forall\ j = 1,\dots,s$.
Output: $\mathsf{P}_{\text{Đ}_2}(\bar{u}_j)$.
Initialization: Set $\bar{x}_1 = \bar{u}_j$, $\bar{p}_1$ and $\bar{q}_1$ to be zero vectors $\in \mathbb{R}^\xi$ .
for $k = 1, 2, \dots$ do
    $\bar{y}_k = \max(0, \bar{x}_k + \bar{p}_k)$.
    $\bar{p}_{k+1} = \bar{x}_k + \bar{p}_k - \bar{y}_k$
    $\bar{x}_{k+1} = \bar{y}_k + \bar{q}_k$
    $\bar{x}_{k+1} = \frac{\bar{x}}{\max(\|\bar{x}\|_2/\sqrt{\xi},1)}$
    $\bar{q}_{k+1} = \bar{y}_k + \bar{q}_k - \bar{x}_{k+1}$
    if $\|\bar{y}_k - \bar{x}_{k+1}\|_{\text{F}} < 10^{-3}$ then
        Exit
    end if
end for
```

The map $\mathcal{S}_{\lambda/\rho}$ is defined component-wise as follows:

$$\mathcal{S}_{\lambda/\rho}(\varTheta) = \begin{cases} \theta_{i,j} + \frac{\lambda}{2\rho} & \text{if } \theta_{i,j} < -\frac{\lambda}{2\rho} \\ 0 & \text{if } |\theta_{i,j}| < \frac{\lambda}{2\rho} \\ \theta_{i,j} - \frac{\lambda}{2\rho} & \text{if } \theta_{i,j} > \frac{\lambda}{2\rho} \end{cases}$$

The convergence properties of ADMM when applied to non-convex problems of the form (4.2) have been studied by e.g. [117]. They show that whenever the sequence of iterates produced by (4.4) converges, the limit satisfies the the KKT-conditions (i.e., the first-order necessary conditions for optimality) which can be expressed as

$$D = U, \quad H = V,$$

$$\Lambda = -(Y - DH)H^{\mathrm{T}}, \quad \bar{\Lambda} = -D^{\mathrm{T}}(Y - DH),$$
$$-\Lambda \in \partial\Phi_{\mathrm{dic}}(D), \quad -\bar{\Lambda} \in \partial\Phi_{\mathrm{rep}}(H).$$

The convergence result is somewhat weak, but empirical evidence suggests that applying ADMM to non-convex problems often works well in practice [11]. It is interesting to note that the point $D = U = 0$ and $H = V = 0$ satisfies the KKT-conditions, and although it is a stationary point, it is clearly not a local minima. For this reason, we avoid initializing with zeros. We initialize U with some of the images in the training set and we set $V = [I\ 0]$ (i.e., the leading s columns of V is the identity matrix).

The KKT-conditions can be used to formulate stopping criteria. We use the following conditions

$$\frac{\|D - U\|_{\max}}{\max(1, \|D\|_{\max})} \leq \varepsilon \quad \wedge \quad \frac{\|H - V\|_{\max}}{\max(1, \|H\|_{\max})} \leq \varepsilon \tag{4.5a}$$
$$\frac{\|\bar{\Lambda} - D^{\mathrm{T}}(DH - Y)\|_{\max}}{\max(1, \|\bar{\Lambda}\|_{\max})} \leq \varepsilon \quad \wedge \quad \frac{\|\Lambda - (DH - Y)H^{\mathrm{T}}\|_{\max}}{\max(1, \|\Lambda\|_{\max})} \leq \varepsilon \tag{4.5b}$$

where $\varepsilon > 0$ is a given tolerance.

The KKT-conditions can also be used to derive an upper bound $\tilde{\lambda}$ for the regularization parameter λ. It follows from the optimality conditions that for $H = 0_{s\times t}$, $\bar{\Lambda} = -D^{\mathrm{T}}Y$ and hence for some $\tilde{\lambda}$ and all $D \in \text{Đ}$ we have

$$D^{\mathrm{T}}Y \in \tilde{\lambda}\, \partial\|0_{s\times t}\|_{\mathrm{sum}},$$

i.e., $H = 0$ satisfies the first-order optimality conditions for all $\lambda \geq \tilde{\lambda}$. If all entries in Y are between 0 and 1, then the upper bound $\tilde{\lambda} = \xi$ can be used for both dictionaries since

$$\sup_{D\in \text{Đ}_2} \|D^{\mathrm{T}}Y\|_{\max} = \max_{j=1,\ldots,t} \sqrt{\xi}\|Ye_j\|_2 \leq \xi$$

and

$$\sup_{D\in \text{Đ}_\infty} \|D^{\mathrm{T}}Y\|_{\max} = \max_{j=1,\ldots,t} \|Ye_j\|_1 \leq \xi$$

which implies that $D^{\mathrm{T}}Y \in \tilde{\lambda}\, \partial\|0_{s\times t}\|_{\mathrm{sum}}$ for all $D \in \text{Đ}$.

4.3 The Reconstruction Problem

Recall that we formulate the discrete tomographic reconstruction problem as $Ax \approx b$, where b contains the noisy data and A is the system matrix (see Section

2.2.2). The vector x represents an $M \times N$ image of absorption coefficients, and these coefficients must be nonnegative to have physical meaning. Hence we must impose a nonnegativity constraint on the solution. A simple/naive tomographic reconstruction problem for Gaussian noise could thus be formulated as

$$\min_x \frac{1}{2} \|Ax - b\|_2^2 \qquad \text{s.t.} \qquad x \in \mathbb{R}_+^n. \tag{4.6}$$

Referring to (2.3), the loss function $\mathscr{L}_{\text{rec}}$ is represented by the residua l's l_2-norm and the non-negativity of the image is imposed as a prior. As investigated in Section 2.3 due to the ill-posed nature of the underlying problem, the lack of other priors results in unsatisfactory result.

We now turn to the reconstruction problem based on the patch dictionary D and problem formulation (3.15). We divide the reconstruction into nonoverlapping blocks of the same size as the patches and use the dictionary D within each block (ensuring that we limit blocking artifacts); conceptually this corresponds to building a global dictionary W from D. For ease of our presentation we assume that the image size is a multiple of the patch size. Since the patch dictionary images are generally much smaller than the desired reconstruction ($p \ll M$ and $r \ll N$), we partition the image into an $(M/p) \times (N/r)$ array of non-overlapping blocks or patches represented by the vectors $x_j \in \mathbb{R}^\xi$ for $j = 1, \dots, q = (M/p)(N/r)$. The advantage of using non-overlapping blocks, compared to overlapping blocks, is that we avoid over-smoothing the image textures when averaging over the overlapping regions, and it requires less computing time.

Each block of x is expressed as a conic combination of dictionary images, and hence the dictionary prior is expressed as

$$\Pi x = W\alpha, \quad W = (I \otimes D), \qquad \alpha \geq 0, \tag{4.7}$$

where Π is a permutation matrix, W is the global dictionary for the image, and

$$\alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_q \end{pmatrix} \in \underbrace{\mathbb{R}^s \times \cdots \times \mathbb{R}^s}_{q \text{ times}}$$

is a vector of coefficients for each of a total of q blocks. With this non-overlapping formulation, it is straightforward to determine the number of unknowns in the problem 4.7. The dimension of α is $sq = sn/\xi$ which is equal to the product of the over-representation factor s/ξ and the number of pixels n in the image. The permutation matrix Π re-orders the vector x such that we reconstruct the image block by block.

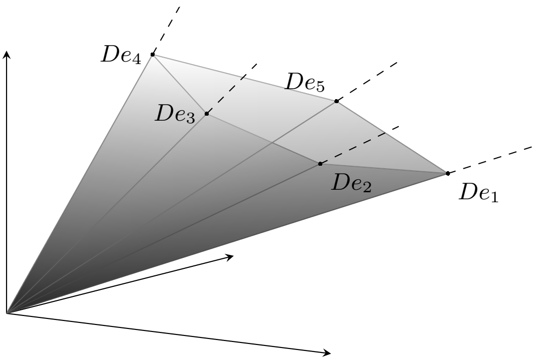


Figure 4.2: Polyhedral cone in $\mathbb{R}^{\xi}_{+}$ spanned by five nonnegative dictionary elements, where e_i denotes the ith canonical unit vector in $\mathbb{R}^{s}$.

In pursuit of a nonnegative image x, we impose the constraint that the vector α should be nonnegative. This implies that each block x_j of x lies inside a polyhedral cone

$$\mathsf{C} = \{Dz \,|\, z \in \mathbb{R}^{s}_{+}\} \tag{4.8}$$

where $\mathsf{C} \subseteq \mathbb{R}^{\xi}_{+}$ since the dictionary images are all nonnegative. This is illustrated in Fig. 4.2. Clearly, if the dictionary contains the standard basis of $\mathbb{R}^{\xi}$, then C is equivalent to the entire nonnegative orthant in $\mathbb{R}^{\xi}$. However, if the cone C is a proper subset of $\mathbb{R}^{\xi}_{+}$, then not all nonnegative images have an exact representation in C, and hence the constraints $x_j \in \mathsf{C}$ may have a regularizing effect even without a sparsity prior on α. This can also be motivated by the fact that the faces of the cone C consist of images x_j that can be represented as a conic combination of at most $\xi - 1$ dictionary images.

Adding a sparsity prior on α, in addition to nonnegativity constraints, corresponds to the belief that x_j can be expressed as a conic combination of a small number of dictionary images and hence provides additional regularization. We include a l_1-norm regularizer in our reconstruction problem as an approximate sparsity prior on α.

Reconstruction based on non-overlapping blocks often gives rise to block artifacts in the reconstruction because the objective in the reconstruction problem does not penalize jumps across the boundaries of neighboring blocks. To mitigate this type of artifact, we add a penalty term that discourages such jumps. We choose a penalty of the form

$$\psi(z) = \frac{1}{M(M/p-1)+N(N/r-1)} \frac{1}{2} \|Lz\|_2^2 \tag{4.9}$$

where L is a matrix such that Lz is a vector with finite-difference approximations of the directional derivatives across the block boundaries. The denominator is the total number of pixels along the boundaries of the blocks in the image.

The constrained least squares reconstruction problem is then given by

$$\begin{array}{ll} \text{minimize}_{\alpha} & \frac{1}{2}\frac{1}{m}\|A\Pi^{\mathrm{T}}(I \otimes D)\alpha - b\|_2^2 + \mu\,\frac{1}{q}\|\alpha\|_1 + \delta^2\,\psi(\Pi^{\mathrm{T}}(I \otimes D)\alpha) \\ \text{subject to} & \alpha \geq 0 \end{array} \tag{4.10}$$

with regularization parameters $\mu, \delta > 0$. We seek to make the problem formulation normalized by i) division of the squared residual norm by the number of measurement m, ii) division of the l_1-norm constraint by the number of blocks q, and iii) the scaling used in ψ (4.9).

Relaxing the non-negativity constraint or the l_1-norm penalty on the representation vector α in (4.10) can be considered as a different choice of priors (less strong ones) under the same problem formulation assumptions in (3.15). The problem (4.10) is a convex but non-differentiable optimization problem which belongs to the class of sparse approximation problems, for which several algorithms have been developed recently (see Section 3.2 for details).

4.4 Numerical Experiments

In this section we use numerical examples to demonstrate and quantify the behavior of our two-stage algorithm and evaluate the computed reconstructions. In particular we explore the influence of the dictionary structure and its parameters (number of elements, patch sizes) on the reconstruction, in order to illustrate the role of the learned dictionary.

The underlying idea is to compute a regularized least squares fit in which the solution is expressed in terms of the dictionary, and hence it lies in the cone C (4.8) defined by the dictionary elements. Hence there are two types of errors in the reconstruction process. Typically, the exact image does not lie in the cone C, leading to an *approximation error*. Moreover, we encounter a *regularization error* due to the combination of the error present in the data and the regularization scheme.

In the learning stage we use a data set of images which are similar to the ones we wish to reconstruct. The ground-truth or exact image x^{exact} is not contained in the training set, so that we avoid committing an inverse crime. All images are gray-level and scaled in the interval $[0, 1]$.

We use the discrete TV regularization problem formulation as

$$\min_{x \in \mathrm{Q}} \quad \frac{1}{2} \|A\, x - b\|_2^2 + \lambda_{\mathrm{TV}} \sum_{1 \le i \le n} \|D_i^{\mathrm{fd}} x\|_2 \tag{4.11}$$

where $\mathrm{Q} = \{x \in \mathbb{R}^n \mid 0 \le x_i \le 1\}$, D_i^{fd} computes a finite-difference approximation of the gradient at each pixel, and $\lambda_{\mathrm{TV}} > 0$ is the TV regularization parameter.

All experiments were run in MATLAB (R2011b) on a 64-bit Linux system. The reconstruction problems are solved using the software package TFOCS (Templates for First-Order Conic Solvers) [8]. We compare with TV reconstructions computed by means of the MATLAB software TVReg [59], with filtered back projection solutions computed by means of MATLAB's `iradon` function, and solutions computed by means of the algebraic reconstruction technique (ART, also known as Kaczmarz's method) with nonnegativety constraints implemented in the MATLAB package AIR Tools [47]. (We did not compare with Krylov subspace methods because they are inferior to ART for images with sharp edges.)

4.4.1 The Training Image and the Tomographic Test Problem

The test images in Sections 4.4.2–4.4.4 are chosen as square patches from a high-resolution photo of peppers with uneven surfaces, making them interesting test images for studies of the reconstruction of textures. Figure 4.3 shows the 1600 × 1200 high-resolution image and the exact image of dimensions $M \times N = 200 \times 200$. This size allows us to perform many numerical experiments in a reasonable amount of time; we demonstrate the performance of our algorithm on a larger test problem in Section 4.4.5.

All test problems represent a parallel-beam tomographic measurement, and we use the function `paralleltomo` from the MATLAB package AIR Tools [47] to compute the matrix A. The data associated with a set of parallel rays is called a projection and the number of rays in each projection is given by $N_{\mathrm{r}} = \lfloor \sqrt{2} N \rfloor$. If the total number of projections is N_{p} then the number of rows in A is $m = N_{\mathrm{r}} N_{\mathrm{p}}$ while the number of columns is $n = MN$. Recall that we are interested in scenarios with a small number of projections. The exact data is generated with the forward model after which we add Gaussian white noise, i.e., $b = Ax^{\mathrm{exact}} + e$.

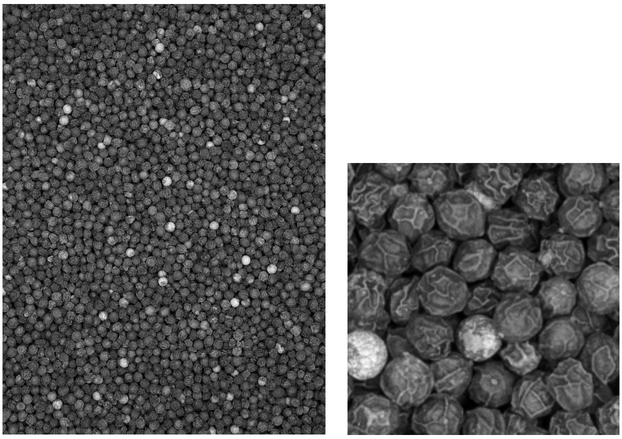


Figure 4.3: Left: the high-resolution image from which we obtain the training image patches. Right: the 200×200 exact image x^{exact}.

4.4.2 Studies of the Dictionary Learning Stage

It is not straightforward to evaluate the performance of the dictionary learning algorithm, considering that we are dealing with a non-convex optimization problem. In addition, the computed dictionary must be validated to estimate how well it will perform in practice. We are aware that the parameters of the dictionary learning algorithm may have an impact on the obtained dictionary, so it is of our great interest to study how these parameters affect the dictionary and – as a result – the reconstruction.

A good dictionary should preserve the structural information of the training images as much as possible and, at the same time, admit a sparse representation as well as a small factorization error. These requirements are related to the number of dictionary elements, i.e., the number of columns s in the matrix D. Since we want a compressed representation of the training images we choose s such that $\xi \leq s \ll t$, and the precise value will be investigated. The optimal patch size $p \times r$ is unclear and will also be studied; without loss of generality we assume $p = r$.

The regularization parameter λ in (4.1) balances the matrix factorization error and the sparsity constraint on the elements of the matrix H. The larger the λ, the more weight is given to minimization of $\|H\|_{\text{sum}}$, while for small λ more weight is given to minimization of the factorization error. If $\lambda = 0$ then (4.1) reduces to the classical nonnegative matrix factorization problem.

From the analysis of the upper bound on the regularization parameter λ in the Section 4.2, we know $\lambda \geq \xi$ implies $H = 0$; so λ can be varied in the interval $(0, \xi]$ to find dictionaries with different sparsity priors. Note that the scaling of the training images affects the scaling of the matrix H as well as the regularization parameter λ.

To evaluate the impact of the dictionary parameters, we use three different patch sizes (5×5, 10×10, and 20×20) and the number of dictionary elements s is chosen to be 2, 3, and 4 times the of the number of rows ξ in dictionary D.

The training patches are easy to acquire. Note that for example in a 256×256-size image, about $61,000$ overlapping 10×10 patches can be extracted. We extract more than $50,000$ patches from the high-resolution image in Fig. 4.3, and for different combinations of patch sizes and number of dictionary elements we solve the dictionary learning problem (4.1). Figure 4.4 shows examples of such learned dictionaries, where columns of D are represented as images; we see that the penalty constraint $D \in Ð_\infty$ gives rise to "binary looking" dictionary elements while $D \in Ð_2$ results in dictionary elements that use the whole gray-scale range.

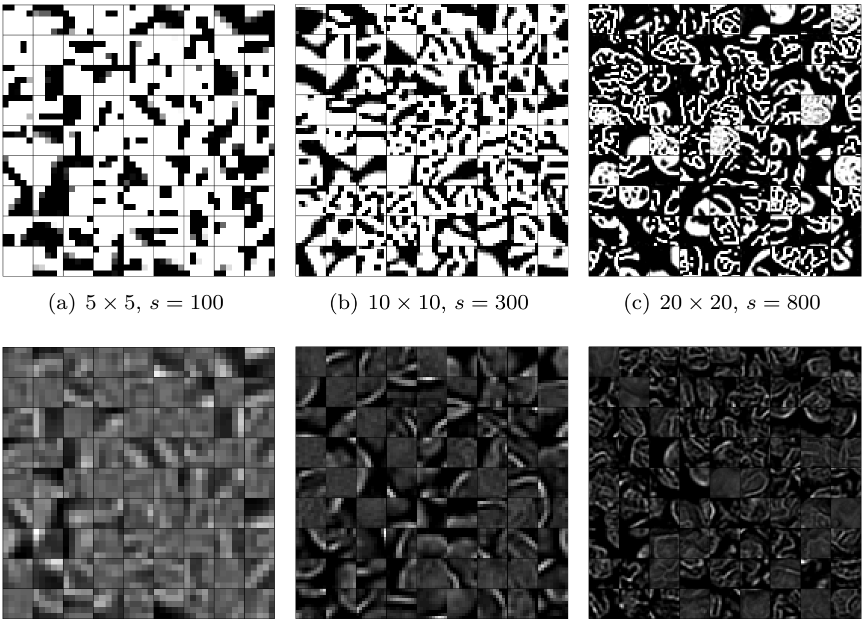
To evaluate the approximation error, i.e., the distance of the exact image x^{exact} to its projection on the cone C (4.8), we compute the solutions $\alpha_j^\star$ to the q approximation problems for all blocks $j = 1, 2, \ldots, q$ in x^{exact},

$$\min_{\alpha_j} \frac{1}{2} \left\| D\alpha_j - x_j^{\text{exact}} \right\|_2^2, \qquad \text{s.t.} \qquad \alpha_j \geq 0. \tag{4.12}$$

If P_{C} is the projection into the cone C, then $P_{\mathsf{C}}(x_j^{\text{exact}}) = D\alpha_j^\star$ is the best representation/approximation of the jth block in the cone. The mean approximation error (MAE) is then computed as

$$\text{MAE} = \frac{1}{q} \sum_{j=1}^{q} \frac{1}{\sqrt{\xi}} \left\| P_{\mathsf{C}}(x_j^{\text{exact}}) - x_j^{\text{exact}} \right\|_2. \tag{4.13}$$

The ability of the dictionary to represent features and textures from the training images, which determines how good reconstructions we are able to compute, depends on the regularization parameter λ, the patch size, and the number of dictionary elements. Figure 4.5 shows how the mean approximation error MAE (4.13) and mean l_1-norm of the columns of H (i.e. $\|H\|_{\text{sum}}/t$) associated with the dictionary varies with patch size ξ, number of dictionary elements s, and regularization parameter λ. An advantage of larger patch sizes is that the variation of MAE with s and λ is less pronounced than for small patch sizes, so overall we tend to prefer larger patch sizes. In particular, for a large patch size



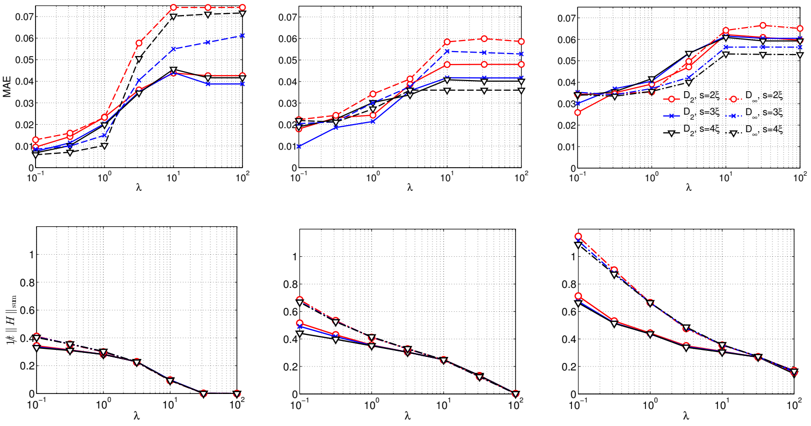
(a) 5×5, $s = 100$ (b) 10×10, $s = 300$ (c) 20×20, $s = 800$

(d) 5×5, $s = 100$ (e) 10×10, $s = 300$ (f) 20×20, $s = 800$

Figure 4.4: Examples of dictionary elements. Top row: with the constraint $D \in \text{Đ}_\infty$ the images appear as "binary looking." Bottom row: with the constraint $D \in \text{Đ}_2$ the images appear to use the whole gray-scale range.

we can use a smaller over-representation factor s/ξ than for a small patch size. From the analysis of the upper bound on λ (see Section 4.2) we expect that for 5×5, 10×10 and 20×20 patch sizes with $\xi = 25$, 100 and 400 respectively, $\|H\|_{\text{sum}} = 0$. This analysis is consistent with the values of mean l_1-norm of columns of H plotted in Fig. 4.5. As λ approaches ξ we have that $\|H\|_{\text{sum}}$ approaches 0, , for relatively large value of λ with respect to the patch sizes, the dictionary D takes arbitrary values, and the approximation errors level off at a maximum value. Regarding the two different constraints $D \in \text{Đ}_\infty$ and $D \in \text{Đ}_2$ we do not see any big difference in the approximation errors for 10×10 and 20×20 patches. From the given $\|H\|_{\text{sum}}/t$ plots in Fig. 4.5, we can argue that for representing gray-scale patches (as in the reference image) with binary looking images in Đ_∞, a larger number of dictionary elements may be needed. To limit the amount of results we now use $D \in \text{Đ}_2$.

The computational work depends on the patch size and the number of dictionary elements which, in turn, affects the approximation error: the larger the



5 × 5 patches

10 × 10 patches

20 × 20 patches

Figure 4.5: Mean approximation errors (4.13) and $1/t\|H\|_{\mathrm{sum}}$. Results for both $D \in Ð_\infty$ and $D \in Ð_2$ with different patch sizes and different s.

dictionary, the smaller the approximation error, but at a higher computational cost. We have found that a good trade-off between the computational work and the approximation error can be obtained by increasing the number of dictionary elements until the approximation error levels off.

Convergence plots for $\lambda = 0.1,\ 1,\ 10$, $p = r = 10$ and $s = 300$ are shown in Fig. 4.6. For $\lambda = 10$ we put emphasis on minimizing the sparsity penalty, and after few iterations we have reached convergence where the residual term dominates the objective function. For $\lambda = 0.1$ we put more emphasis on minimizing the residual term, and we need more iterations to converge; since the objective function is dominated by the sparsity penalty. The objective values in Fig. 4.6 are slightly smaller for dictionary elements in $Ð_2$.

4.4.3 Studies of the Reconstruction Stage

Here we evaluate the overall reconstruction framework including the effect of the reconstruction parameters as well as their connection to the dictionary learning parameter λ and the patch size.

We solve the reconstruction problem (4.10) using the exact image given in Fig. 4.3. We choose $N_{\mathrm{p}} = 25$ projections corresponding to uniformly distributed angles in $[0°, 180°]$. Hence the matrix A has dimensions $m = \lfloor\sqrt{2}\cdot 200\rfloor \cdot 25 =$

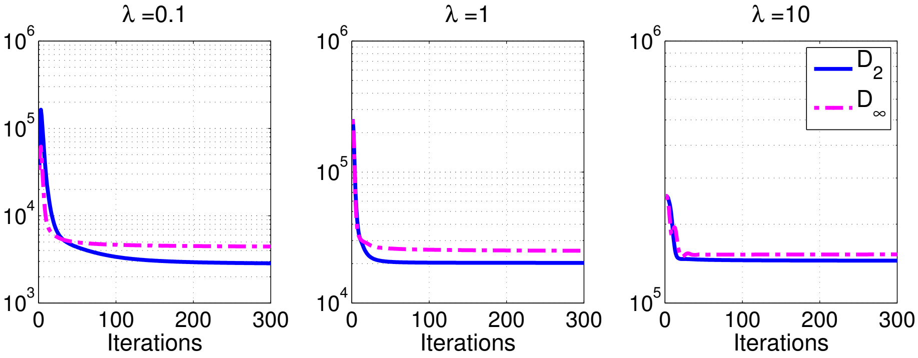


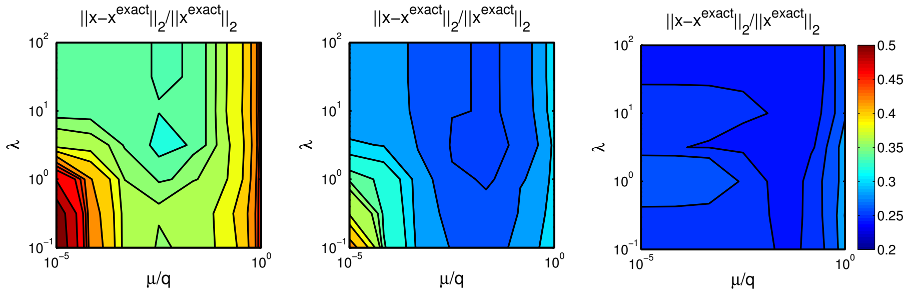
Figure 4.6: Convergence of ADMM algorithm in Section 4.2 for $\lambda = 0.1$, 1, and 10, $\xi = 100$ and $s = 300$. We plot $1/2\|Y - DH\|_F^2 + \lambda\|H\|_{\text{sum}}$ versus the number of iterations for both $D \in Đ_\infty$ and $D \in Đ_2$. Note the different scalings of the axes.

$7,075$ and $n = 200^2 = 40,000$, so the problem is highly underdetermined. We use the relative noise level $\|\epsilon\|_2/\|Ax^{\text{exact}}\|_2 = 0.01$. Moreover, we use 5×5, 10×10 and 20×20 patches and corresponding dictionary matrices $D^{(5)}$, $D^{(10)}$, and $D^{(20)}$ in $Đ_2$ of size 25×100, 100×300, and 400×800, respectively. Examples of the dictionary elements are shown in the bottom row of Fig. 4.4.

We first investigate the reconstruction's sensitivity to the choice of λ in the dictionary learning problem and the parameters μ and δ in the reconstruction problem. To simplify the notation of (4.10) we define $\tau = \mu/q$. It follows from the optimality conditions of (4.10) that $\alpha^\star = 0$ is optimal when $\tau \geq \tilde{\tau} = \frac{1}{m}\|(I \otimes D^{\mathrm{T}})\Pi A^{\mathrm{T}} b\|_\infty$ and hence we choose $\tau \in [0, \tilde{\tau}]$. Large values of τ refer to the case where the sparsity prior is strong and the solution is presented with too few dictionary elements. On the other hand if τ is small and a sufficient number of dictionary elements are included, the reconstruction error worsens only slightly when τ decreases. In the next chapter we show that we may, obtain reasonable reconstructions even with $\tau = 0$.

To investigate the effect of regularization parameters λ and τ, we first perform experiments with $\delta = 0$ corresponding to no image prior. The quality of a solution x is evaluated by the reconstruction error (RE) (2.14) shown as contour plots in Fig. 4.7. The reconstruction error is smaller for larger patch sizes, and also less dependent on the regularization parameters λ and τ. The smallest reconstruction errors are obtained in all dictionary sizes for $\lambda \approx 3$.

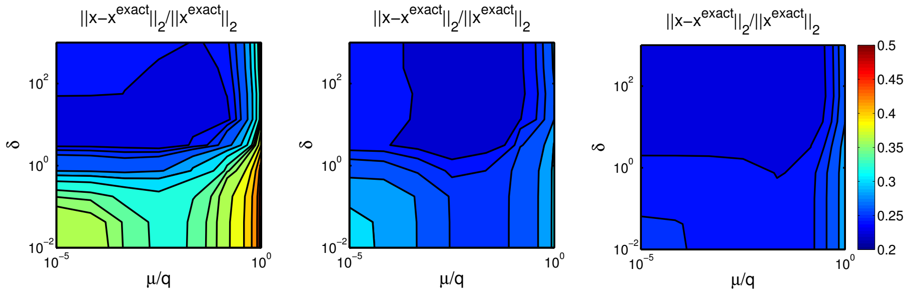
Let us now consider the reconstructions when $\delta > 0$ in order to reduce block artifacts. Figure 4.8 shows contour plots of the reconstruction errors versus τ



(a) 5 × 5 patches (b) 10 × 10 patches (c) 20 × 20 patches

Figure 4.7: Contour plots of the reconstruction error RE (2.14) versus λ and $\tau = \mu/q$.

and δ, using a fixed $\lambda = 3.16$. It is no surprise that introducing δ acts as a regularizer that can significantly improve the reconstruction. Sufficiently large values of δ yield smaller reconstruction errors. In consistence with the results from Fig. 4.7 the reconstruction errors are smaller for 10 × 10 and 20 × 20 patch sizes than for 5 × 5 patches. For larger patch sizes (which allow for capturing more structure in the dictionary elements) the reconstruction error is quite insensitive to the choice of δ and τ. The contour plots in Fig. 4.8 suggest that with our problem specification, we should choose $\delta \geq 1$.



(a) 5 × 5 patches (b) 10 × 10 patches (c) 20 × 20 patches

Figure 4.8: Contour plots of the reconstruction errors RE (2.14) versus $\tau = \mu/q$ and δ for a fixed $\lambda = 3.16$.

The approximation error i.e., $\|P_{\mathcal{C}}(x^{\text{exact}}) - x^{\text{exact}}\|_2$, as well as the reconstruction errors are listed in Table 4.1. These errors show that how well we can represent the exact image in the cone defined by the dictionary, as well as how well we can find a solution as close as possible to this representation, i.e., $P_{\mathcal{C}}(x^{\text{exact}})$.

As can be seen in Fig. 4.5, the MAE for $\lambda = 3.16$ is quite similar for $D^{(5)}$ and $D^{(10)}$ while it is higher for $D^{(20)}$, which leads to a higher representation error using the dictionary with 20 × 20 patches, however the sparse approximation solution error ($\|P_{\mathrm{C}}(x^{\mathrm{exact}}) - x\|_2$) is smaller for 20 × 20 patches using a smaller over representation factor.

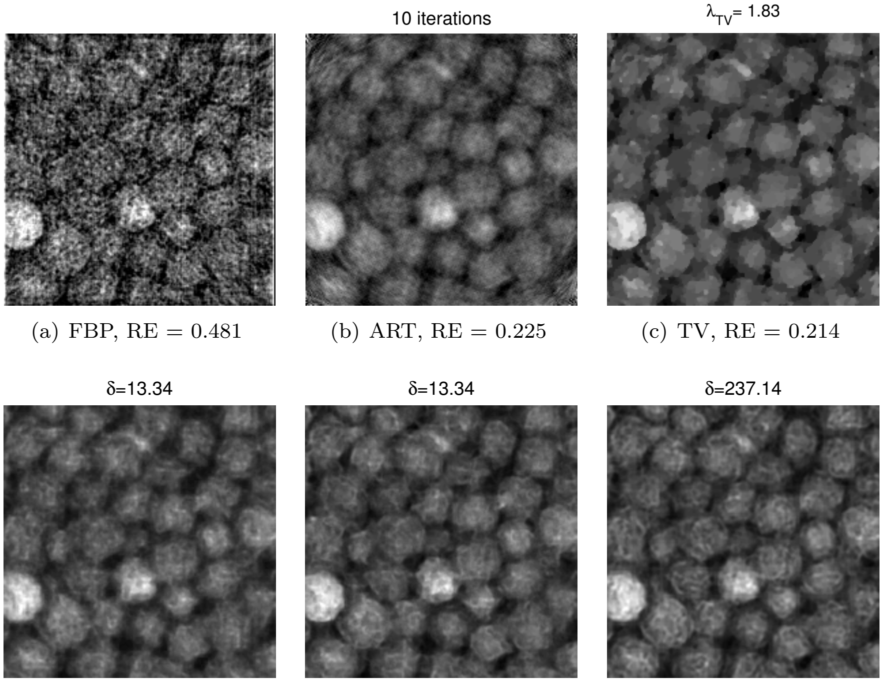
Table 4.1: The corresponding errors for the reconstruction and the best representation of the exact image x^{exact} in the cone defined by the dictionary.

	$\|P_{\mathrm{C}}(x^{\mathrm{exact}}) - x^{\mathrm{exact}}\|_2$	$\|P_{\mathrm{C}}(x^{\mathrm{exact}}) - x\|_2$	$\|x - x^{\mathrm{exact}}\|_2$
5×5	7.70	11.98	15.27
10×10	7.46	12.04	14.98
20×20	9.60	10.54	15.37

Finally, in Fig. 4.9 we compare our reconstructions with those computed by means of filtered back projection (FBP), the algebraic reconstruction technique (ART), and TV regularization. We used the Shepp-Logan filter in `iradon`. To be fair, the TV regularization parameter and the number of ART iterations were chosen to yield an optimal reconstruction. Note that the TV solution for this tomographic scenario and test image is the same as the solution given in Fig. 2.5 Chapter 2.

- The FBP reconstruction contains the typical artifacts associated with this method for underdetermined problems, such as line structures.
- The ART reconstruction – although having about the same RE as our reconstruction – is blurry and contains artifacts such as circle structures and errors in the corners.
- The TV reconstruction has the typical "cartoonish" appearance of TV solutions and hence it fails to include most of the details associated with the texture; the edges of the pepper grains are distinct but geometrically somewhat un-smooth.
- Our reconstructions, while having about the same RE as the TV reconstruction, include more texture and some of the details from the exact image (but not all) are recovered, especially with $D^{(20)}$. Also the pepper grain edges resemble more the smooth edges from the exact image.

We conclude that our dictionary-based reconstruction method appears to have an edge over the other three methods.



(a) FBP, RE = 0.481 (b) ART, RE = 0.225 (c) TV, RE = 0.214

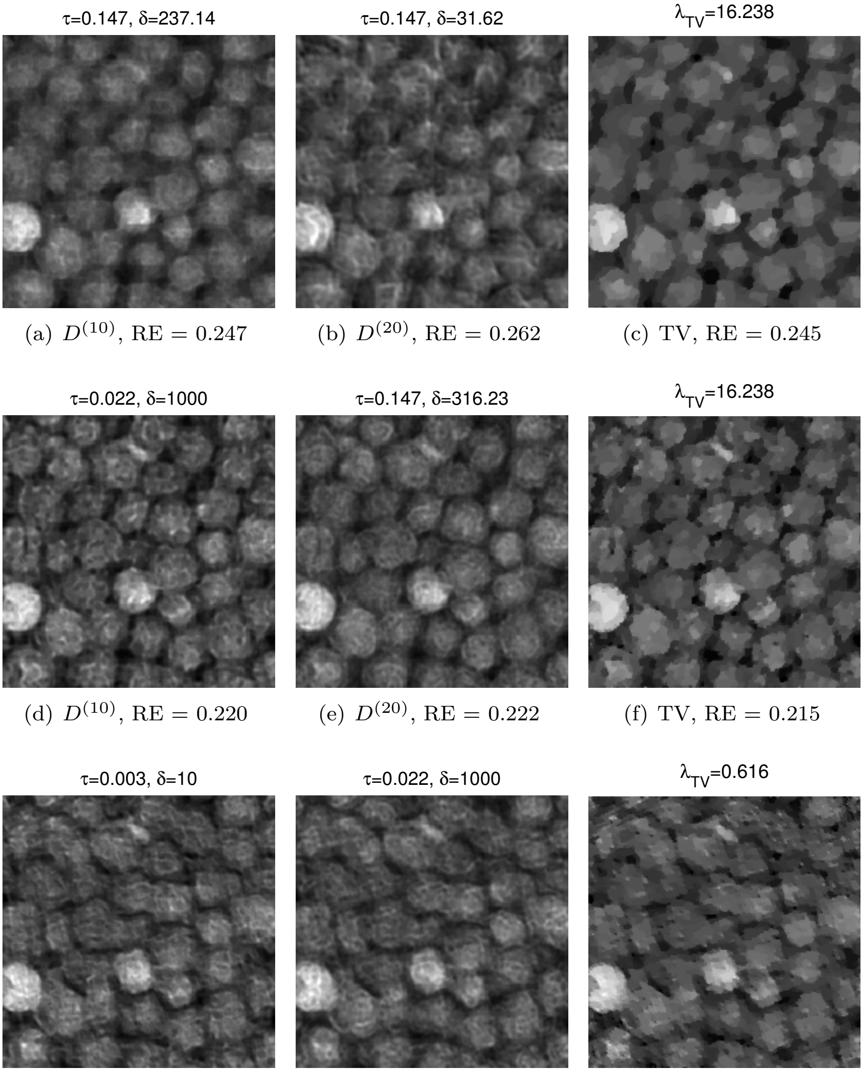
(d) 5×5, RE = 0.224 (e) 10×10, RE = 0.220 (f) 20×20, RE = 0.226

Figure 4.9: Reconstructions for different patch sizes, with $D \in \text{Đ}_2$, $\lambda = 3.16$, and $\tau = 0.022$, compared with the FBP, ART and TV solutions. RE denotes the reconstruction error (2.14).

4.4.4 Studies of Sensitivity to Noise and Limited-Angle Data

To further study the performance of our algorithm, in this section we consider reconstructions based on (4.10) with more noise in the data, and with projections within a limited range. The first two sets use 25 and 50 projections with uniform angular sampling in $[0°, 180°]$ and with relative noise level = 0.05, i.e., a higher noise level than above. For our highly underdetermined problems we know that both filtered back projection and algebraic iterative techniques give unsatisfactory solutions, and therefore we only compare our method with TV. As before the regularization parameters λ and τ are chosen from numerical experiments such that a solution with the smallest error is obtained.

The reconstructions are shown in the top and middle rows of Fig. 4.10. The



(a) $D^{(10)}$, RE = 0.247 (b) $D^{(20)}$, RE = 0.262 (c) TV, RE = 0.245

(d) $D^{(10)}$, RE = 0.220 (e) $D^{(20)}$, RE = 0.222 (f) TV, RE = 0.215

(g) $D^{(10)}$, RE = 0.255 (h) $D^{(20)}$, RE = 0.261 (i) TV, RE = 0.246

Figure 4.10: The left and middle columns show our reconstructions with $\lambda = 3.16$ using $D^{(10)}$ and $D^{(20)}$, respectively; the right column shows the TV reconstructions. Top and middle rows: $N_\mathrm{p} = 25$ and $N_\mathrm{p} = 50$ projections in $[0°, 180°]$ and relative noise level 0.05. Bottom row: $N_\mathrm{p} = 25$ projections in $[0°, 120°]$ and relative noise level 0.01.

reconstruction errors are still similar across the methods. Again, the TV reconstructions have the characteristic "cartoonish" appearance while the dictionary-based reconstructions retain more the structure and texture but have other artifacts – especially for $N_{\mathrm{p}} = 25$. We also note that these artifacts are different for the two different dictionaries.

The third set uses 25 projections uniformly distributed in the limited range $[0°, 120°]$ and with relative noise level 0.01. In this case the TV reconstructions display additional artifacts related to the limited-angle situation, while such artifacts are somewhat less pronounced in the reconstructions by our algorithm.

4.4.5 A Large Test Case

We finish the numerical experiments of this chapter with a verification of our method on two larger test problems that simulate the analysis of microstructure in materials science. Almost all common metals, and many ceramics, are polycrystalline, i.e., they are composed of many small crystals or grains of varying size and orientation, and the variations in orientation can be random. A grain boundary is the interface between two grains. It is of particular interest to study how these boundaries — the interfaces between grains — change over time, for instance when the material is exposed to external stimuli such heat or pressure. Here we assume that priors of the grain structure are available in the form of training images.

The simulated data was computed using images of steel and zirconium grains. The steel microstructure image from [55] is of dimensions 900×1280 and the zirconium grain image (produced by a scanning electron microscope) is 760×1020. More than $50,000$ patches are extracted from these images to learn dictionaries $D^{(20)} \in Đ_2, Đ_\infty$ of size 400×800. To avoid doing inverse crime, we obtain the exact images of dimensions 520×520 by first rotating the high-resolution image and then extracting the exact image. The high-resolution images and the exact images are shown in Fig. 4.11.

We consider a parallel-beam tomographic scenario with $N_{\mathrm{p}} = 50$ projections corresponding to 50 uniformly distributed projections in $[0°, 180°]$, leading to $m = 36,750$ data values. We add Gaussian white noise with relative noise level 0.01 and compute reconstructions by our method as well as the TV method; these reconstruction are shown in Fig. 4.12. All regularization parameters were chosen to give the best reconstruction as measured by the RE, and we note that the reconstruction errors are dominated by the error coming from the regularization of the noisy data; the approximation errors $\|P_{\mathsf{C}}(x^{\text{exact}}) - x^{\text{exact}}\|_2 / \|x^{\text{exact}}\|_2$ are of the order 0.03 and 0.05 for the steel and zirconium images, respectively.

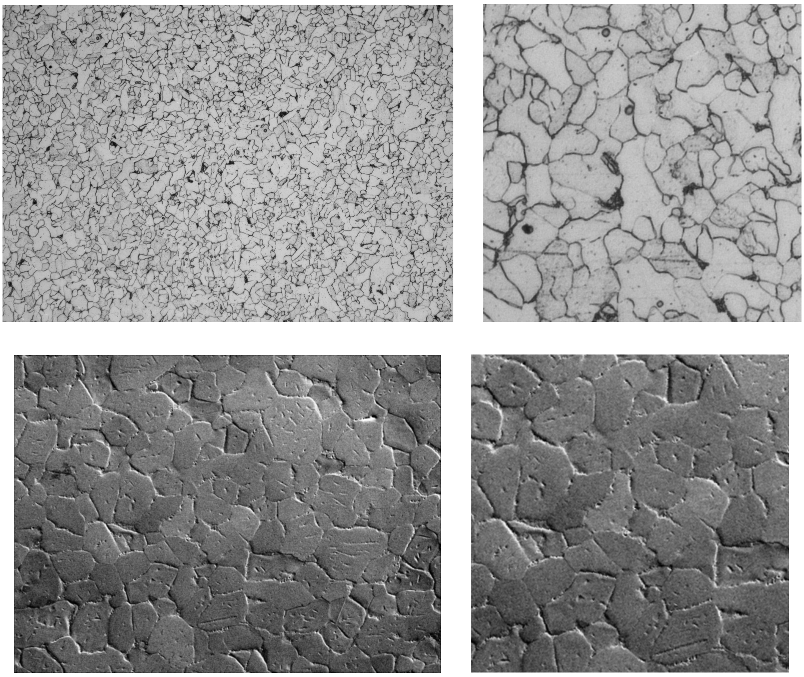
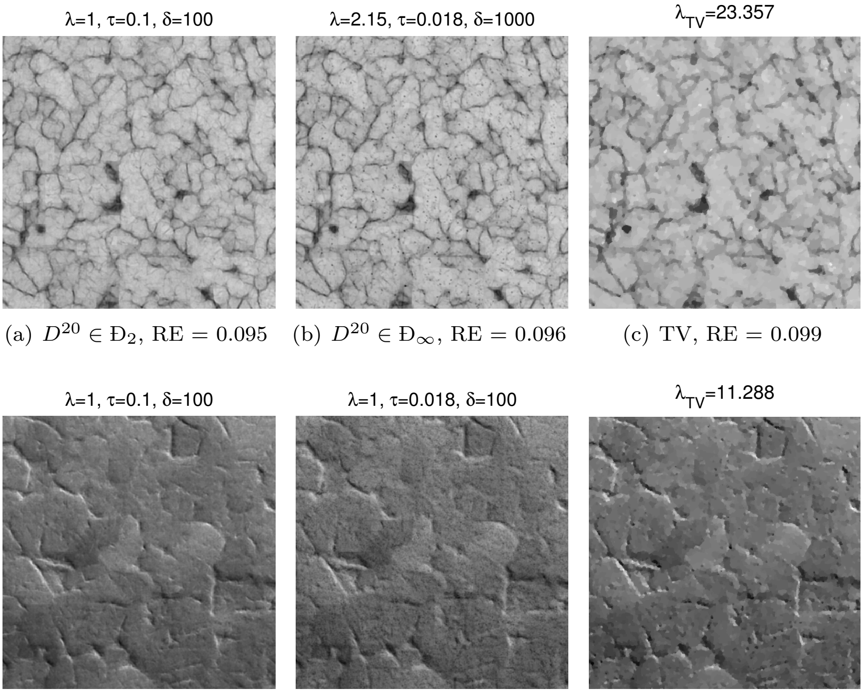


Figure 4.11: Left: high-resolution images of steel micro-structure [55] (top) and zirconium grains (bottom) used to generate the training images. Right: the corresponding exact images of size 520 × 520.

We see that our algorithm, for both $Đ_2$ and $Đ_\infty$, performs better than the TV method for recovering the textures and, in particular, the grain boundaries that are of interest here. Our reconstructions for $Đ_\infty$ have the sharpest grain boundaries, but some small black "dots" have appeared which are not present for $Đ_2$; in both cases the images are suited for post-processing via image analysis.

As expected, the TV reconstructions exhibit "cartoonish" artifacts, and for the steel grains the black interfaces tend to be too thick and they are not so well resolved. Our method, for both $Đ_2$ and $Đ_\infty$, recovers better the grain interfaces that are of interest here. We obtain the sharpest interfaces for $Đ_\infty$ but some small black "dots" have appeared which are not present for $Đ_2$; in both cases the images are suited for postprocessing via image analysis.



(a) $D^{20} \in \text{Đ}_2$, RE = 0.095 (b) $D^{20} \in \text{Đ}_\infty$, RE = 0.096 (c) TV, RE = 0.099

(d) $D^{20} \in \text{Đ}_2$, RE = 0.146 (e) $D^{20} \in \text{Đ}_\infty$, RE = 0.158 (f) TV, RE = 0.137

Figure 4.12: Reconstructions of the 520×520 images by our method (left and middle) and by the TV method (right). Top: steel microstructure. Bottom: zirconium grains.

4.5 Summary

In this chapter we describe and examine an algorithm that incorporates training images as priors in computed tomography (CT) reconstruction problems. This type of priors can be useful in low-dose CT where we are faced with underdetermined systems of equations.

Our algorithm has two stages. In the first stage a learned dictionary from a set of training images is computed using a regularized nonnegative matrix factorization (NMF). In the second stage, via a regularized least squares fit a nonnegative reconstruction lying in the cone defined by the dictionary elements is computed; the reconstruction is sparse with respect to the dictionary. Hence, regularization is obtained by enforcing that the reconstruction is within the range of the dictionary elements and by the sparsity constraint.

The proposed algorithm works with non-overlapping image patches; the same dictionary is used for all patches, and the blocking artifacts are minimized by an additional regularization term. This reduces the computational complexity, compared to all other proposed algorithms that apply a dictionary-based regularization based on overlapping patches around every pixel in the image.

Our algorithm includes several regularization parameters. In the first stage a parameter is used to control the sparsity in the NMF, and in the second stage one parameter to control the sparsity of the representation in the dictionary and another parameter to avoid blocking artifacts are used. A series of numerical experiments with noisy data and without committing inverse crime are performed, where the interplay between these parameters and the computed reconstructions are demonstrated, and it is shown that the reconstructions are not very sensitive to these parameters.

In conclusion the training images can be useful as a strong prior for regularization of low-dose CT problems, through a sparse representation in a nonnegative dictionary learned from the training images. Our reconstructions are (not surprisingly) superior to those computed by classical methods such as filtered back projection and algebraic iterative methods, and they are competitive with total variation (TV) reconstructions. Specifically, in our test problems our algorithm tends to be able to include more texture and also produces edges whose location is more correct.

Chapter 5

Studies of Sensitivity

In Chapter 4 we formulated and implemented a two-stage algorithm for using training images in tomographic reconstruction, in which we first form a dictionary from patches extracted from the training images and then by means of a sparsity prior on all the non-overlapping patches in the image, the dictionary is used for finding a tomographic solution in the cone defined by the dictionary. Being successful in incorporating the desirable features of the training image in the dictionary prior, leads to a superior solution comparing to classical tomographic reconstruction methods.

There is no guarantee that the training images have the correct orientation or scale when trying to solve the image reconstruction problem for an unknown object, which is often neglected when using learned dictionary approaches in tomographic image reconstruction, e.g., see [109, 116]. On the other hand in Chapter 4 we have been working under the assumption that the representation in the learned dictionary is nonnegative and that it is sparse and the solution lies in the cone spanned by the learned dictionary elements. Imposing both non-negativity and a sparsity constraint on the representation vector and only searching for solutions in the cone spanned by the dictionary elements are strong assumptions in the reconstruction formulation. Therefore we are interested to investigate how relaxing this assumption affects the reconstructed solution.

In this Chapter, we continue the work initiated in Chapter 4, in order to increase an understanding of the model's limitations and capabilities. In Sections 5.1 and 5.2 we use numerical examples to demonstrate and quantify the behavior of our two-stage algorithm when we encounter uncertainty in the tomographic reconstruction stage such as model assumptions and changes in the scale and orientation of the object.

The main contributions of this chapter are:

- The robustness of our problem formulation in Chapter 4 is further studied. The influence of relaxing the representation in the cone defined by the dictionary as well as the constraints in the problem formulation is explored.
- The sensitivity and robustness of our algorithm to scale and rotation variances with various computational tests are analyzed.
- Algorithms to detect rotation and scale of the image, prior to the reconstruction step, from the sinogram of the tomographic measurement data are proposed.

All experiments are run in MATLAB (R2014a) on a 64-bit Linux system. We use an implementation of the ADMM algorithm presented in Section 4.2 to obtain a dictionary and the reconstruction problems are solved using the software package TFOCS version 1.3.1 [8]. Our computational test setup is identical to the numerical setup described in the intro of Section 4.4.

5.1 Simplifying the Reconstruction Problem

In this section we perform an empirical study of the reconstruction's robustness to the assumptions in the reconstruction step and that the solution is a conic combination of dictionary elements and their effects on the success of reconstruction.

5.1.1 The Constraints of The Reconstruction

We have been working under the assumption that $\alpha \geq 0$ and that it is sparse. Imposing both non-negativity and a l_1-norm constraint on the representation vector α are strong assumptions in the reconstruction formulation.

If we drop the non-negativity constraint in the image reconstruction problem, then (4.10) can be reformulated as a constrained least squares problem:

$$\min_{\alpha} \frac{1}{2} \left\| \begin{pmatrix} \frac{1}{\sqrt{m}} A\, \Pi^{\mathrm{T}}(I \otimes D) \\ \frac{\delta}{\sqrt{\vartheta}} L\, \Pi^{\mathrm{T}}(I \otimes D) \end{pmatrix} \alpha - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2 \qquad \text{s.t.} \quad \|\alpha\|_1 \leq \gamma. \qquad (5.1)$$

where $\gamma > 0$ and $M(M/p-1) + N(N/r-1) = \vartheta$.

Alternatively we can relax the parameter τ. This is motivated by the results in Section 4.4.3 which suggest that for sufficiently large λ, δ and patch sizes, the reconstruction error is almost independent of τ as long as it is small. When $\tau = 0$, we exclude the l_1-norm constraint on the representation vector α, and (4.10) reduces to a nonnegative constrained least square problem:

$$\min_{\alpha} \frac{1}{2} \left\| \begin{pmatrix} \frac{1}{\sqrt{m}} A\, \Pi^{\mathrm{T}}(I \otimes D) \\ \frac{\delta}{\sqrt{\vartheta}} L\, \Pi^{\mathrm{T}}(I \otimes D) \end{pmatrix} \alpha - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2 \qquad \text{s.t.} \qquad \alpha \geq 0. \qquad (5.2)$$

We use the peppers test problem (Fig. 4.3) with 25 projections and relative noise level 0.01. We solve problem (5.1) for 10×10 patches and corresponding dictionary matrix $D^{(10)}$, in Đ of size 100×300, which resulted in the smallest reconstruction error when solving (4.10) (cf. Fig. 4.9). Likewise we choose 10×10 and 20×20 patch sizes and $D^{(10)}$ and $D^{(20)} \in$ Đ$_2$ of size 400×800 to solve the nonnegativity constrained least square problem (5.2). Figures 5.1 and 5.2 show reconstructions when solving the two above problems (5.1) and (5.2), respectively.

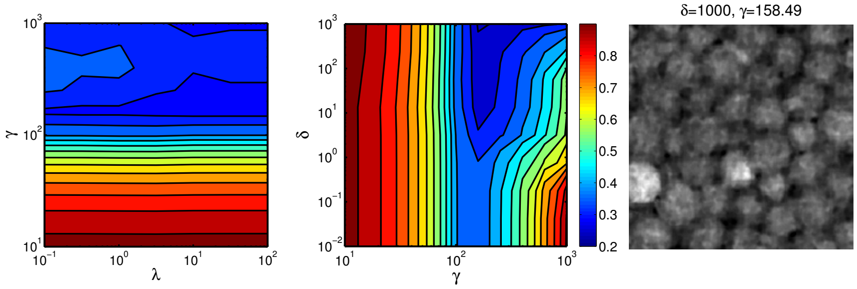


Figure 5.1: Contour plots of the reconstruction error RE for problem (5.1). Left: RE versus λ and γ when $\delta = 0$. Middle: RE versus γ and δ with fixed $\lambda = 10$. Right: The best reconstruction with RE = 0.243.

There are two difficulties with the reconstructions computed via (5.1). The lack of a nonnegativity constraint on α can lead to negative pixel values in the

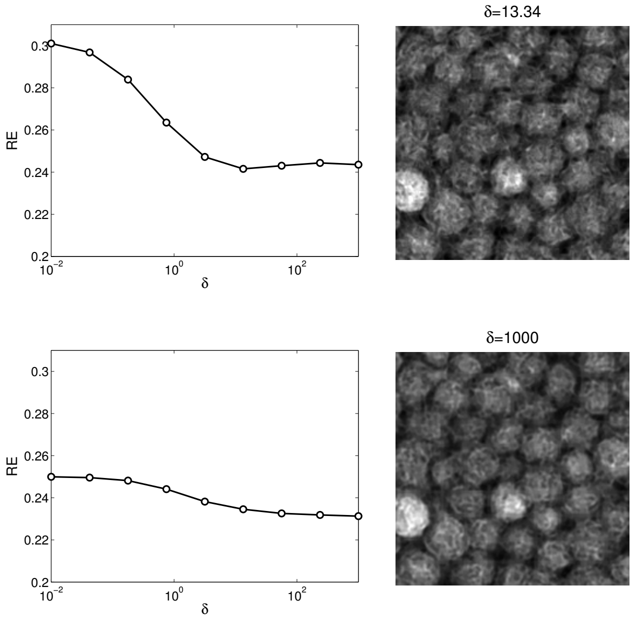


Figure 5.2: Left: plots of reconstruction error versus δ for problem (5.2), using fixed $\lambda = 3.16$ and $\tau = 0$. Right: the best reconstructions with RE $= 0.242$ and RE $= 0.231$. The top and bottom rows correspond to patch sizes 10×10 and 20×20, respectively.

reconstruction, and this is undesired because it is nonphysical and it leads to a larger reconstruction error. Also, as can be seen in Fig. 5.1, the reconstruction is very sensitive to the choice of the regularization parameter γ, it must be sufficiently large to allow the solution to be represented with a sufficient number of dictionary elements, and it should be carefully chosen to provide an acceptable reconstruction. This shows that the non-negatively constraint plays an extra role of regularization.

The solution to problem (5.2) for a 20×20 patch size, compared to the solution shown in Fig. 4.9, is not significantly worse both visually and in terms of reconstruction error. This suggests that using the dictionary obtained from (4.1) with a proper choice of λ and patch size and a nonnegativity constraint may be sufficient for the reconstruction problem, i.e., we can let $\tau = 0$. While this seems to simplify the problem – going from (4.10) to (5.2) – it does not significantly simplify the computational optimization problem, since the l_1-norm constraint is handled by simple thresholding in the software; but it helps us to get rid of a parameter in the reconstruction process. Also, when the l_1-norm constraint is omitted, additional care is necessary when choosing λ and the patch sizes to avoid introducing artifacts or noise in the reconstruction.

5.1.2 Importance of the Representation in the Cone Defined by the Dictionary

Our formulation in 4.10 enforces that the solution is an exact representation in the dictionary, searching for a reconstruction in the cone spanned by the dictionary, i.e., $\Pi x = (I \otimes D)\alpha$, is a very strong prior. Let us construct our tomographic reconstruction formulation in a different way.

Here to incorporate our dictionary prior, we consider $\Pi x \approx (I \otimes D)\alpha$ rather than assuming that $\Pi x = (I \otimes D)\alpha$, i.e., x does not have an exact representation in the dictionary and instead it is close to a solution that lies in the space spanned by the dictionary elements. Thus we consider the following reconstruction problem:

$$\min_{x,\alpha} \frac{1}{2m} \|Ax - b\|_2^2 + \delta^2 \psi(x) + \beta \|x - \Pi^{\mathrm{T}}(I \otimes D)\alpha\|_2^2, \qquad (5.3)$$
$$\text{s.t.} \quad x \geq 0, \;\; \alpha \geq 0,$$

where the function $\psi(\cdot)$ is defined in equation (4.9). For simplicity of this study, we dropped the sparsity prior $\mu/q\|\alpha\|_1$ from (4.10) in (5.3). This is motivated by the results from Sections 4.4.3 and 5.1.1 that for sufficiently large values of δ and patch sizes, the reconstruction error is almost independent of μ as long as it is small.

The problem (5.3) can equivalently be written as:

$$\min_{x,\alpha} \frac{1}{2} \left\| \begin{pmatrix} \frac{1}{\sqrt{m}} A & 0 \\ \frac{\delta}{\sqrt{\vartheta}} L & 0 \\ \sqrt{2\beta} I & -\sqrt{2\beta}\Pi^{\mathrm{T}}(I \otimes D) \end{pmatrix} \begin{pmatrix} x \\ \alpha \end{pmatrix} - \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix} \right\|_2^2 \qquad (5.4)$$
$$\text{s.t.} \quad \begin{pmatrix} x \\ \alpha \end{pmatrix} \geq 0.$$

Note the similarity of the (5.4) to the generic nonnegative least squares problem formulation (4.6).

The regularization parameter β in (5.3) and (5.4) balances the fitting term and the regularization induced by the dictionary. The larger the β, the more weight is given to minimization of $\|x - \Pi^{\mathrm{T}}(I \otimes D)\alpha\|_2^2$, while for small β more weight is given to fitting the noisy data, resulting in solutions that are less regular (we obtain the problem (4.6) and the naive solution when $\beta = 0$). We expect that for sufficiently large β we obtain solutions not far from solutions obtained with the exact dictionary approach (i.e., from problem (4.10)).

Consider the tomographic problem from Section 4.4.3 with $N_{\mathrm{p}} = 25$ projections and 1% additive relative noise. Moreover, we use the 20 by 20 patch dictionary $D^{(20)} \in Ð_2$ of size 400×800.

The reconstructions for various values of β are shown in Fig. 5.3; they are similar across the larger values of β, however pronounced artifacts have appeared for small values of β from over-fitting the noisy data and reducing the weight on the dictionary prior. As can be see in Fig. 5.3, with larger values of β and less weight given to fit the tomographic data, the solution tends to be smooth.

We define the relative dictionary misfit by $\|\Pi^{\mathrm{T}}(I \otimes D)\alpha - x\|_2/\|x\|_2$. Plots of the reconstruction error and the relative dictionary misfit are given in Fig. 5.3. As illustrated by these plots the reconstruction error decreases and then levels off for large values of β, e.g., RE= 0.2238 for $\beta = 1000$. The relative dictionary misfit exponentially decreases for large values of β, indicating that the approximation $x \approx \Pi^{\mathrm{T}}(I \otimes D)\alpha$ is almost exact for β sufficiently large.

By considering the problem formulation (5.4) instead of (4.10) we are introducing β as a new regularization parameter, which needs further investigations to find a suitable value for it. In general relaxing $\Pi x = (I \otimes D)\alpha$ does not give an advantage, i.e., approximating a solution by $\Pi x \approx (I \otimes D)\alpha$ does not particularly improve the reconstruction quality, and one can compute a good reconstruction as a conic combination of the dictionary elements.

5.2 Rotation and Scale

It may be crucial to include the acts of rotation and geometric scaling of the training images when using the learned dictionaries in the tomographic reconstruction, where there is no guarantee that the training set will have the correct orientation and geometric scaling. Rotation and scaling are two unknown parameters that are needed to be considered in the reconstruction formulation and hence it is advantageous to determine the correct rotation and scaling parameters or obtain a scale and rotation invariant dictionary prior to the reconstruction process.

Invariance to rotation and scale are desirable in many practical applications. For example, in pattern recognition the widely used scale-invariant feature transform

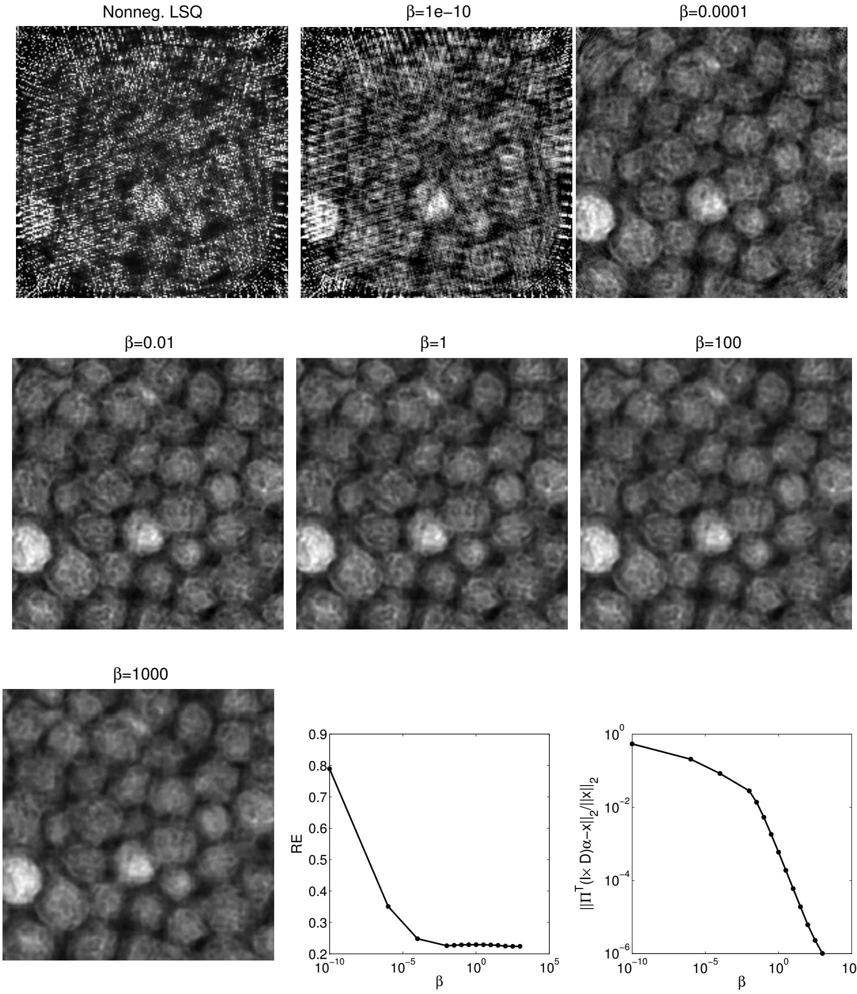


Figure 5.3: Reconstruction results from solving (5.3) with $\beta \in [10^{-10}, 1000]$. Bottom: Middle: plot of reconstruction error versus β. Right: plot of the relative dictionary misfit versus β.

(SIFT) algorithm successfully detects the training image under changes in image scale, noise and rotation [77]. The paper [52] presents a face recognition method which uses features that are extracted from the log-polar images which are invariant to scale and rotation. Dictionary learning methods that are independent of orientation and scale, with applications in classification of images or clustering, have also been recently developed. A shift, scale and rotation invariant dictionary learning method for multivariate signals and a hierarchical dictionary learning method for invariant classification have been proposed in [4] and [3] respectively. These methods learn a dictionary in a log-polar domain. In the paper [23] a rotation and scale invariant clustering algorithm using dictionaries is presented where the image features are extracted in the Radon transform domain.

To the best of our knowledge, no study has investigated and explored the role played by scale and rotation in tomographic reconstruction approaches using dictionaries.

5.2.1 Sensitivity to Scale

It is possible that the scale of the training images differ from the one we would like to achieve in the reconstruction process. While the dictionary learning approaches in image processing problems such as image denoising and image restoration do not directly suffer from scale issues, it has been explored that with the existence of multi-scale features in images, using multi-scale dictionaries would result in superior reconstructions compared to single-scale dictionaries (see, e.g., [81, 82, 89]). Such dictionaries enforce sparsity at multiple scales.

One idea is to train the dictionary on many possible scaling of the training images, this approach is computationally expensive in both the learning and reconstruction stage. Inspired by a multi-scale dictionary, first we investigate if a generic dictionary of smaller patches (with a fixed patch size) or a learned dictionary from different scaling of the training images could result in a “better” reconstruction for an off-scale image.

If the image is represented by a function X then we say $\bar{X}$ is a scaled copy of X with scale factor η if $\bar{X}(u,v) = X(\eta u, \eta v)$. We look at three test examples that we call “peppers”, “matches”, and “binary” images. The binary test image – a random image with binary pixel values – is generated by the `phantomgallery` function from the MATLAB package AIR Tools [47]. The exact test images of size 200 × 200 with the scale factor $\eta = 1.5$ are shown in Fig. 5.4.

To generate different dictionaries for our tests, we consider a large training

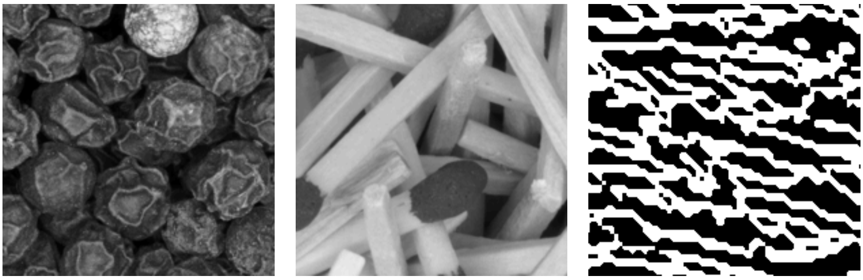


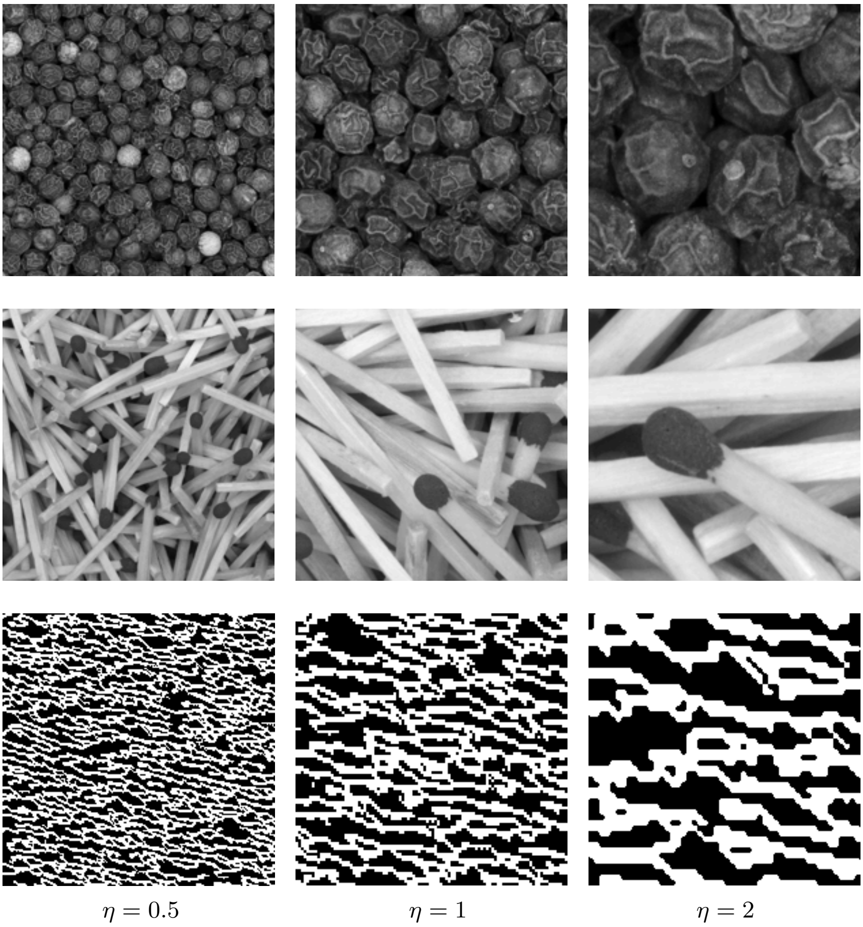
Figure 5.4: The 200×200 exact images x^{exact} with scale factor $\eta = 1.5$. Left: peppers, middle: matches, and right: binary test images.

image for each test case and we denote its scale to be the reference scale (scale 1). Knowing that the scale of the training image is different from the image we want to reconstruct, we can argue that we need a greater over-representation factor to learn a generic dictionary and be able to represent off-scale images. Hence for $\eta = 1$ we learned dictionaries of 5×5 and 10×10 patch sizes with over-representation factors of 10 and 5, respectively, i.e., $D^{(5)} \in \mathbb{R}^{25\times 250}$ and $D^{(10)} \in \mathbb{R}^{100\times 500}$. We also learn a 20×20 patch dictionary of size 400×1200 in which the training patches are chosen randomly from training images that are scaled by a factor of 0.5, 1 and 2. Figure 5.5 shows examples of 200×200 sub-images of our three training test images with scale factors $\eta = 0.5, 1, 2$. The learned multi-scale dictionaries with 20×20 patches and generic dictionaries with 10×10 patches and $\lambda = 1$ are given in Fig. 5.6. We clearly see the multi-scale features of the dictionary with 20×20 patches.

We solve the reconstruction problem (4.10) using the exact images given in Fig. 5.4. We choose $N_{\text{p}} = 25$, projections with uniformly distributed angles in $[0°, 180°]$, $N_{\text{r}} = 283$ and 1% additive noise level. In Fig. 5.7 we compare our reconstructions with those computed by the multi-scale dictionary with $20 \times 20\,(\eta = 0.5,\ 1,\ 2)$ patches and the generic dictionaries of scale factor $\eta = 1$ with 5×5 and 10×10 patch sizes. To be fair, the regularization parameters τ and δ were chosen to yield an optimal reconstruction in terms of the reconstruction error.

The reconstructions shown in the right column of Fig. 5.7 show no particular advantage in terms of reconstruction errors when using a multi-scale dictionary (learned from patches of various scale) over a sufficiently large generic dictionary of smaller patch sizes, with the reconstructions shown in left and middle columns of Fig. 5.7.

Now to better understand the role played by the scale parameter η, we solve the



$\eta = 0.5$ $\eta = 1$ $\eta = 2$

Figure 5.5: Examples of 200×200 sub-images of the training test images with scale factors $\eta = 0.5$, 1, 2. Top: peppers, middle: matches, and bottom: binary test images.

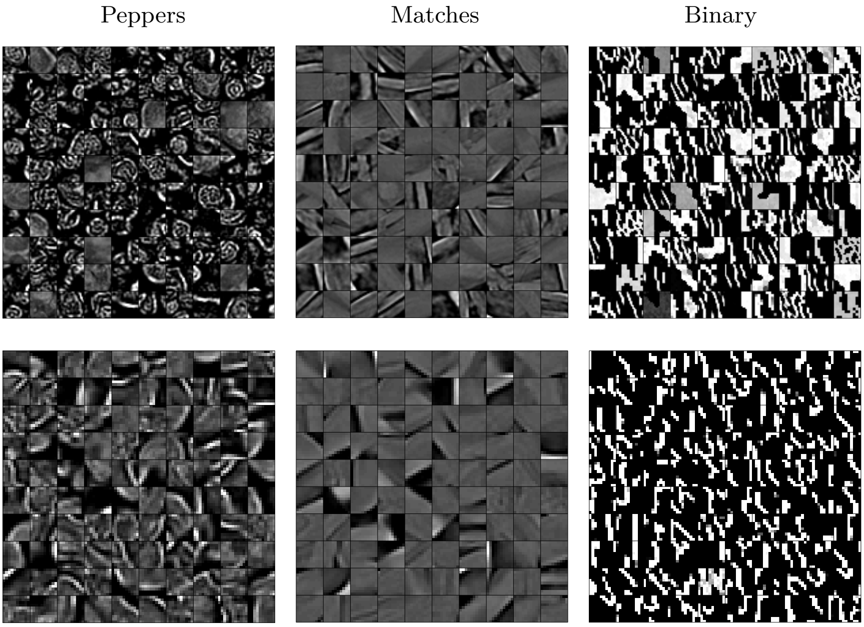
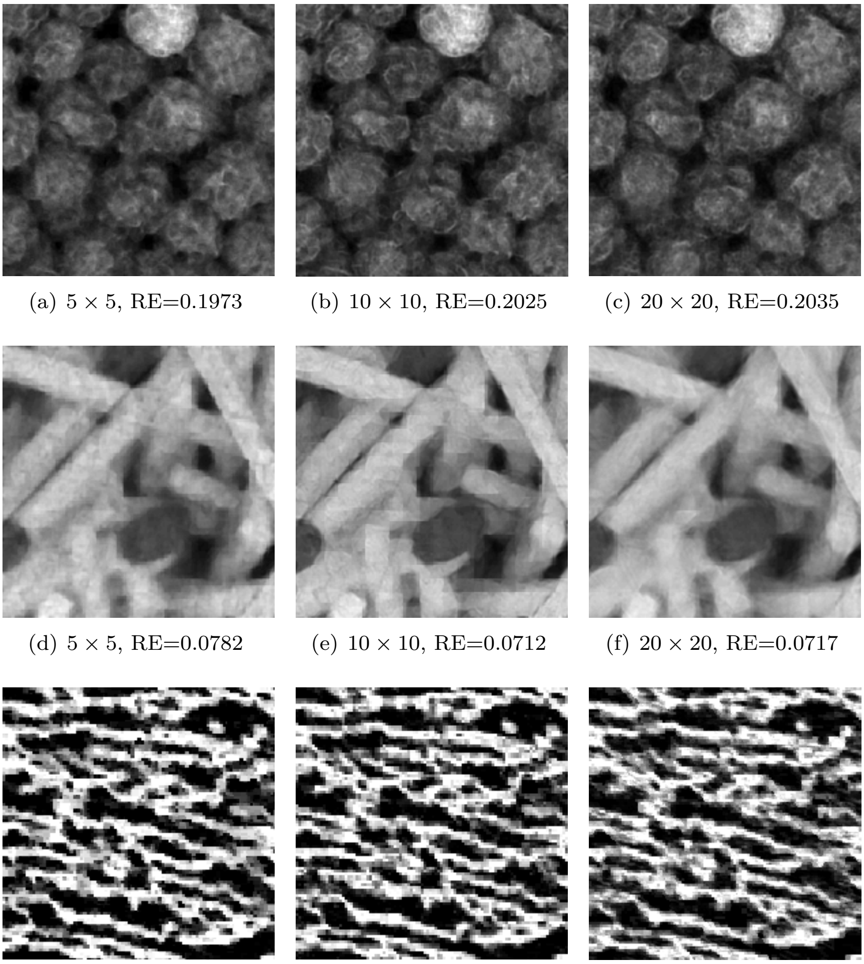


Figure 5.6: Top: Examples of the multi-scale dictionary elements (images) with 20×20 patches and $\lambda = 1$. Bottom: Examples of the generic dictionary elements (images) with scale factor 1, 10×10 patches and $\lambda = 1$.

peppers tomographic reconstruction problem from the Section 4.4.3 with the exact image given in the Fig. 4.3 and the matches test problem of size 200×200 where the exact image is given in Fig. 5.8. The scale factor of these test images is assumed to be $\eta = 1$. We use $N_{\mathrm{p}} = 25$ projections with angles in $[0°, 180°]$ and relative noise level 0.01. We keep the size of the patches 10×10 and the dictionary size $s = 500$, and we learn 11 new dictionaries of size 100×500 where the scale factor of the training images η is varied in the interval [0.4, 4]. Plots of the reconstruction error versus the scale factor of the training patches, which we learned our dictionaries from, are given in Fig. 5.9. We also plot the structural similarity index measure (SSIM) [114] for measuring the similarity between the reconstructed solution and the exact images in Figures 4.3 and 5.8. Recall that a larger SSIM means a better reconstruction.

Figure 5.9 shows that unless we are looking for a solution with a higher resolution than the training images, i.e., if the scale of the training images are smaller than the desired image that we want to reconstruct, the reconstruction is not very sensitive to the scaling factor, choosing a generic dictionary and sufficiently



(a) 5 × 5, RE=0.1973

(b) 10 × 10, RE=0.2025

(c) 20 × 20, RE=0.2035

(d) 5 × 5, RE=0.0782

(e) 10 × 10, RE=0.0712

(f) 20 × 20, RE=0.0717

(g) 5 × 5, RE=0.4236

(h) 10 × 10, RE=0.4577

(i) 20 × 20, RE=0.4624

Figure 5.7: Reconstructions for the generic and multi-scale dictionaries with different patch sizes (Fig. 5.6), using the exact images given in Fig. 5.4. RE denotes the reconstruction error.

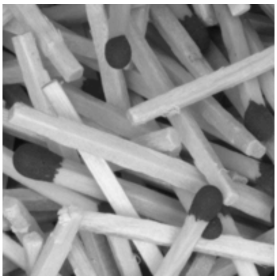


Figure 5.8: The 200×200 matches exact image x^{exact} with scale factor $\eta = 1$.

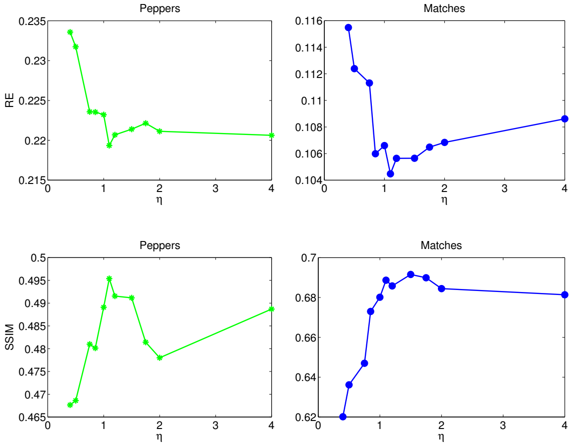


Figure 5.9: Top: left: peppers, and right: matches reconstruction errors versus the scaling factor of dictionaries. Bottom: left: peppers, and right: matches SSIM measures versus the scaling factor of dictionaries.

large number of elements. This is no surprise, one cannot expect to perfectly reconstruct a high resolution image from a dictionary learned from lower resolution training images since some important details of textures and structure are missing in those images.

5.2.2 An Algorithm to Determine Scale

One may think of a preprocessing step to find the appropriate scale of the image before training the dictionary. Then the simplest case is downsizing the training images and learn the dictionary in the right scale or downsizing/shrinking the dictionary images in the right way. One simple way to determine the correct scale is to reconstruct a naive FBP solution and compare the solution with the training images to find the correct scale. The scale can be detected by comparing similar single objects in both images; however the limited tomographic data and presence of noise often result in obtaining unreliable naive solutions where most textures and image structures have disappeared, which makes such an estimation difficult.

Another option is to find scales from the sinogram of the 2D unknown image. Recall that the tomographic data can be represented – for some 2D applications – as a matrix called the sinogram. We denote the sinogram by the matrix S. The 2D Radon transform is graphically represented as the sinogram, which means by the intensity values in the coordinate system of variables $(\text{ţ}, \theta)$. Recall the definition of the Radon transform of a two variable function X from the equation (2.10). By swapping the coordinate system:

$$\begin{aligned} \text{ţ} &= x_1 \cos\theta + x_2 \sin\theta & \text{ş} &= -x_1 \sin\theta + x_2 \cos\theta \\ x_1 &= \text{ţ} \cos\theta - \text{ş} \sin\theta & x_2 &= \text{ţ} \sin\theta + \text{ş} \cos\theta, \end{aligned}$$

the radon transform can be equally expressed as

$$\mathrm{R}_\theta X(\text{ţ}) = \int_{-\infty}^{+\infty} X(\text{ţ}\cos\theta - \text{ş}\sin\theta, \text{ţ}\sin\theta + \text{ş}\cos\theta) d\text{ş},$$

$$(\text{ţ}, \theta) \in (-\infty, \infty) \times [0, \pi).$$

Let $\bar{X}$ be a scaled copy of X with the scaling factor η. Then the Radon transforms of $\bar{X}$ and X are related as follows:

$$\begin{aligned} \mathrm{R}_\theta \bar{X}(\text{ţ}) &= \int_{-\infty}^{+\infty} \bar{X}(\text{ţ}\cos\theta - \text{ş}\sin\theta, \text{ţ}\sin\theta + \text{ş}\cos\theta) d\text{ş} && (5.5) \\ &= \int_{-\infty}^{+\infty} X(\eta\text{ţ}\cos\theta - \eta\text{ş}\sin\theta, \eta\text{ţ}\sin\theta + \eta\text{ş}\cos\theta) d\text{ş} && (5.6) \\ &= \frac{1}{\eta} \mathrm{R}_\theta X(\eta\text{ţ}). && (5.7) \end{aligned}$$

Let us define:

$$M_X = \max_{\text{ţ}, \theta} |\mathrm{R}_\theta X(\text{ţ})|$$

Then for any pair $\bar{X}$ and X related by $\bar{X}(u, v) = X(\eta u, \eta v)$ with $\eta > 0$ the following holds:

$$M_{\bar{X}} = \frac{1}{\eta} M_X.$$

Since from (5.5):

$$\begin{aligned} M_{\bar{X}} &= \max_{\mathfrak{t},\theta} |\mathrm{R}_\theta \bar{X}(\mathfrak{t})| \\ &= \max_{\mathfrak{t},\theta} |\frac{1}{\eta} \mathrm{R}_\theta X(\eta \mathfrak{t})| \\ &= \frac{1}{\eta} \max_{\mathfrak{t},\theta} |\mathrm{R}_\theta X(\eta \mathfrak{t})| \\ &= \frac{1}{\eta} \max_{\mathfrak{t},\theta} |\mathrm{R}_\theta X(\psi)| \quad (\text{if } \psi = \eta \mathfrak{t}) \\ &= \frac{1}{\eta} M_X . \end{aligned}$$

This proof is adopted from [23]. In the sinogram matrix S given by the discretized Radon transform, column indices correspond to discrete values of θ, while row indices correspond to discrete values of $\mathfrak{t}$. Hence M_X is the element-wise maximum of the values in the sinogram matrix.

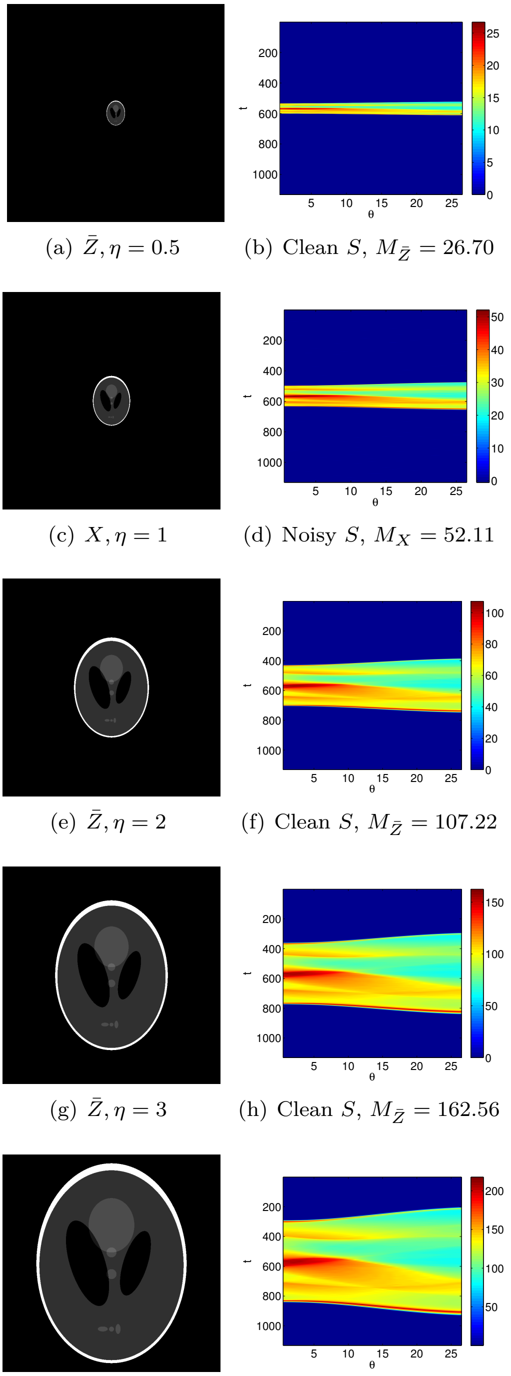
Consider an unknown image X, where a noisy sinogram of X is available. We can make an artificial sinogram of a sub-image of the training image with the same tomographic setting/scenario. We can claim that if the training image $\bar{Z}$ with a similar dimension as X is given, then we can compute the relative scale factor η by

$$\eta \approx \frac{M_X}{M_{\bar{Z}}} .$$

We emphasize that the practical use of this approach relies on a careful implementation, and use of the Radon transform such that the integrals are correctly evaluated. Matlab's `radon` satisfies this requirement.

For a test problem we use the 200×200 resolution Shepp-Logan phantom in a 800×800 image grid given in Fig. 5.10 with $\eta = 1$. We compute the matrix A and the measurement data b with $N_\mathrm{p} = 25$ projections, $N_\mathrm{r} = 1131$ rays per projection and 1% additive noise. We construct $\bar{Z}$ as reference training images with scale factors 0.5, 2, 3, and 4 (see Fig. 5.10). We should here mention that it is important that all of these images have the same number of pixels, to avoid scaling issues with the numerical computations. We create an artificial noise-free sinogram of this training images. The images X, $\bar{Z}$ and the corresponding sinograms of our tomographic data are shown in Fig. 5.10. The number of pixels in the images given in Fig. 5.10 is 800^2.

We compute M_X and $M_{\bar{Z}}$ from the given sinograms in Fig. 5.10. We obtain $\eta = [0.51, 2.05, 3.11, 4.17]$, which are an approximation to the correct scale factors $[0.5, 2, 3, 4]$.



(a) $\bar{Z}, \eta = 0.5$

(b) Clean S, $M_{\bar{Z}} = 26.70$

(c) $X, \eta = 1$

(d) Noisy S, $M_X = 52.11$

(e) $\bar{Z}, \eta = 2$

(f) Clean S, $M_{\bar{Z}} = 107.22$

(g) $\bar{Z}, \eta = 3$

(h) Clean S, $M_{\bar{Z}} = 162.56$

(i) $\bar{Z}, \eta = 4$

(j) Clean S, $M_{\bar{Z}} = 217.86$

Figure 5.10: Left: the reference Shepp-logan phantom image X, $\eta = 1$ and training images $\bar{Z}$ with scale factor $\eta = 0.5, 2, 3, 4$. Right: the clean sinograms ($S \in \mathbb{R}^{N_\mathrm{r} \times N_\mathrm{p}}$) of $\bar{Z}$ and noisy sinogram of X with $N_\mathrm{p} = 25$ projections and $N_\mathrm{r} = 1131$ rays.

Now let us consider our textural 200×200 peppers test image with $\eta = 1$ given in Fig. 4.3. We identically consider $\bar{Z}$ as training images of size 200×200, similar to our test image with scale factors 0.5, 2, 3, 4 and compute the sinogram matrix S with analogous tomographic scenario, i.e., $N_{\mathrm{p}} = 25$ projections in $[0°, 180°]$, $N_{\mathrm{r}} = 283$ rays per projection and 1% additive noise (see Fig. 5.11). Computing M_X and $M_{\bar{Z}}$ from the given sinograms in Fig. 5.11 results in approximating the scale factors to be [0.9647, 1.0000, 1.0738, 0.9649, 1.3451], showing that this method is not suited for images with textures without a zero patching of pixels around the object under study.

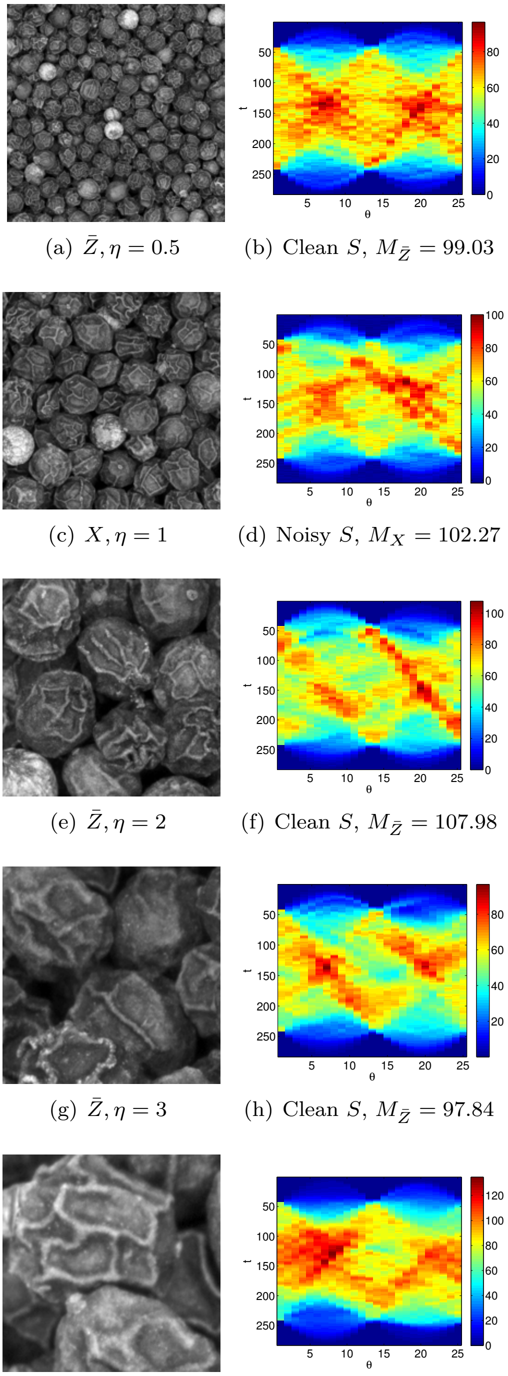
We can conclude that this method only works well if the unknown image is a single object with an unknown scale, and a training image includes a similar object with a different scale.

Finding the scale factor in 3D tomographic reconstruction where the tomographic data is available in form of projection images in which a multitude of details of the shapes and features are already visible, is a fairly straightforward process. Because the shapes in 2D slices of training images can be compared with similar shapes in the 2D projection data and the scale factor can be found with simple mathematical functions from geometry, e.g., we need to find a corresponding side in each similar shape in two images where we can measure the length of both. The ratio between the length of these sides is the scale factor.

5.2.3 Sensitivity to Rotation

In this section we analyze the sensitivity of the reconstruction results to a rotation parameter. We use three test images of size 200×200 which we call “peppers”, “binary” and “D53”. The D53 test image is chosen from the normalized brodatz texture database [54]. For the peppers test image we use the exact image given in Fig. 4.3. The binary and D53 test images are given in Fig. 5.12. We expect that the peppers test image is invariant to rotation while the binary and D53 test images, as can be seen in Fig. 5.12, are highly directional and sensitive to rotation.

We choose rotation angles of [5°, 10°, 30°, 45°, 60°, 90°] and we rotate the test images with the chosen angles. Since the rotated images are not exactly equivalent to the original test images, for the comparison of the reconstruction qualities to be fair, we extracted 4 smaller test images of size 50×50 from each rotated image. We use a reconstruction scenario with 12 projections and 70 rays in $[0°, 180°]$ and 1% noise. We obtain a reconstruction for each 50×50 image in every rotation and average over the reconstruction errors and SSIM measures. Figure 5.13 shows the plots for the average reconstruction errors (RE) and SSIM



(a) $\bar{Z}, \eta = 0.5$

(b) Clean S, $M_{\bar{Z}} = 99.03$

(c) $X, \eta = 1$

(d) Noisy S, $M_X = 102.27$

(e) $\bar{Z}, \eta = 2$

(f) Clean S, $M_{\bar{Z}} = 107.98$

(g) $\bar{Z}, \eta = 3$

(h) Clean S, $M_{\bar{Z}} = 97.84$

(i) $\bar{Z}, \eta = 4$

(j) Clean S, $M_{\bar{Z}} = 137.33$

Figure 5.11: Left: the reference peppers image X, $\eta = 1$ and training images $\bar{Z}$ with scale factor $\eta = 0.5$, 2, 3, 4. Right: the clean sinograms ($S \in \mathbb{R}^{N_\mathrm{r} \times N_\mathrm{p}}$) of $\bar{Z}$ and noisy sinogram of X with $N_\mathrm{p} = 25$ projections and $N_\mathrm{r} = 283$ rays.

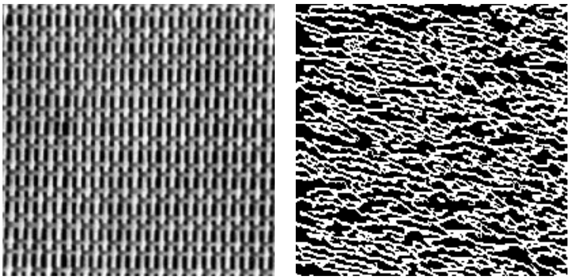


Figure 5.12: The 200 × 200 test images for the rotation sensitivity analysis. Left: the D53 and right: the binary test images.

measures versus the rotation angles for our three test images.

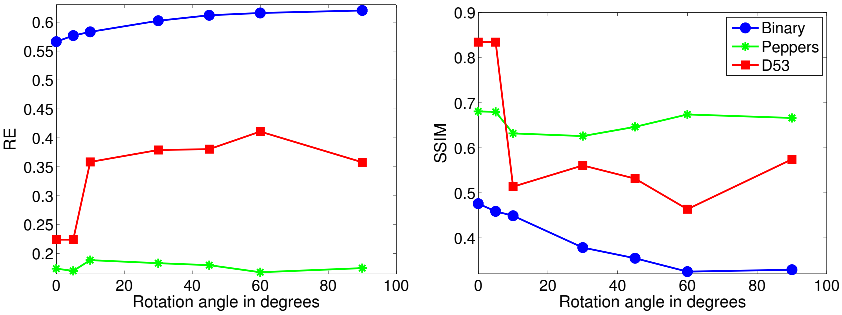


Figure 5.13: Left: The plots for the average reconstruction errors (RE) and, Right: SSIM measures versus the rotation degrees for our three test images, where 4 smaller test images of size 50 × 50 are extracted from each rotated test image.

The plots in Fig. 5.13 show that while, as expected, no particular sensitivity trends for the peppers test image can be detected by changing the rotation angles, the binary test image and D53 are highly sensitive to rotation and the worst reconstructions in term of RE and SSIM measures are obtained with the 60° and 90° rotation.

5.2.4 An Algorithm to Determine Rotation Angle

If the angle of rotation is known in advance to the reconstruction step, one can learn the dictionaries with larger patch sizes and then rotate the dictionary images by the known angle. The pixels that fall outside the boundaries of the original dictionary image are, in MATLAB, set to 0 and appear as a black background in the rotated image. However, we can specify a smaller patch size and exclude the boundary pixels with zero values in the rotated dictionary elements and extract rotated dictionary images of smaller size than the original one, to include in the reconstruction step.

If an image is given, the principal direction of the image can be estimated from the Radon transform of the image [23]. The Radon transform can be used to detect linear trends in images. For general images, the principal orientation may be taken as the direction along which the Radon transform has the maximum variability [23].

Let v_j denote the variance of the sinogram data for the jth projection, i.e., the jth column of the sinogram matrix S:

$$v_j = \frac{1}{N_\mathrm{r}-1}\sum_{k=1}^{N_\mathrm{r}} \left(S_{k,j} - \mathscr{M}_j\right)^2, \quad \forall\, j = 1, \ldots, N_\mathrm{p},$$

where $\mathscr{M}_j$ is the mean of each column vector in S,

$$\mathscr{M}_j = \frac{1}{N_\mathrm{r}}\sum_{k=1}^{N_\mathrm{r}} S_{k,j}, \quad \forall\, j = 1, \ldots, N_\mathrm{p}.$$

An important observation in [58] was that the sinogram $\mathrm{R}_\theta X(t)$ along θ has larger variations with respect to t for the principal angle with most directional lines. Hence in our case with angles θ_j, $j = 1, \ldots, N_\mathrm{p}$:

$$\Theta = \theta_{j^\star}, \quad j^\star = \arg\max_j v_j$$

is the direction with most linear trends along it. Such an estimate is useful for estimating the presence of rotation in the images.

We can assume that $\tilde{z} \in \mathbb{R}^n$ is a sub-image from the training image of a similar size as the unknown image x. We compute the sinogram of $\tilde{z}$ by generating the tomographic data by $A\tilde{z}$ and representing it as a matrix. We compute $\max_\theta \tilde{v}_\theta$ and find $\tilde{\Theta}$ to be the angle of most directional trends in the sub-image $\tilde{z}$. We refer to $\tilde{\Theta}$ as the reference angle of the training image. Similarly, we compute $\Theta = \arg\max_\theta v_\theta$ for the unknown image x. Then the rotation is approximately the difference between the angles, i.e., $\Theta - \tilde{\Theta}$.

To test this claim, let us choose 200×200 test images – similar to the D53 test image given in Fig. 5.12 – rotated by [5°, 10°, 30°, 45°, 60°, 90°], making six test images. We consider a training image with no rotation, i.e., with rotation angle 0° of size 200 × 200. In our first computational test, to find the correct rotation angle, we consider a tomographic scenario with a full data set, i.e., projections from all possible angles. The $N_{\mathrm{p}} = 180$ projections are sampled with equidistant steps over [0°, 180°], moreover we consider $N_{\mathrm{r}} = 283$ and 1% noise in the data.

Figure 5.14 shows the variance plots of the sinograms of our training image and rotated test images with different orientations. The sinogram of the reference training image with no rotation is noise free, while noise is present in the sinograms of the rotated test images. Note that the variance of the projections has two local maxima at 90° and 179° for the reference training image with no rotation. The local maximum at 179° is narrower compared with the local maximum at 90°, because there are more straight lines along 179°. Hence 179° is the reference orientation.

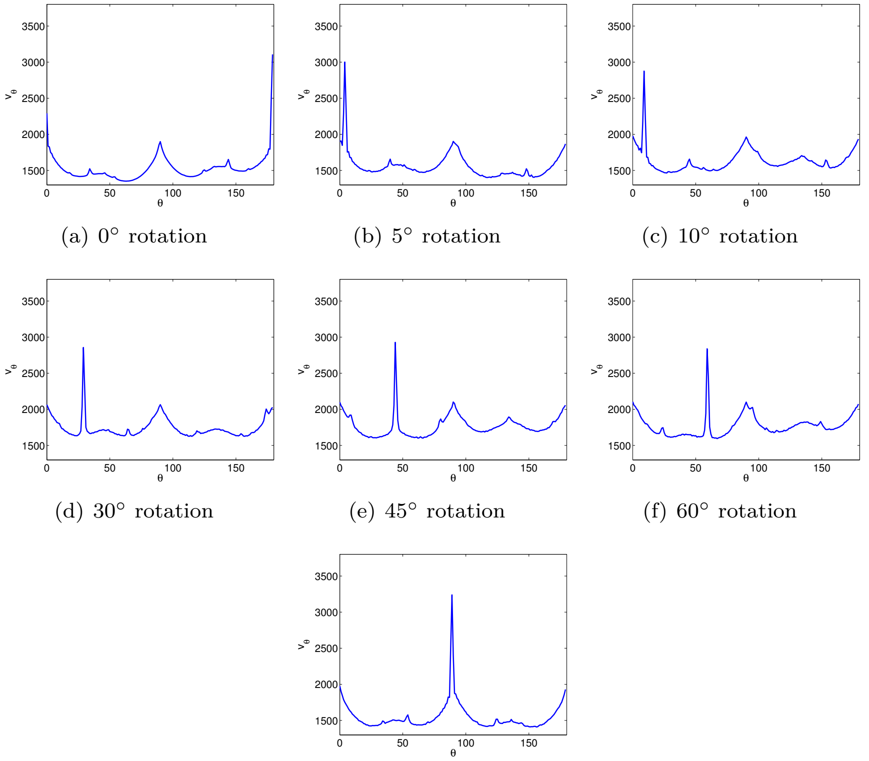
Given the plots in Fig. 5.14, we calculate the rotation degrees by finding the angle with the maximum variance in each plot, the difference to the original orientation in the reference training image gives the correct rotation. The estimations based on the full tomographic data are accurate and we obtain all the rotation angles, i.e., 5°, 10°, 30°, 45°, 60°, and 90°.

We now consider tomographic data with data from few projections of the same directional D53 images, we use 50 projections with uniform angular sampling in $[0°, 180°]$ and with relative noise level 1%, i.e., the same noise level as above. The variances of the sinograms of the training image and the test images with rotational angles [5°, 10°, 30°, 45°, 60°, 90°] are given in Fig. 5.15.

The variance plots in Fig. 5.15 indicate that with limited tomographic data where the projection data along some directions are missing and the information of the variances along all the directions are not attainable, we may not be able to find the correct orientation of the directional textures in the image. Note how the peak in the variance plot with the 10° rotation is missing. We find the rotation angles to be

$$[3.67°, 180°, 29.39°, 44.08°, 58.78°, 88.16°].$$

We observe that the method fails to find the correct orientation for the image with 10° rotation. One possible way to compensate for the missing projection data and construct new data points for these missing projections from the known ones, is to use interpolation of the tomographic data in the sinogram. Using linear 2D interpolation for gridded data, we approximate the rotated angles as before, where we obtain [4°, 0°, 29°, 44°, 59°, 88°] as the rotations. Although we still can not achieve the correct orientation for the image with 10° rotation,



(a) 0° rotation

(b) 5° rotation

(c) 10° rotation

(d) 30° rotation

(e) 45° rotation

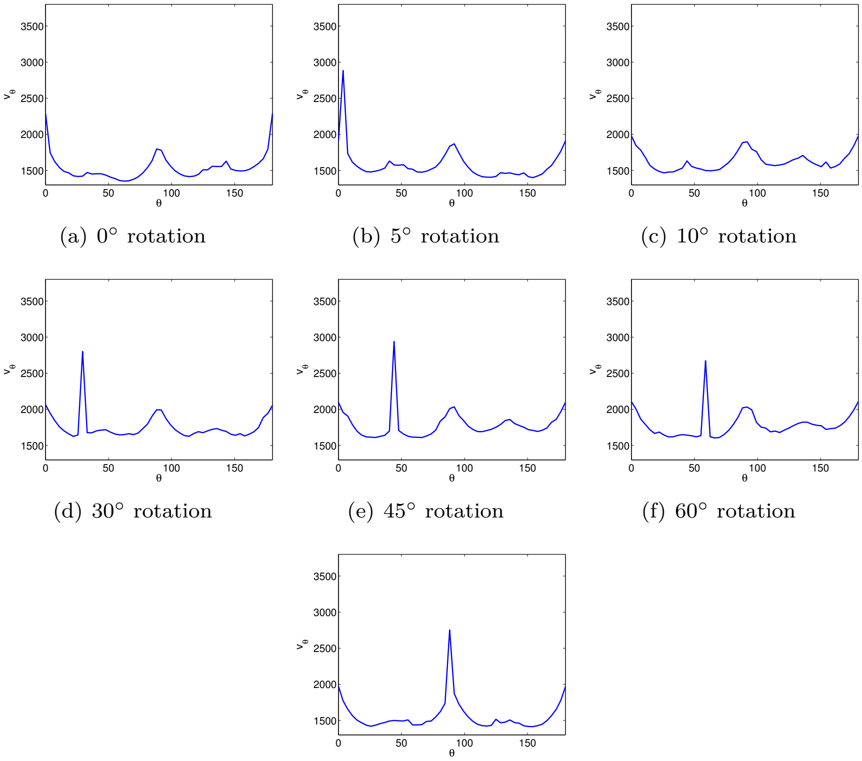
(f) 60° rotation

(g) 90° rotation

Figure 5.14: The variance of sinograms from the 200×200 D53 six test images with different rotation angles with full tomographic data comparing to a similar training image with no rotation of 200×200 size. Note how the maximum in the variance plots changes as the rotation degrees varies.

in the presence of noise we can still approximate other rotation angles with a small error.

To complete this picture, we consider a tomographic problem where an exact image is given in Fig. 5.16. This exact image is rotated by 30° from the reference training image. We consider the same tomographic scenario with 50 projections in $[0°, 180°]$ and 1% noise. By the above method for the noisy sinogram we approximate the rotation angle to be 29°. A dictionary from 20×20 patches from the training image, i.e., $D \in \mathbb{R}^{400 \times 800}$, is computed; each dictionary image is rotated by 29° and then 10×10 dictionary elements are extracted from the



(a) 0° rotation

(b) 5° rotation

(c) 10° rotation

(d) 30° rotation

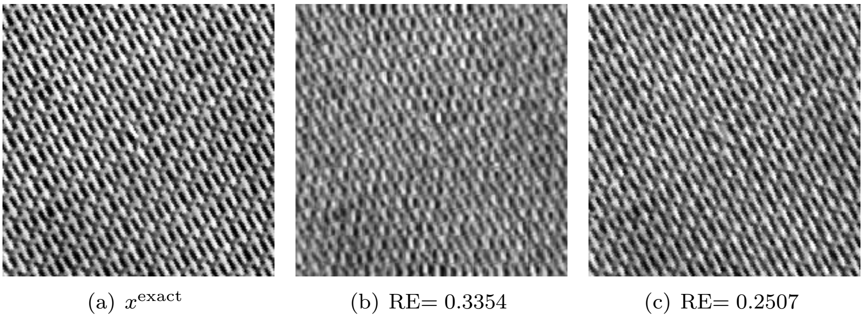
(e) 45° rotation

(f) 60° rotation

(g) 90° rotation

Figure 5.15: The variance of sinograms from the six 200 × 200 D53 test images with different rotation angles with limited tomographic data compared to a similar training image with no rotation of 200×200 size. Note how the maximum in the variance plots changes as the rotation degrees varies. With the limited tomographic data, the maximum disappears when rotating the reference image with 10°.

rotated 20 × 20 dictionary basis images, and then 300 dictionary elements are randomly chosen from these 800 rotated dictionary images. Now a new rotated dictionary such that $\bar{D} \in \mathbb{R}^{100\times300}$ is at hand. We reconstruct the image using the rotated dictionary $\bar{D}$ and compare it with a reconstruction obtained using 10 × 10 dictionary elements and $s = 300$, obtained from the reference training image with 0° rotation. The results are illustrated in Fig. 5.16 which shows clearly how using a correctly rotated dictionary can improve the reconstruction significantly.



(a) x^{exact}

(b) RE= 0.3354

(c) RE= 0.2507

Figure 5.16: Left: The 30° rotated exact image. Middle: The tomographic reconstruction using a dictionary obtained from our reference training image without any knowledge of rotation. Right: The reconstructed solution with a rotated dictionary where the degree of rotation is approximated from the noisy sinogram of the tomographic data.

5.3 Summary

The work in this Chapter is an extension of the computational experiments in Chapter 4, while here, we further examined our problem formulation by numerically investigating the sensitivity of the reconstruction formulation to the representation by the dictionary and the model constraints as well as inconsistencies in scale and rotation of the unknown image to the training images. It is shown that using the nonnegative dictionary and representation had a regularizing effect on the solution.

In addition, algorithms to determine the correct scale and rotation degree of the unknown image from the tomographic sinogram are suggested. Numerical examples showed that both methods can be advantageous in obtaining the correct scale and rotation of the unknown image from the measurement data, however, future work concerning approximating the correct scale of unknown textural images from the given sinogram where the proposed method fails should be considered.

Chapter 6

A Tensor-Based Dictionary Learning Approach to CT

Images are naturally two-dimensional objects and we find it fundamentally reasonable to work with them in their natural form (as a matrix). For example we are interested in capturing image-to-image correlations (not just pixel-to-pixel) that let us reduce the overall redundancy in the data. As illustrated in Fig. 6.1 by vectorizing images, the spatial coherences of the features in images may be lost.

One common feature in the dictionary learning literature and sparse representation in terms of these dictionaries is the reliance on the (invertible) mapping of 2D images to vectors and subsequent use of a linear algebraic framework: Matrices are used for the dictionary representation (the columns represent vectorized forms of image features) and the use of a linear combination of the columns of the dictionary gives the expression of the image, in its vectorized form. However, the training data itself can be more naturally represented as a multidimensional array, called a tensor. For example, a collection of K gray-scale images of size $M \times N$ could be arranged in an $M \times K \times N$ array, also known as a third-order tensor. Recent work in imaging applications such as facial recognition [49], video completion [120] has shown that using the right kind of factorizations of particular tensor-based representations of the data can have a distinct advantage over matrix-based counterparts. For this reason, in this

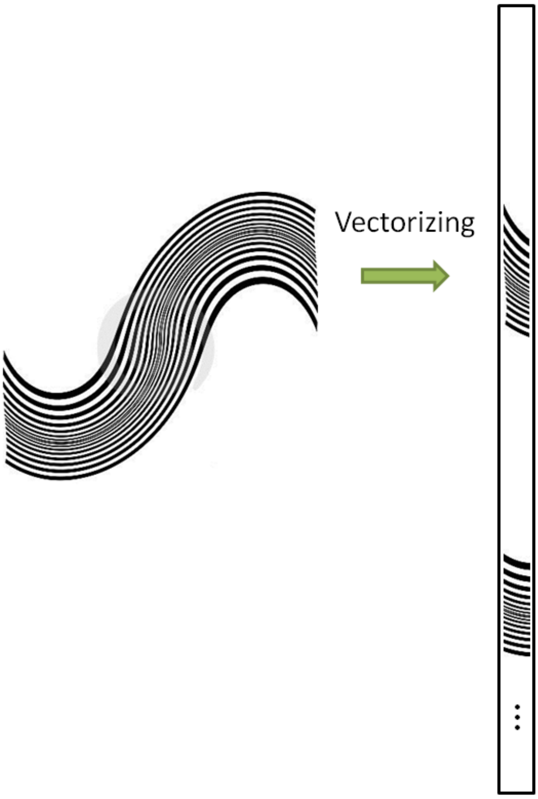


Figure 6.1: By arranging images in vectors the correlations between pixels may be lost or distort. The image is from http://www.dreamstime.com

chapter we will develop a fundamentally new approach for both the dictionary learning and image reconstruction tasks that is based on a particular type of tensor decomposition, i.e., the t-product introduced in [64].

There are several different tensor factorizations and decompositions such as CANDECOMP/PARAFAC (CP) [62] and Tucker decomposition [111]. The use of different decompositions is driven by applications as well as the properties of the decompositions. For an extensive list of tensor decompositions, their applications, and further references, see [65]. It is natural to use higher-order tensor decomposition approaches in imaging problems, which are nowadays frequently used in image analysis and signal processing [2, 17, 24, 49, 63].

Some recent works provide algorithms and analysis for tensor sparse coding and dictionary learning based on different factorization strategies. Caiafa and Cichocki [17] discuss multidimensional compressed sensing algorithms using the Tucker decomposition. Zubair and Wang [123] propose a tensor learning algorithm based on the Tucker model with a sparsity constraint on the core tensor. Tensor-based extensions of the method of optimal directions (MOD) [33] and the KSVD algorithm [1] have been studied in [95] for separable multidimensional dictionaries. An algorithm for tensor dictionary learning based on the CP decomposition, called K-CPD, is presented in [29]. In the context of tomography, we are only aware of the work by Tan et. al. [106] applying a tensor-MOD dictionary learning approach using Tucker decomposition in dynamic tomographic reconstruction.

Recent work by Kilmer et al. [63] sets up a new theoretical framework which facilitates a straightforward extension of matrix factorizations to third-order tensors based on a new tensor multiplication definition, called the *t-product*. The motivation for our work is to use the t-product as a natural extension for the dictionary learning problem and image reconstruction in a third-order tensor formulation with the factorization based on the framework in [64] and [63].

In this chapter we re-visit the dictionary learning approach introduced in Chapter 4 for X-ray CT reconstruction, now using a tensor formulation of the problem. We will consider a collection of training patches as a third-order tensor, with each 2D image making up a slice of the data tensor. We describe this approach in more details in this chapter.

The main contributions of this chapter are:

- It is shown that the new tensor factorization formulation is not a trivial reformulation of the matrix-based decomposition.
- A third-order tensor representation for the training images is used and a tensor dictionary learning problem for our tensor training data is formulated using the tensor product introduced in [64].
- An algorithm based on the alternating direction method of multipliers (ADMM) for solving the tensor dictionary learning problem is described.
- The reconstruction problem is formulated in terms of recovering the expansion coefficients in the tensor dictionary, i.e., recasting a tensor formulation for the reconstruction problem in terms of a convex optimization problem.
- It is shown that considering a tensor formulation over a matrix formulation significantly reduces the approximation error by the dictionary.

- It is demonstrated that in the tensor formulation, a much sparser representation is obtained of both the dictionary and the reconstruction, due to the ability of representing repeated features compactly in the dictionary.

6.1 Notations and Preliminaries on Tensors

In this section we present the definitions and notations that will be used throughout this chapter. We exclusively consider the tensor definitions and the tensor product notation introduced in [64] and [63]. Throughout the chapter, a capital italics letter such as A denotes a matrix and a capital calligraphy letter such as $\mathcal{A}$ denotes a tensor.

A *tensor* is a multidimensional array of numbers. The *order* of a tensor refers to its dimensionality. Thus, if $\mathcal{A} \in \mathbb{R}^{l\times m\times n}$ then we say $\mathcal{A}$ is a third-order tensor. A $1 \times 1 \times n$ tensor is called a *tube fiber*. A graphical illustration of a third-order tensor decomposed into its tube fibers is given in the upper right image of Fig. 6.2. Thus, one way to view a third-order tensor is as a matrix of tube fibers. In particular, an $\ell \times 1 \times n$ tensor is a vector of tube fibers. To make this clear, we use the notation $\overrightarrow{\mathcal{A}_j} = \mathcal{A}(:, j, :)$ to denote the jth "column" or *lateral slice* of the third-order tensor (see the middle figure of the bottom row of Fig. 6.2). The kth *frontal slice*, which is an $\ell \times m$ matrix, is denoted by $A^{(k)} \equiv \mathcal{A}(:, :, k)$. Frontal slices and other decompositions of a third-order tensor are shown in Fig. 6.2.

We can consider an $l\times 1\times n$ tensor is a matrix oriented into the third dimension. It will therefore be useful to use notation from [64] that allows us to easily move between $l\times n$ matrices and their $l\times 1\times n$ counterparts (see Fig 6.3). Specifically, the `squeeze` operation on $\overrightarrow{\mathcal{X}} \in \mathbb{R}^{l\times 1\times n}$ is identical to the `squeeze` function in MATLAB:

$$X = \texttt{squeeze}(\overrightarrow{\mathcal{X}}) \quad \Rightarrow \quad X(i, k) = \overrightarrow{\mathcal{X}}(i, 1, k).$$

The `vec` function unwraps the tensor $\mathcal{A}$ into a vector of length ℓmn by column stacking of frontal slices, i.e., in MATLAB notation: $\texttt{vec}(\mathcal{A}) \equiv \mathcal{A}(:)$. For the tensor $\mathcal{A}$ we define the `unfold` and `fold` functions in terms of frontal slices:

$$\texttt{unfold}(\mathcal{A}) = \begin{pmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(n)} \end{pmatrix} \in \mathbb{R}^{ln\times m}, \qquad \texttt{fold}(\texttt{unfold}(\mathcal{A})) = \mathcal{A}.$$

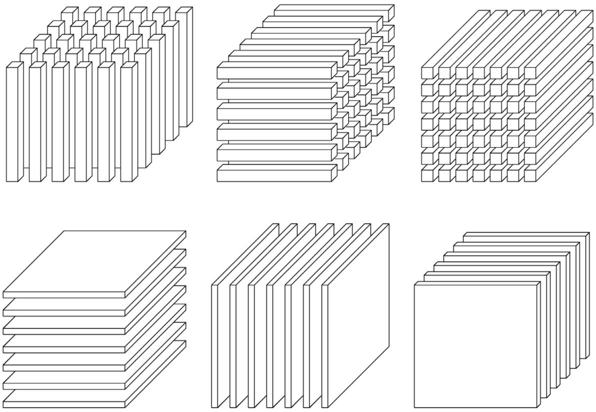


Figure 6.2: Different representations of a third-order tensor, from [65]. Top left to right: column, row, and tube fibers. Bottom left to right: horizontal, lateral, and frontal slices.

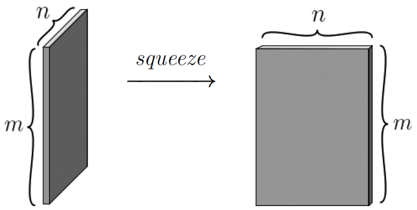


Figure 6.3: $m \times 1 \times n$ tensors and $m \times n$ matrices related through the squeeze operation, figure from [63].

The block circulant matrix of size $\ell n \times nm$ that is generated via $\texttt{unfold}(\mathcal{A})$ is given as

$$\texttt{circ}(\mathcal{A}) = \begin{pmatrix} A^{(1)} & A^{(n)} & A^{(n-1)} & \cdots & A^{(2)} \\ A^{(2)} & A^{(1)} & A^{(n)} & \cdots & A^{(3)} \\ \vdots & & & \ddots & \vdots \\ A^{(n)} & A^{(n-1)} & A^{(n-2)} & \cdots & A^{(1)}. \end{pmatrix}.$$

Definition 1 *Let $\mathcal{B} \in \mathbb{R}^{l \times p \times n}$ and $\mathcal{C} \in \mathbb{R}^{p \times m \times n}$. Then the* ***t-product*** *from [64] is defined by*

$$\mathcal{A} = \mathcal{B} * \mathcal{C} \equiv \texttt{fold}\big(\texttt{circ}(\mathcal{B})\,\texttt{unfold}(\mathcal{C})\big),$$

from which it follows that $\mathcal{A}$ is an $\ell \times m \times n$ tensor.

The t-product can be considered as a natural extension of the matrix multiplication [13]. In general the t-product is not commutative between two arbitrary tensors, but it is commutative between tube fibers.

Definition 2 *Given m tube fibers $\mathbf{c}_j \in \mathbb{R}^{1\times1\times n}$, $j = 1, \ldots, m$ a **t-linear combination** [63] of the lateral slices $\overrightarrow{\mathcal{A}_j} \in \mathbb{R}^{\ell\times1\times n}$, $j = 1, \ldots, m$, is defined as*

$$\overrightarrow{\mathcal{A}_1} * \mathbf{c}_1 + \overrightarrow{\mathcal{A}_2} * \mathbf{c}_2 + \cdots + \overrightarrow{\mathcal{A}_m} * \mathbf{c}_m \equiv \mathcal{A} * \overrightarrow{\mathcal{C}},$$

where

$$\overrightarrow{\mathcal{C}} = \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_m \end{bmatrix} \in \mathbb{R}^{m\times1\times n} .$$

The multiplication $\mathbf{c}_j * \overrightarrow{\mathcal{A}}_j$ is not defined unless $\ell = 1$.

Definition 3 *The **identity tensor** $\mathcal{I}_{mmn}$ is the tensor whose first frontal slice is the $m \times m$ identity matrix, and whose other frontal slices are all zeros.*

Definition 4 *An $m \times m \times n$ tensor $\mathcal{A}$ has an **inverse** $\mathcal{B}$, provided that*

$$\mathcal{A} * \mathcal{B} = \mathcal{I}_{mmn} \quad \textit{and} \quad \mathcal{B} * \mathcal{A} = \mathcal{I}_{mmn}.$$

Definition 5 *Following [64], if $\mathcal{A}$ is $l \times m \times n$, then the **transposed tensor** $\mathcal{A}^{\mathrm{T}}$ is the $m \times l \times n$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n.*

Definition 6 *Let a_{ijk} be the i, j, k element of $\mathcal{A}$. Then the **Frobenius norm** of the tensor $\mathcal{A}$ is*

$$\|\mathcal{A}\|_{\mathrm{F}} = \|\mathrm{vec}(\mathcal{A})\|_2 = \sqrt{\sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n} a_{ijk}^2}.$$

We also use the following notation:

$$\|\mathcal{A}\|_{\mathrm{sum}} = \|\mathrm{vec}(\mathcal{A})\|_1 = \sum_{i,j,k} |a_{ijk}|, \qquad \|\mathcal{A}\|_{\max} = \|\mathrm{vec}(\mathcal{A})\|_\infty = \max_{i,j,k} |a_{ijk}|.$$

If A is a matrix then $\|A\|_{\mathrm{sum}} = \sum_{i,j} |a_{ij}|$. Let σ_i, $i = 1, \ldots, \min\{m, n\}$ denote the singular values of A. The *nuclear norm* (also known as the trace norm) is defined as

$$\|A\|_* = \mathtt{trace}(\sqrt{A^{\mathrm{T}}A}) = \sum_{i=1}^{\min\{m,n\}} \sigma_i.$$

6.2 Tensor Dictionary Learning

In recent years there has been an increasing interest in obtaining a non-negative *tensor* factorization (NTF) (often based on CP and Tucker decompositions) as a natural generalization of the NMF for a nonnegative data. Similar to NMF, the sparsity of the representation has been empirically observed in NTF based on CP and Tucker decompositions. For NTF based on a subset of tensor decomposition methods, we refer to [24]. Unlike the work in [24], we express the dictionary learning problem in a third-order tensor framework based on the t-product. This will be described in detail below, but the key is a t-product-based NTF reminiscent of the NMF.

The NTF based on the t-product was proposed in [48], where preliminary work with MRI data showed the possibility that sparsity is encouraged when non-negativity is enforced. Here, we extend the work by incorporating sparsity constraints and we provide the corresponding optimization algorithm. Given the patch tensor dictionary $\mathcal{D}$, we compute reconstructed images that have a sparse representation in the space defined by the t-product and $\mathcal{D}$. Thus, both the dictionary and the sparsity of the representation serve to regularize the ill-posed problem.

6.2.1 Tensor Factorization via t-Product

Let the third-order *data tensor* $\mathcal{Y} \in \mathbb{R}_+^{p\times t\times r}$ consist of t training image patches of size $p \times r$, arranged as the lateral slices of $\mathcal{Y}$, i.e.,

$$\overrightarrow{\mathcal{Y}_j} = \mathcal{Y}(:,j,:), \quad \text{for} \quad j = 1,\ldots,t,$$

see Fig. 6.4.

Our non-negative tensor decomposition problem, based on the t-product, is the problem of writing the non-negative data tensor as a product $\mathcal{Y} = \mathcal{D} * \mathcal{H}$ of two tensors $\mathcal{D} \in \mathbb{R}^{p\times s\times r}$ and $\mathcal{H} \in \mathbb{R}^{s\times t\times r}$. The tensor $\mathcal{D}$ consists of s dictionary 2D image patches of size $p \times r$ arranged as the lateral slices of $\mathcal{D}$, while $\mathcal{H}$ is the tensor of coefficients.

The main difference between NTF and NMF is that the $s \times t \times r$ tensor $\mathcal{H}$ has r times more degrees of freedom in the representation than the $s \times t$ matrix H. To make this clear, an illustration of the tensor factorization versus the matrix factorization is given in Fig. 6.5.

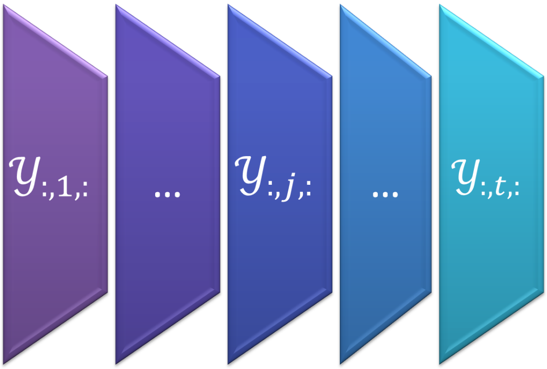


Figure 6.4: The third-order training tensor $\mathcal{Y} \in \mathbb{R}_{+}^{p\times t\times r}$, where training patches are arranged as the lateral slices of $\mathcal{Y}$.

The t-product from Definition 1 involves unfolding and forming a block circulant matrix of the given tensors. Using the fact that a block circulant matrix can be block-diagonalized by the Discrete Fourier Transform (DFT) [39, §4.7.7], the t-product is computable in the Fourier domain [63]. Specifically, we can compute $\mathcal{Y} = \mathcal{D} * \mathcal{H}$ by applying the DFT along tube fibers of $\mathcal{D}$ and $\mathcal{H}$:

$$\widehat{\mathcal{Y}}(:,:,k) = \widehat{\mathcal{D}}(:,:,k)\widehat{\mathcal{H}}(:,:,k), \qquad k = 1,\dots,r,$$

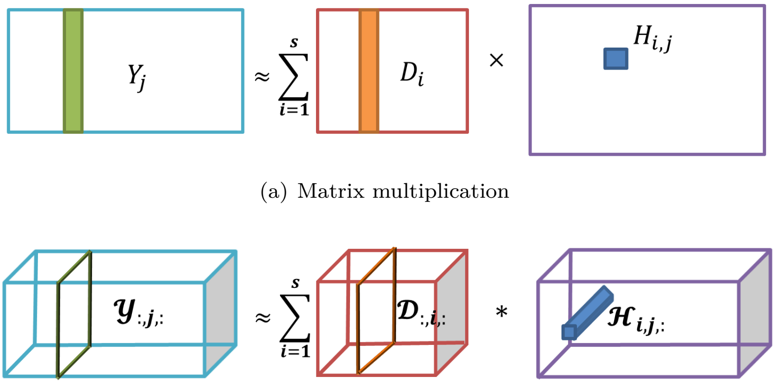
where $\widehat{\ }$ denotes DFT; in MATLAB notation we apply the DFT across the third dimension: $\widehat{\mathcal{D}} = \texttt{fft}(\mathcal{D},[\,],3)$, $\widehat{\mathcal{H}} = \texttt{fft}(\mathcal{H},[\,],3)$, then: $\mathcal{Y} = \texttt{ifft}(\widehat{\mathcal{Y}},[\,],3)$. Working in the Fourier domain conveniently reduces the number of arithmetic operations [49], and since the operation is separable in the third dimension it allows for parallelism.

Although the representation of the training patches in the third-order tensor resembles the matrix formulation, it is not a re-formulation of the matrix problem packaged as tensors. In fact, the tensor formulation gives a richer approach of formulating the problem, as we now describe.

Recall that the jth patch Y_j is the jth lateral slice of $\mathcal{Y} = \mathcal{D} * \mathcal{H}$, i.e., $Y_j = \texttt{squeeze}\big(\mathcal{Y}(:,j,:)\big)$. Hence, as shown in [48],

$$Y_j = \sum_{i=1}^{s} \texttt{squeeze}\big(\mathcal{D}(:,i,:)\big)\,\texttt{circ}\Big(\texttt{squeeze}\big(\mathcal{H}(j,i,:)^{\mathrm{T}}\big)\Big). \tag{6.1}$$

In other words, the jth patch is a sum over all the lateral slices of $\mathcal{D}$, each one "weighted" by multiplication with a circulant matrix derived from a tube fiber of $\mathcal{H}$.



(a) Matrix multiplication

(b) Tensor multiplication

Figure 6.5: A visual interpretation of a third order tensor ($\mathcal{Y}(:,j,:)$) approximated as a sum of t-products of lateral slices in $\mathcal{D}$ ($\mathcal{D}(:,i,:)$) and tubal scalars of $\mathcal{H}$. Note that if the third dimension, is one, the t-product becomes regular matrix multiplication, and then this illustration collapses to an approximation of a matrix as a sum of products of the vectors in D.

We use a small example to show why this is significant. Consider the 3×3 down-shift matrix and the (column) circulant matrix generated by the vector v:

$$Z = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \qquad C[v] = \mathtt{circ}(v) = \begin{pmatrix} v_1 & v_3 & v_2 \\ v_2 & v_1 & v_3 \\ v_3 & v_2 & v_1 \end{pmatrix}.$$

Noting that

$$C[v] = \sum_{k=1}^{3} v_k Z^{k-1} = v_1 I + v_2 \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} + v_3 \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}.$$

it follows that

$$D\,C[v] = \sum_{k=1}^{3} v_k D Z^{k-1}.$$

Extrapolating to (6.1), we obtain the following result.

Theorem 6.1 *Let Z denote the $n \times n$ down-shift matrix. With $D_i = \mathtt{squeeze}\big(\mathcal{D}(:,i,:)\big)$ and $h^{(ij)} = \mathtt{squeeze}(\mathcal{H}(j,i,:)^{\mathrm{T}})$, the jth image patch is given by*

$$Y_j = \sum_{i=1}^{s} D_i C[h^{(ij)}] = \sum_{i=1}^{s} \left(h_1^{(ij)} D_i + \sum_{k=2}^{n} h_k^{(ij)} D_i Z^{k-1} \right). \tag{6.2}$$

To show the relevance of this result we note that the product $D_i Z^{k-1}$ is D_i with its columns cyclically shifted left by $k-1$ columns. Assuming that D_i represents a "prototype" element/feature in the image, we now have a way of also including shifts of that prototype in our dictionary *without* explicitly storing those shifted bases in the dictionary. Note that if $h_k^{(ij)} = 0$, $k = 2, \dots, n$ then Y_j is a (standard) linear combination of matrices D_i; this shows that our new approach effectively subsumes the matrix-based approach from Chapter 4, while making the basis richer with the storage of only a few entries of a circulant matrix rather than storing extra basis image patches!

6.2.2 Formulation of the Tensor-Based Dictionary Learning Problem

One is usually not interested in a perfect factorization of the data because overfitting can occur, meaning that the learned parameters do fit well the training data, but have a bad generalization performance. This issue is solved by making a priori assumptions on the dictionary and coefficients.

Based on the approximate decomposition $\mathcal{Y} \approx \mathcal{D} * \mathcal{H}$, we consider the generic tensor-based dictionary learning problem (similar to the matrix formulation in 3.1):

$$\min_{\mathcal{D},\mathcal{H}} \quad \mathscr{L}_{\text{dic}}(\mathcal{Y}, \mathcal{D} * \mathcal{H}) + \Phi_{\text{dic}}(\mathcal{D}) + \Phi_{\text{rep}}(\mathcal{H}). \tag{6.3}$$

The misfit of the factorization approximation is measured by the loss function $\mathscr{L}_{\text{dic}}$, (e.g., the Frobenius norm). Different priors on the dictionary $\mathcal{D}$ and the representation tensor $\mathcal{H}$ are controlled by the regularization functions $\Phi_{\text{dic}}(\mathcal{D})$ and $\Phi_{\text{rep}}(\mathcal{H})$.

NTF itself results in a sparse representation. Imposing sparsity-inducing norm constraints on the representation allows us to further control sparsity of the representation of the training image, i.e., the training patches being represented as a combination of a small number of dictionary elements. At the same time this alleviates the non-uniqueness drawback of the NTF.

Therefore, similar to the dictionary learning formulation in (4.1) we pose the tensor dictionary learning problem as a non-negative sparse coding problem [53]:

$$\min_{\mathcal{D},\mathcal{H}} \frac{1}{2}\|\mathcal{Y} - \mathcal{D} * \mathcal{H}\|_{\text{F}}^2 + \lambda\|\mathcal{H}\|_{\text{sum}} + I_{\mathsf{D}}(\mathcal{D}) + I_{\mathbb{R}_+^{s \times t \times r}}(\mathcal{H}). \tag{6.4}$$

Here D is a closed set defined below, I_Z denotes the indicator function of a set Z, and $\lambda \geq 0$ is a regularization parameter that controls the sparsity-inducing penalty $\|\mathcal{H}\|_{\text{sum}}$. If we do not impose bound constraints on the dictionary elements, then the dictionary and coefficient tensors $\mathcal{D}$ and $\mathcal{H}$ can be arbitrarily scaled, because for any $\zeta > 0$ we have $\|\mathcal{Y} - (\zeta\mathcal{D}) * (\frac{1}{\zeta}\mathcal{H})\|_{\text{F}}^2 = \|\mathcal{Y} - \tilde{\mathcal{D}} * \tilde{\mathcal{H}}\|_{\text{F}}^2$. We define the compact and convex set D such that $\mathcal{D} \in \mathsf{D}$ prevents this inconvenience:

$$\mathsf{D} \equiv \{\mathcal{D} \in \mathbb{R}_+^{p\times s\times r} \mid \|\mathcal{D}(:,i,:)\|_{\text{F}} \leq \sqrt{pr},\ i = 1,\dots,s\}. \tag{6.5}$$

When $r = 1$ then (6.4) collapses to the standard non-negative sparse coding problem.

6.2.3 The Tensor-Based Dictionary Learning Algorithm

The optimization problem (6.4) is non-convex, while it is convex with respect to each variable $\mathcal{D}$ or $\mathcal{H}$ when the other is fixed. Computing a local minimizer can be done using the ADMM method [11], which is a splitting method from the augmented Lagrangian family. We therefore consider an equivalent form of (6.4):

$$\begin{array}{ll} \text{minimize}_{\mathcal{D},\mathcal{H},\mathcal{U},\mathcal{V}} & \frac{1}{2}\,\|\mathcal{Y} - \mathcal{U} * \mathcal{V}\|_{\text{F}}^2 + \lambda\,\|\mathcal{H}\|_{\text{sum}} + I_{\mathbb{R}_+^{s\times t\times r}}(\mathcal{H}) + I_{\mathsf{D}}(\mathcal{D}) \\ \text{subject to} & \mathcal{D} = \mathcal{U} \quad \text{and} \quad \mathcal{H} = \mathcal{V}, \end{array} \tag{6.6}$$

where $\mathcal{D},\mathcal{U} \in \mathbb{R}^{p\times s\times r}$ and $\mathcal{H},\mathcal{V} \in \mathbb{R}^{s\times t\times r}$. The augmented Lagrangian for (6.6) is

$$\begin{aligned} \mathcal{L}_\rho(\mathcal{D},\mathcal{U},\mathcal{H},\mathcal{V},\Lambda,\bar{\Lambda}) \;=\; & \frac{1}{2}\|\mathcal{Y} - \mathcal{U} * \mathcal{V}\|_{\text{F}}^2 + \lambda\,\|\mathcal{H}\|_{\text{sum}} + I_{\mathbb{R}_+^{s\times t\times r}}(\mathcal{H}) + I_{\mathsf{D}}(\mathcal{D}) \\ & + \Lambda^{\text{T}} \odot (\mathcal{D} - \mathcal{U}) + \bar{\Lambda}^{\text{T}} \odot (\mathcal{H} - \mathcal{V}) \\ & + \rho\big(\frac{1}{2}\|\mathcal{D} - \mathcal{U}\|_{\text{F}}^2 + \frac{1}{2}\|\mathcal{H} - \mathcal{V}\|_{\text{F}}^2\big), \end{aligned} \tag{6.7}$$

where $\Lambda \in \mathbb{R}^{p\times s\times r}$ and $\bar{\Lambda} \in \mathbb{R}^{s\times t\times r}$ are Lagrange multiplier tensors, $\rho > 0$ is the quadratic penalty parameter, and $\odot$ denotes the Hadamard (entrywise) product.

The objective function becomes separable by introducing the auxiliary variables $\mathcal{U}$ and $\mathcal{V}$. The alternate direction method is obtained by minimizing $\mathcal{L}_\rho$ with respect to $\mathcal{D}$, $\mathcal{H}$, $\mathcal{U}$, $\mathcal{V}$ one at a time while fixing the other variables at their most recent values and updating the Lagrangian multipliers Λ and $\bar{\Lambda}$. If P_{D} is the metric projection on D (which is computed using Dykstra's alternating projection algorithm [12]), then the ADMM updates are given by:

$$\begin{aligned}
\mathcal{D}_{k+1} &= \min_{\mathcal{D}\in\mathsf{D}} L_\rho(\mathcal{D},\mathcal{H}_k,\mathcal{U}_k,\mathcal{V}_k,\Lambda_k,\bar{\Lambda}_k) = P_{\mathsf{D}}(\mathcal{U}_k - \rho^{-1}\Lambda_k) && \text{(6.8a)}\\
\mathcal{V}_{k+1} &= \min_{\mathcal{V}} L_\rho(\mathcal{D}_k,\mathcal{H}_k,\mathcal{U}_k,\mathcal{V},\Lambda_k,\bar{\Lambda}_k) && \text{(6.8b)}\\
&= \left(\mathcal{U}_k^{\mathrm{T}} * \mathcal{U}_k + \rho\mathcal{I}\right)^{-1} * \left(\mathcal{U}_k^{\mathrm{T}} * \mathcal{Y} + \bar{\Lambda}_k + \rho\mathcal{H}_k\right)\\
\mathcal{H}_{k+1} &= \min_{\mathcal{H}\in\mathbb{R}_+^{s\times t\times r}} L_\rho(\mathcal{D}_{k+1},\mathcal{H},\mathcal{U}_k,\mathcal{V}_{k+1},\Lambda_k,\bar{\Lambda}_k) && \text{(6.8c)}\\
&= \mathsf{P}_+\left(\mathsf{S}_{\lambda/\rho}(\mathcal{V}_{k+1} - \rho^{-1}\bar{\Lambda}_k)\right)\\
\mathcal{U}_{k+1} &= \min_{\mathcal{U}} L_\rho(\mathcal{D}_{k+1},\mathcal{H}_k,\mathcal{U},\mathcal{V}_{k+1},\Lambda_k,\bar{\Lambda}_k) && \text{(6.8d)}\\
&= \left(\mathcal{Y} * \mathcal{V}_{k+1}^{\mathrm{T}} + \Lambda_k + \rho\mathcal{D}_{k+1}\right) * \left(\mathcal{V}_{k+1} * \mathcal{V}_{k+1}^{\mathrm{T}} + \rho\mathcal{I}\right)^{-1}\\
\Lambda_{k+1} &= \Lambda_k + \rho(\mathcal{D}_{k+1} - \mathcal{U}_{k+1}) && \text{(6.8e)}\\
\bar{\Lambda}_{k+1} &= \bar{\Lambda}_k + \rho(\mathcal{H}_{k+1} - \mathcal{V}_{k+1}). && \text{(6.8f)}
\end{aligned}$$

Here $\mathsf{P}_+(\Theta)_{i,j} = \max\{\theta_{i,j}, 0\}$ and $\mathsf{S}_{\lambda/\rho}$ denotes soft thresholding. The updates for $\mathcal{U}_{k+1}$ and $\mathcal{V}_{k+1}$ are computed in the Fourier domain.

The KKT-conditions for (6.7) can be expressed as

$$\begin{aligned}
\mathcal{D} = \mathcal{U}, &\quad \mathcal{H} = \mathcal{V},\\
\Lambda = -(\mathcal{Y} - \mathcal{D} * \mathcal{H}) * \mathcal{H}^{\mathrm{T}}, &\quad \bar{\Lambda} = -\mathcal{D}^{\mathrm{T}} * (\mathcal{Y} - \mathcal{D} * \mathcal{H}),\\
-\Lambda \in \partial\Phi_{\mathrm{dic}}(\mathcal{D}), &\quad -\bar{\Lambda} \in \partial\Phi_{\mathrm{rep}}(\mathcal{H}),
\end{aligned}$$

where $\partial f(\mathcal{X})$ denotes the sub-differential of f at $\mathcal{X}$. The KKT conditions are used to formulate stopping criteria for the ADMM algorithm, and we use the following conditions:

$$\frac{\|\mathcal{D} - \mathcal{U}\|_{\max}}{\max(1, \|\mathcal{D}\|_{\max})} \le \epsilon, \quad \frac{\|\mathcal{H} - \mathcal{V}\|_{\max}}{\max(1, \|\mathcal{H}\|_{\max})} \le \epsilon, \tag{6.9a}$$

$$\frac{\|\bar{\Lambda} - \mathcal{D}^{\mathrm{T}} * (\mathcal{D} * \mathcal{H} - \mathcal{Y})\|_{\max}}{\max(1, \|\bar{\Lambda}\|_{\max})} \le \epsilon, \quad \frac{\|\Lambda - (\mathcal{D} * \mathcal{H} - \mathcal{Y}) * \mathcal{H}^{\mathrm{T}}\|_{\max}}{\max(1, \|\Lambda\|_{\max})} \le \epsilon, \tag{6.9b}$$

where $\epsilon > 0$ is a given tolerance. Algorithm 2 summarizes the algorithm to solve (6.4). Note that satisfaction of the KKT conditions produces a local minimum; this is not a guarantee of convergence to the global optimum.

Under rather mild conditions the ADMM method can be shown to converge for all values of the algorithm parameter ρ in the Lagrange function $\mathcal{L}_\rho$ (6.7), cf. [11]. Small values of ρ lead to slow convergence; larger values give faster convergence but puts less emphasis on minimizing the residual for the NTF. For the convergence properties of ADMM and the impact of the parameter ρ see [37] and the references therein.

Algorithm 2 Tensor Dictionary Learning Algorithm

```
Input: Tensor of training image patches $\mathcal{Y} \in \mathbb{R}_+^{p\times t\times r}$, number of dictionary images $s$, tolerances $\rho, \epsilon > 0$.
Output: Tensor dictionary $\mathcal{D}_k \in \mathbb{R}_+^{p\times s\times r}$, tensor representation $\mathcal{H}_k \in \mathbb{R}_+^{s\times t\times r}$.
Initialization: Let the lateral slices of $\mathcal{U}$ be randomly selected training patches, let $\mathcal{V}$ be the identity tensor, let $\mathcal{H} = \mathcal{V}$, and let $\Lambda, \bar{\Lambda}$ be zero tensors of appropriate sizes.
for $k = 1, \dots$ do
    Update $\mathcal{D}_k, \mathcal{H}_k, \mathcal{U}_k, \mathcal{V}_k, \Lambda_k, \bar{\Lambda}_n$ by means of (6.8).
    if all stopping criteria (6.9) are met then
        Exit.
    end if
end for
```

6.3 Tomographic Reconstruction with Tensor Dictionary

Recall that a linear tomographic problem is often written $Ax \approx b$ with $A \in \mathbb{R}^{m\times n}$, where the vector x represents the unknown $M \times N$ image, the vector b is the inaccurate/noisy data, and the matrix A represents the forward tomography model. Since we assume that the vector x represents an image of absorption coefficients we impose a nonnegativity constraint on the solution.

Without loss of generality we, similar to the matrix-based formulation, assume that the size of the image is a multiple of the patch sizes in the dictionary. We partition the image into $q = (M/p)(N/r)$ non-overlapping patches of size $(M/p) \times (N/r)$, i.e., $X_j \in \mathbb{R}^{p\times r}$ for $j = 1, \dots, q$.

In the matrix-based formulation of the reconstruction problem in Chapter 4, once the patch dictionary is formed we write the image patches we want to recover (sub-vectors of the reconstructed image x) as conic combinations of the patch dictionary columns. The inverse problem then becomes one of recovering the expansion coefficients subject to non-negativity constraints (which produces a nonnegative x because the dictionary elements are nonnegative).

Here we define a similar reconstruction problem in our tensor-based formulation. We arrange all the patches X_j of the reconstructed image as lateral slices of a $p \times q \times r$ tensor $\mathcal{X}$, i.e.,

$$X_j = \texttt{squeeze}(\overrightarrow{\mathcal{X}_j}), \qquad \overrightarrow{\mathcal{X}_j} = \mathcal{X}(:j,:), \qquad j = 1, \dots, q.$$

Moreover, we assume that there exists a $s \times q \times r$ coefficient tensor $\mathcal{C}$ such that the image patches can be written as t-linear combinations of the patch dictionary elements, i.e.,

$$\mathcal{X} = \mathcal{D} * \mathcal{C} \qquad \Leftrightarrow \qquad \overrightarrow{\mathcal{X}_j} = \mathcal{D} * \overrightarrow{\mathcal{C}_j}, \quad j = 1, \dots, q, \tag{6.10}$$

where the tube fibers of $\overrightarrow{\mathcal{C}_j} = \mathcal{C}(:, j, :)$ can be considered as the expansion coefficients. In other words, we restrict our solution so that it is a t-linear combination of the dictionary images.

Then, similar to (6.1), each patch X_j in the reconstruction can be built from the matrices $\texttt{squeeze}(\overrightarrow{\mathcal{D}_i})$, $i = 1 \dots, s$:

$$X_j = \texttt{squeeze}\big(\mathcal{D} * \overrightarrow{\mathcal{C}_j}\big) = \sum_{i=1}^{s} \texttt{squeeze}\big(\overrightarrow{\mathcal{D}_i}\big)\,\texttt{circ}\Big(\texttt{squeeze}\big(\overrightarrow{\mathcal{C}_j}(i, 1, :)^{\mathrm{T}}\big)\Big). \tag{6.11}$$

Since the circulant matrices are not scalar multiples of the identity matrix, X_j is not a simple linear combination of the matrices $\texttt{squeeze}(\overrightarrow{\mathcal{D}_i})$.

Thus, we want to find a tensor $\mathcal{C}$ such that $\mathcal{X} = \mathcal{D} * \mathcal{C}$ solves the reconstruction problem, and to ensure a nonnegative reconstruction, we enforce non-negativity constraints on $\mathcal{C}$. Then we write the vectorized image as $x = \Pi\texttt{vec}(\mathcal{D} * \mathcal{C})$, where the permutation matrix Π ensures the correct shuffling of the pixels from the patches. Then our generic reconstruction problem takes the form

$$\min_{\mathcal{C}} \ \mathscr{L}_{\mathrm{rec}}\big(A\Pi\texttt{vec}(\mathcal{D} * \mathcal{C}), b\big) + \Phi_{\mathrm{sp}}(\mathcal{C}) + \Phi_{\mathrm{im}}(\mathcal{D} * \mathcal{C}), \qquad \mathcal{C} \geq 0. \tag{6.12}$$

The data fidelity is measured by the loss function $\mathscr{L}_{\mathrm{rec}}$, and regularization is imposed via Φ_{sp} which enforces a sparsity prior on $\mathcal{C}$, and Φ_{im} which enforces an image prior on the reconstruction. By choosing these three functions to be convex, we can solve (6.12) by means of convex optimization methods.

Our patches are non-overlapping because overlapping patches tend to produce blurring in the overlap regions of the reconstruction. Similar to the matrix-based formulation non-overlapping patches may give rise to block artifacts in the reconstruction, because the objective in the reconstruction problem does not penalize jumps across the values at the boundary of neighboring patches. To mitigate this type of jumps, we add the image penalty term $\Phi_{\mathrm{im}}(\mathcal{D} * \mathcal{C}) = \delta^2 \psi(\Pi\texttt{vec}(\mathcal{D} * \mathcal{C}))$ that discourages such artifacts, where δ is a regularization parameter, and the function ψ is defined by equation (4.9).

We consider two different ways to impose a sparsity prior on $\mathcal{C}$ in the form $\Phi_{\mathrm{sp}}(\mathcal{C}) = \mu\,\varphi_\nu(\mathcal{C})$, $\nu = 1, 2$, where μ is a regularization parameter and

$$\varphi_1(\mathcal{C}) = \frac{1}{q}\|\mathcal{C}\|_{\mathrm{sum}}, \qquad \varphi_2(\mathcal{C}) = \frac{1}{q}\big(\|\mathcal{C}\|_{\mathrm{sum}} + \|C\|_*\big), \tag{6.13}$$

in which the $sq \times r$ matrix C is defined as

$$C = \begin{pmatrix} \texttt{squeeze}(\overrightarrow{\mathcal{C}_1}) \\ \vdots \\ \texttt{squeeze}(\overrightarrow{\mathcal{C}_q}) \end{pmatrix}.$$

The first prior φ_1 corresponds to a standard sparsity prior in reconstruction problems. The second prior φ_2, which tends to produce a sparse and low-rank C, is inspired by a similar use in compressed sensing [38].

To summarize, we consider a reconstruction problem of the form

$$\begin{aligned} &\text{minimize}_{\mathcal{C}} && \tfrac{1}{2m}\|A\Pi\texttt{vec}(\mathcal{D} * \mathcal{C}) - b\|_2^2 + \mu\varphi_\nu(\mathcal{C}) + \delta^2\psi\big(\Pi\texttt{vec}(\mathcal{D} * \mathcal{C})\big) \\ &\text{subject to} && \mathcal{C} \geq 0, \end{aligned} \tag{6.14}$$

where μ and δ are regularization parameters. We note that (6.14) is a convex but non-differentiable optimization problem. It is solved using the software package TFOCS [8]. The implementation details are included in Appendix B.

We note that imposing the non-negativity constraint on the solution implies that each image patch X_j belongs to a closed set defined by

$$\mathsf{G} = \{\mathcal{D} * \overrightarrow{\mathcal{Z}} \mid \overrightarrow{\mathcal{Z}} \in \mathbb{R}_+^{s\times 1\times r}\} \subseteq \mathbb{R}_+^{p\times 1\times r}. \tag{6.15}$$

The set G is a cone, since for any $\overrightarrow{\mathcal{V}} \in \mathsf{G}$ and any nonnegative tube fiber $\mathbf{c} \in \mathbb{R}^{1\times 1\times r}$ the product $\overrightarrow{\mathcal{V}} * \mathbf{c}$ belongs to G. Clearly, if the dictionary $\mathcal{D}$ contains the standard basis that spans $\mathbb{R}_+^{p\times 1\times r}$ then G is equivalent to the entire nonnegative orthant $\mathbb{R}_+^{p\times 1\times r}$, and any image patch X_j can be reconstructed by a t-linear combination of dictionary basis images. However, in the typical case where G is a proper subset of $\mathbb{R}_+^{p\times 1\times r}$ then not all nonnegative images have an exact representation in G, leading to an approximation error.

6.4 Numerical Experiments

We conclude with computational tests to examine the tensor formulation. All experiments are run in MATLAB (R2014a) on a 64-bit Linux system. The reconstruction problems are solved using the software package TFOCS version 1.3.1 [8] and compared with results from the matrix-based approach in Chapter 4.

In Sections 6.4.1–6.4.2.2 we use the 1600×1200 high-resolution photo of peppers; from this image we extract the $p \times r$ training image patches. We also use the

200×200 ground-truth or exact image x^{exact} from Fig. 4.3. The exact image is not contained in the training set, so that we avoid committing an inverse crime. All the images are gray-level and scaled in the interval $[0, 1]$.

6.4.1 Dictionary Learning Experiments

Recall that the problem (6.4) is non-convex. To the best of our knowledge there is no global convergence results for non-convex optimization problems. To estimate how well the computed dictionary performs in practice, it should be validated in application. We first test the convergence of our tensor dictionary learning algorithm. Then we measure how tensor dictionary performs at sparsely encoding the training data given in $\mathcal{Y}$ as well as approximating similar images compared to the matrix dictionary learning algorithm from Chapter 4.

Patch sizes should be sufficiently large to capture the desired structure in the training images, but the computational cost of the dictionary learning increases with the patch size. The study of the patch size $p \times r$ and number s of elements in Chapter 4 shows that a reasonably large patch size gives a good trade-off between the computational work and the approximation error by the dictionary, and that the over-representation factor $s/(pr)$ can be smaller for larger patches. For these reasons, we have chosen $p = r = 10$ and (unless otherwise noted) $s = 300$ for both the dictionary learning and tomographic reconstruction studies. We extract $52,934$ patches from the high-resolution image and apply Algorithm 2 to learn the dictionary. The tensor dictionary $\mathcal{D}$ and the coefficient tensor $\mathcal{H}$ are $10 \times 300 \times 10$ and $300 \times 52934 \times 10$, respectively.

Convergence plots for $\lambda = 0.1$, 1, and 10 are shown in Fig. 6.6. For $\lambda = 10$ we put emphasis on minimizing the sparsity penalty, and after about 200 iterations we have reached convergence where the residual term dominates the objective function. For $\lambda = 0.1$ we put more emphasis on minimizing the residual term, and we need about 500 iterations to converge; now the objective function is dominated by the sparsity penalty.

Next we consider the approximation errors mentioned in the previous section. Following the study in Section 4.4.2, a way to bound these errors is to consider how well we can approximate the exact image x^{exact} with patches in the cone G (6.15) defined by the dictionary. Consider the q approximation problems for all blocks X_j^{exact}, $j = 1, 2, \ldots, q$, of the exact image:

$$\min_{\overrightarrow{\mathcal{C}_j}} \tfrac{1}{2} \left\| \mathcal{D} * \overrightarrow{\mathcal{C}_j} - X_j^{\text{exact}} \right\|_{\mathrm{F}}^2, \qquad \text{s.t.} \qquad \overrightarrow{\mathcal{C}_j} \geq 0.$$

If $\overrightarrow{\mathcal{C}_j}^{\star}$ denotes the solution to the jth problem, then $\mathrm{vec}(\mathcal{D} * \overrightarrow{\mathcal{C}_j}^{\star})$ is the best

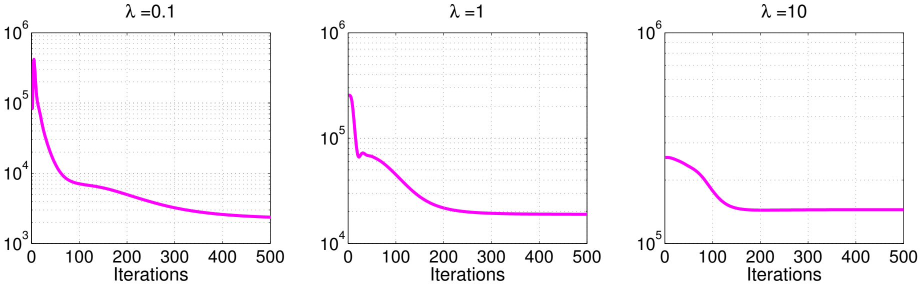


Figure 6.6: Convergence of Algorithm 2 for $\lambda = 0.1$, 1, and 10. We plot $\frac{1}{2}\|\mathcal{Y} - \mathcal{D} * \mathcal{H}\|_{\mathrm{F}}^2 + \lambda\|\mathcal{H}\|_{\mathrm{sum}}$ versus the number of iterations. Note the different scalings of the axes.

approximation in G of the jth block X_j^{exact}. We define the *mean approximation error* for our tensor formulation as

$$\mathrm{MAE} = \frac{1}{\sqrt{prq}} \sum_{j=1}^{q} \|\mathcal{D} * \vec{\mathcal{C}}_j^{\star} - X_j^{\mathrm{exact}}\|_{\mathrm{F}}.$$

Figure 6.7 shows how these MAEs vary with the number of nonzeros of $\mathcal{H}$ and H, as a function of λ, for both $s = 200$ and $s = 300$. This plot shows that for a given number of nonzeros in $\mathcal{H}$ or H we obtain approximately the same mean approximation error. In other words despite the fact that the $s \times t \times r$ tensor $\mathcal{H}$ has r times more degrees of freedom in the representation than the $s \times t$ matrix H, we do not need more nonzero values to represent our training images.

In Fig. 6.7 we note that for large enough λ both $\mathcal{H}$ and H consist entirely of zeros, in which case the dictionaries $\mathcal{D}$ and D are solely determined by the constraints. Hence, as λ increases the MAE settles at a value that is almost independent on λ.

To determine a suitable value of the regularization parameter λ in (6.4) we plot the residual norm $\|\mathcal{Y} - \mathcal{D} * \mathcal{H}\|_{\mathrm{F}}$ versus $\|\mathcal{H}\|_{\mathrm{sum}}$ for various $\lambda \in [0.1, 100]$ in Fig. 6.8. We define the optimal parameter to be the one that minimizes $\|\mathcal{H}\|_{\mathrm{sum}}^2 + \|\mathcal{Y} - \mathcal{D} * \mathcal{H}\|_{\mathrm{F}}^2$, which is obtained for $\lambda = 3.1623$, and we use this value throughout the rest of our experiments for the peppers test image.

Figure 6.9 shows examples of tensor and matrix dictionary elements/images, where lateral slices of the tensor dictionary and columns of the matrix dictionary are represented as images. The dictionary images are sorted according to increasing variance. The tensor and matrix dictionary images are different but they are visually similar.

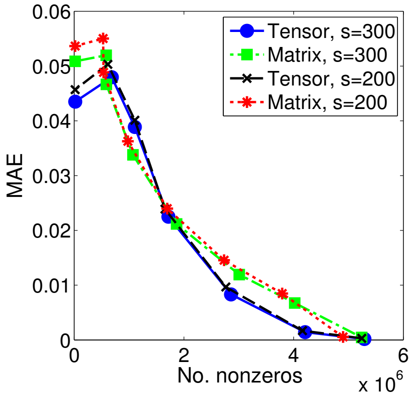


Figure 6.7: The mean approximation error MAE (6.4.1) for the tensor and matrix formulations versus the number of nonzeros of $\mathcal{H}$ and H, respectively, as functions of λ (small λ give a larger number of nonzeros).

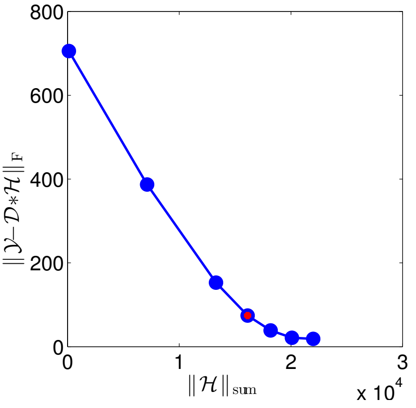


Figure 6.8: A trade-off curve for the tensor dictionary learning problem; the red dot denotes the value $\lambda = 3.1623$ that yields the smallest value of $\|\mathcal{H}\|_{\mathrm{sum}}^2 + \|\mathcal{Y} - \mathcal{D} * \mathcal{H}\|_{\mathrm{F}}^2$.

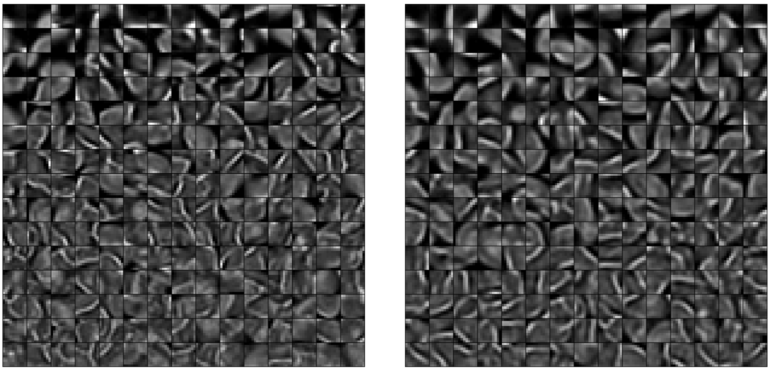


Figure 6.9: Examples of dictionary elements/images from the tensor formulatio (left) and the matrix formulation (right) with 10×10 patches and $\lambda = 3.1623$ and $s = 300$.

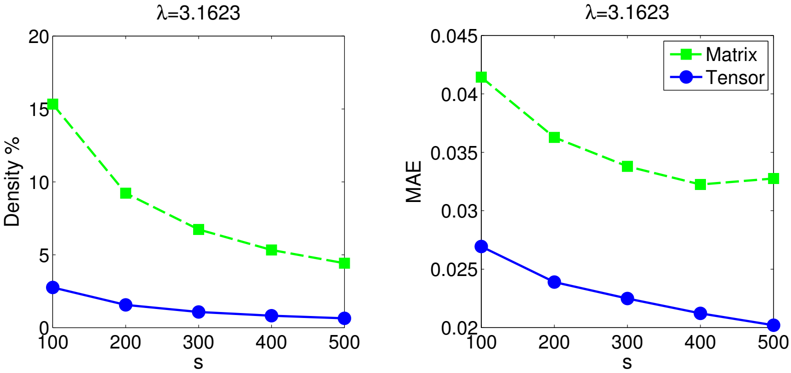


Figure 6.10: Dependence of the dictionary on the number of dictionary elements s, for both the tensor and matrix formulations. Left: the density of $\mathcal{H}$ and H. Right: the MAE associated with the dictionaries.

We conclude these experiments with a study of how the number s of dictionary elements influences the dictionary, for the fixed $\lambda = 3.1623$. Specifically, Fig. 6.10 shows how the density and the MAE varies with s in the range from 100 to 500. As we have already seen for $s = 300$ the density of $\mathcal{H}$ is consistently much lower than that of H, and it is also less dependent on s in the tensor formulation. We also see that the MAE for the tensor formulation is consistently lower for the tensor formulation: even with $s = 400, 500$ dictionary elements in the matrix formulation we cannot achieve the tensor formulation's low MAE for $s = 100$. These results confirm our intuition that the tensor formulation is better suited for sparsely representing the training image, because due to the ability of capturing repeating features we can use a much smaller dictionary.

6.4.2 Reconstruction Experiments

In this section we present numerical experiments for 2D tomographic reconstruction in few-projection and noisy settings. We perform two different experiments to analyze our algorithm: first we examine the role of different regularization terms and then we study the reconstruction quality in different tomography scenarios. We also present results using a more realistic test problem.

We consider parallel-beam geometry and the test problem is generated by means of the function `paralleltomo` from AIR Tools [47]. The exact data is generated by the forward model $b^{\text{exact}} = Ax^{\text{exact}}$, to which we add white Gaussian noise.

The accuracy of the reconstruction is measured by the relative 2-norm error

$$\text{RE} = \|x^{\text{exact}} - x\|_2 \,/\, \|x^{\text{exact}}\|_2 .$$

We also report the structural similarity index measure (SSIM) [114] (recall that a larger SSIM means a better reconstruction). We remind that the error is due to the combination of the approximation error, the error from the data, and the regularization error.

The parameters δ and μ in the reconstruction problem (6.14) both play a role in terms of regularization; to simplify (6.14) we set $\tau = \mu/q$. As described in Section 4.4.3, a nonnegative constraint in the reconstruction problem plays an extra role of regularization and therefore the reconstruction is not very sensitive to the regularization parameters δ and τ, hence they are chosen from a few numerical experiments such that a solution with the smallest error is obtained.

We compare our method with FBP, Tikhonov regularization, and TV. The FBP solution is computed using MATLAB's `iradon` function with the "Shepp-Logan" filter. The Tikhonov solution is obtained by solving (2.4) and the TV solution is obtained using (4.11). We solve the TV problem with the software TVReg [59]. The Tikhonov and TV regularization parameters are chosen to yield the smallest reconstruction error.

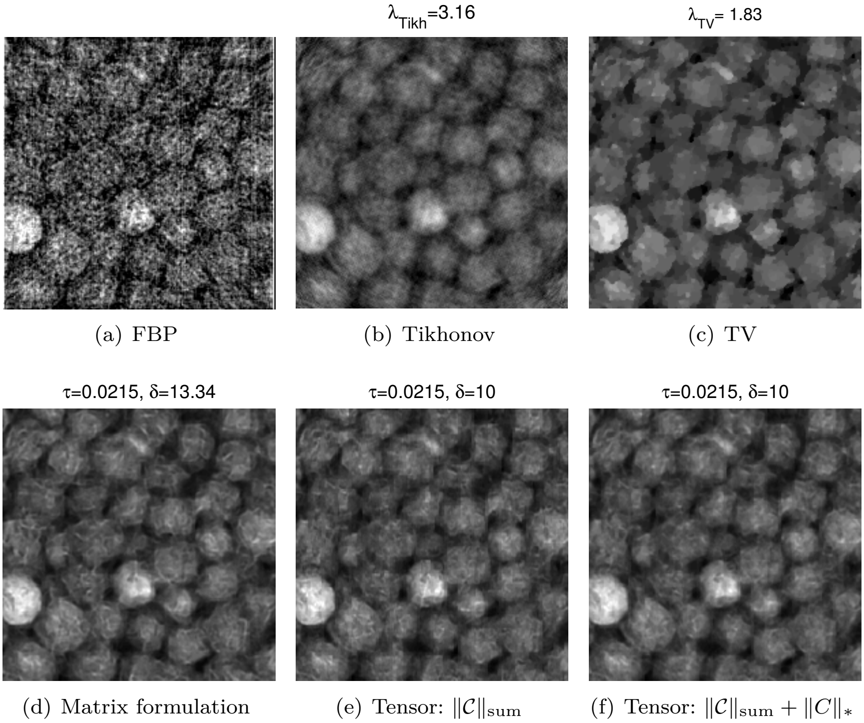
The computational bottleneck of the objective function evaluation in solving (6.14) is calculating $\mathcal{D} * \mathcal{C}$, where $\mathcal{D} \in \mathbb{R}^{p\times s\times r}$ and $\mathcal{C} \in \mathbb{R}^{s\times q\times r}$. Recall that the computation is done in the Fourier domain, and since $\log(r) < q, p$ the computational complexity of the t-product is $O(sqpr + s(p+q)r\log(r)) = O(sqpr)$ [49]. In the matrix formulation the computational bottleneck is the matrix multiplication $D\,\texttt{reshape}\big(\alpha, s, q\big)$ where $D \in \mathbb{R}^{pr\times s}$ and $\alpha \in \mathbb{R}^{sq\times 1}$, also with complexity $O(sqpr)$. This gives the tensor formulation an advantage, since we can use a much smaller s here, say, 2–3 times smaller than in the matrix formulation.

Since computation times vary between different computers, and since we did not pay specific attention to efficiency, we report the number of objective function evaluations returned by TFOCS. We stop the iterations when the relative change in the iteration vector is less than 10^{-7}. For the comparison to be fair, the starting point in all the computations is the zero vector/matrix of appropriate size.

6.4.2.1 Study of Regularization Terms

We solve the reconstruction problem using the exact image shown in Fig. 4.3. Moreover, we use 10×10 patches, $s = 300$, and $\lambda = 3.1623$. For the problems in this section we use $N_{\mathrm{p}} = 25$ projections, $N_{\mathrm{r}} = 283$ rays per projection, and 1% noise. We compare two different regularization terms in the reconstruction problem (6.14). The l_1-norm (sparsity) regularization $\|\mathcal{C}\|_{\mathrm{sum}}$ is similar to the l_1-norm regularization in the dictionary learning problem (6.4). The regularization term $\|\mathcal{C}\|_{\mathrm{sum}} + \|C\|_*$ results in coefficient tensors that are simultaneously low rank and sparse.

We compare the tensor reconstruction solution with the solutions obtained by the matrix formulation as well as FBP, Tikhonov regularization, and TV. The reconstructions are shown in Fig. 6.11. The corresponding relative errors, SSIM, and densities of $\mathcal{C}$ as well as the number of objective function evaluation are listed in Table 6.1. The table also lists the compressibility, defined as the percentage of coefficients which have values larger than 10^{-4}. Both the density and the compressibility show that we obtain very sparse representations of the reconstructed image.



(a) FBP (b) Tikhonov (c) TV

(d) Matrix formulation (e) Tensor: $\|\mathcal{C}\|_{\text{sum}}$ (f) Tensor: $\|\mathcal{C}\|_{\text{sum}} + \|C\|_*$

Figure 6.11: Comparison of the best solutions computed by different reconstruction methods. Subfigures (e) and (f) correspond to our new tensor formulation with two different regularization terms; we used $\lambda = 3.1623$.

Table 6.1: Comparison of the best solutions computed by different reconstruction methods. The bold numbers indicate the lowest iteration number, density of $\mathcal{C}$ and compression percentages, and highest SSIM measure.

Method	Itr.#	Density%	Compr.%	RE%	SSIM
FBP:	-	-	-	54.81	0.2981
Tikhonov reg.:	-	-	-	21.99	0.5010
TV:	-	-	-	**21.37**	0.4953
Matrix alg.	36843	12.53	5.31	22.00	0.4903
$\|\mathcal{C}\|_{\text{sum}}$ reg.	48787	**5.30**	**0.67**	22.21	0.4890
$\|\mathcal{C}\|_{\text{sum}} + \|C\|_*$ reg.	**8002**	10.27	3.26	21.55	**0.5061**

The FBP, Tikhonov, and TV methods fail to produce desirable reconstructions, although the 2-norm reconstruction error for the TV solution is slightly smaller than that for our solutions. The RE and SSIM do not tell the full story, and using a dictionary clearly improves recovering the *texture* of the image. The reconstructed images in Fig. 6.11 are similar across the matrix and tensor formulations; however, the results in Table 6.1 show that the tensor-formulation solution is more than 5 times more compressed and more than 2 times sparser than the matrix-formulation solution. Imposing both sparsity and low-rank regularization $\|\cdot\|_{\text{sum}} + \|\cdot\|_*$ produces a marginally more accurate solution with a denser representation.

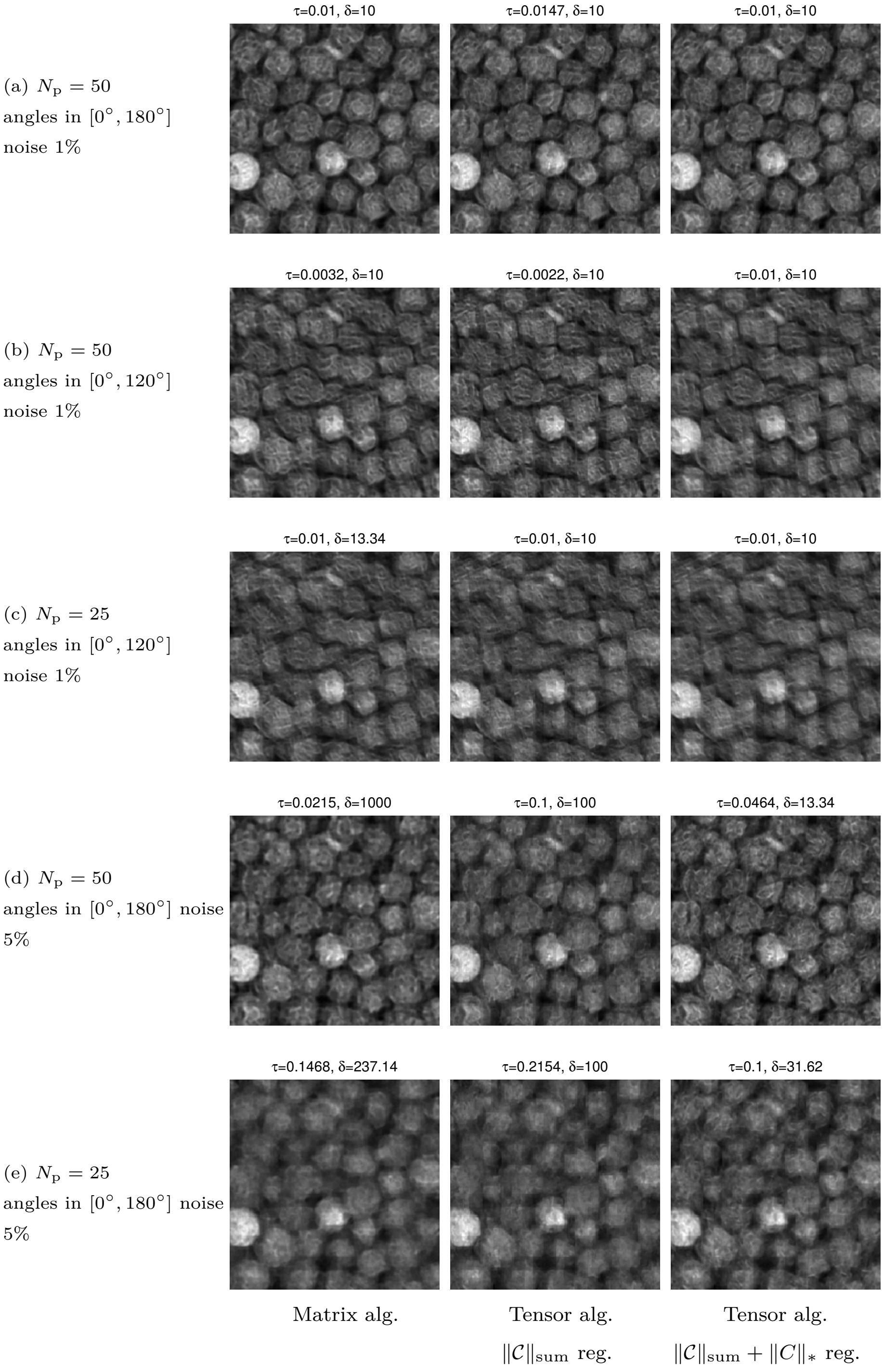
6.4.2.2 More Challenging Tomographic Reconstructions

To further study the performance and robustness of our tensor formulation approach, we consider problems with more noise in the data, or with projection angles in a limited range, still using the same test problem. Knowing that FBP, Tikhonov, and TV give unsatisfactory solutions for such problems, we only compare our method with the matrix formulation approach, and again we consider both regularization terms $\|\mathcal{C}\|_{\text{sum}}$ and $\|\mathcal{C}\|_{\text{sum}} + \|C\|_*$ in (6.14).

- First we compute a reconstruction with $N_{\text{p}} = 50$ projections, uniform angular sampling in $[0°, 180°]$ and with relative noise level 1%. In this scenario we use more projection data than in the previous section.
- Next we use 50 and 25 projections uniformly distributed in the limited range $[0°, 120°]$ and with relative noise level 1%.
- Finally we use 25 and 50 projections with uniform angular sampling in $[0°, 180°]$ and with relative noise level 5%, i.e., a higher noise level than above.

The reconstructions are shown in Fig. 6.12; they are similar across the tensor and matrix formulations, and pronounced artifacts have appeared from the limited angles and the higher noise level.

Table 6.2 lists the corresponding relative error, SSIM, density, and compressibility together with the iteration number. Comparison of Tables 6.1 and 6.2 reveal the same pattern. Algorithm 2 converges faster when imposing the combined regularization term $\|\mathcal{C}\|_{\text{sum}} + \|C\|_*$, and this choice also slightly improves the reconstruction in all scenarios. However, enforcing only the sparsity prior $\|\mathcal{C}\|_{\text{sum}}$ significantly reduce the representation redundancy, leading to a very sparse representation comparing to the matrix formulation. In the scenario



(a) $N_\mathrm{p} = 50$ angles in $[0^\circ, 180^\circ]$ noise 1%

(b) $N_\mathrm{p} = 50$ angles in $[0^\circ, 120^\circ]$ noise 1%

(c) $N_\mathrm{p} = 25$ angles in $[0^\circ, 120^\circ]$ noise 1%

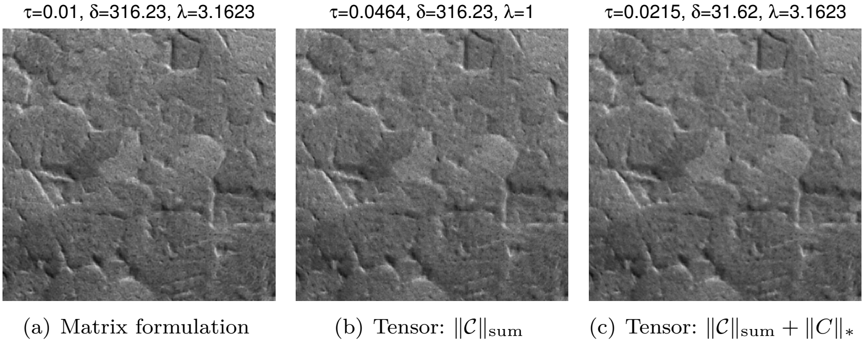
(d) $N_\mathrm{p} = 50$ angles in $[0^\circ, 180^\circ]$ noise 5%

(e) $N_\mathrm{p} = 25$ angles in $[0^\circ, 180^\circ]$ noise 5%

Figure 6.12: Reconstruction experiments from Section 6.4.2.2 with $\lambda = 3.1623$.

Table 6.2: Comparison of tensor and matrix formulation reconstructions in the experiments from Section 6.4.2.2. The methods “Matrix”, “Tensor-1”, and “Tensor-2” refer to the matrix-formulation algorithm and our new tensor-formulation algorithm with regularization terms $\|\mathcal{C}\|_{\text{sum}}$ and $\|\mathcal{C}\|_{\text{sum}} + \|C\|_*$. The bold numbers indicate the lowest iteration number, density and compression, and the highest SSIM.

Settings	Method	Itr.#	Density%	Compr.%	RE%	SSIM
$N_{\text{p}} = 50$	Matrix	41204	20.70	8.80	17.70	0.6368
in $[0°, 180°]$	Tensor-1	52801	**4.46**	**0.79**	17.19	0.6560
noise 1%	Tensor-2	**15676**	17.39	1.84	**16.82**	**0.6688**
$N_{\text{p}} = 50$	Matrix	48873	14.4575	9.43	22.77	0.5695
in $[0°, 120°]$	Tensor-1	61106	**9.08**	**0.98**	22.80	0.5818
noise 1%	Tensor-2	**16177**	23.81	2.07	**22.49**	**0.5883**
$N_{\text{p}} = 25$	Matrix	45775	100	5.91	25.46	0.4536
in $[0°, 120°]$	Tensor-1	59347	**26.00**	**0.73**	25.85	0.4544
noise 1%	Tensor-2	**17053**	27.49	2.29	**25.33**	**0.4676**
$N_{\text{p}} = 50$	Matrix	110322	50.17	8.02	22.05	0.4910
in $[0°, 180°]$	Tensor-1	40695	**8.97**	**0.74**	21.84	0.4846
noise 5%	Tensor-2	**10392**	14.64	1.72	**21.81**	**0.5107**
$N_{\text{p}} = 25$	Matrix	72139	45.51	6.29	24.69	0.3768
in $0°, 180°]$	Tensor-1	37072	**8.60**	**0.64**	25.12	0.3738
noise 5%	Tensor-2	**9076**	13.28	2.4829	**24.67**	**0.4041**



(a) Matrix formulation (b) Tensor: $\|\mathcal{C}\|_{\mathrm{sum}}$ (c) Tensor: $\|\mathcal{C}\|_{\mathrm{sum}} + \|C\|_*$

Figure 6.13: Reconstructions for the realistic test problem, computed with the matrix formulation (a) and the tensor formulation (b) + (c).

with 50 projections and 1% noise, where the regularization and perturbation errors are less dominating, the improvement in reconstructions by the tensor algorithm — compared to the matrix formulation — is more pronounced. Overall, we recommend the use of $\|\mathcal{C}\|_{\mathrm{sum}} + \|C\|_*$ which leads to the faster algorithm.

6.4.2.3 A Larger Test Problem

Tomography is a common tool in materials science to study the structure of grains in polycrystalline materials such as metals. The grain structure is sometimes known a priori in the form of training images. As test image in this experiment we use the high-resolution image of zirconium grains (produced by a scanning electron microscope) of dimension 760 × 1020 shown in the bottom of Fig. 4.11.

Training patches of size 10 × 10 are again extracted from the high-resolution image to learn matrix and tensor dictionaries size 100 × 300 and 10 × 300 × 10, respectively. To avoid committing inverse crime, we first rotate the high-resolution image and then extract the exact image of dimensions 520×520 — also shown in the bottom of Fig. 4.11.

We use a parallel-beam setting with $N_{\mathrm{p}} = 50$ projection angles in $[0°, 180°]$ and $N_{\mathrm{r}} = 707$ rays per projection, and again the matrix is computed by means of the function `paralleltomo` from AIR Tools [47]. We added 1% white Gaussian noise to the clean data. This problem is highly underdetermined with $m = 36750$ measurements and $n = 270400$ unknowns.

Table 6.3: Comparison of reconstruction in the realistic test problem, using the matrix and tensor formulations. The bold numbers indicate the lowest iteration number, density, and compression, and highest SSIM.

Method	Itr.#	Density%	Compr.%	RE%	SSIM
Matrix alg.	73961	48.61	6.86	14.90	0.4887
$\|\mathcal{C}\|_{\text{sum}}$ reg.	74310	**33.18**	**0.76**	15.23	0.4793
$\|\mathcal{C}\|_{\text{sum}} + \|C\|_*$ reg.	**24396**	38.78	3.17	**14.80**	**0.5035**

Figure 6.13 shows the reconstructed images with the matrix and tensor formulations. All regularization parameters are chosen empirically to give the smallest reconstruction errors. All three reconstructions are similar, since the reconstruction errors are dominated by the error coming from the regularization of the noisy data. More information is given in Table 6.3. Imposing the sparsity prior $\|\mathcal{C}\|_{\text{sum}}$ in the tensor formulation produces the sparsest representation. The solution is computed in fewer iterations with the $\|\mathcal{C}\|_{\text{sum}} + \|C\|_*$ regularization term while the reconstruction has a negligible improvement in terms of RE and SSIM. We conclude that our tensor algorithm is also well suited for more realistic tomographic problems.

6.5 Summary

In this chapter we presented the problem of dictionary learning in a tensor formulation and focused on solving the tomographic image reconstruction in the context of a t-product tensor-tensor factorization. The tensor dictionary learning problem is formulated as a non-negative sparse tensor factorization problem, and a regularized nonnegative reconstruction in the tensor-space defined by the t-product is computed. An algorithm based on the alternating direction method of multipliers (ADMM) is proposed for solving the tensor dictionary learning problem and, using the tensor dictionary, a convex optimization problem to recover the solution's coefficients in the expansion under the t-product is formulated.

Numerical experiments on the properties of the representation in the learned tensor dictionary in the context of tomographic reconstruction are presented. The dictionary-based reconstruction quality is superior to well know classical regularization schemes, e.g., filtered back projection and total variation, and the solution representation in terms of the tensor dictionary is more sparse compared to the similar matrix dictionary representations from Chapter 4. The experiments suggest that additional prior constraints improve representation and quality of the reconstruction.

Chapter 7

Conclusions and Remarks

It is often the case that prior to the tomographic process, a guess about the interior of the object under study is known. For example in medical imaging a collection of tomographic image reconstructions from former patients are available, or in material science the prior about the material's structure can be easily available in (higher resolution) photos taken from cross sections of another substance being cut, similar to the object under study.

It is well known that in CT with infinitely many rays, an image of the interior of the object can be reconstructed perfectly [93]. The problem arises when the number of projections are limited and a high resolution reconstruction is desired. Then, the use of training images to compensate for lack of data in tomographic experiments with few projections (to e.g., reduce dose) is reasonable.

This PhD thesis aims at providing an efficient and reliable computational framework for the use of training images – as samples of the prior for the solution – in tomographic image reconstruction. We use dictionary learning to construct a dictionary from a large data set of training patches extracted from the training images, and to obtain prototype elements and features from such training images. The dictionary is then incorporated in the reconstruction problem via a convex optimization formulation, as the prior for the solution. The computational large-scale optimization problem is then solved through first-order methods. The simplicity of this approach is that once the dictionary elements have been determined, the solution to the image reconstruction problem is obtained by an sparse approximation in the dictionary. Both matrix and tensor formulations have been developed to represent the images and formulate the dictionary learning and tomographic image reconstruction problems. Algorithms have been developed based on the ADMM method to compute the matrix and tensor dictionaries. We have tested the robustness and efficiency of our framework for the tomographic image reconstruction.

The dictionaries and sparsity priors play the regularization role for our underdetermined systems in the low-dose tomographic scenarios. The dictionaries and reconstructions are constrained to be nonnegative and it is shown that such nonnegativity constraints tend to have extra regularization effect on the solution. As a result, the reconstruction is not very sensitive to the regularization parameters and a reliable reconstruction can be obtained from a few experiments with the regularization parameters; however, a future work may include designing automatic parameter choice rules.

A formulation to compute the approximation error by the dictionary, i.e., how well we can represent our test images with the dictionary has been used to show how this error depends on the dictionary parameters.

Although the computational tests in this thesis were simulation based and not from real tomographic data, we tried to study the effect of different noise levels and dose constraints and we avoided committing inverse crimes. With this proposed approach, we are able to obtain sharper images with more reliable details; however, one should note that the tomographic solution in very underdetermined and noisy systems is solely determined by the prior and while a TV reconstruction have a “cartoonish” artifact for textural features the solution with dictionary learning approach may have other artifacts in these scenarios.

Our algorithm works with non-overlapping patches in the image and the same dictionary is used for all patches. To minimize blocking artifacts from reconstructing the image block by block, an additional regularization parameter is introduced in the tomographic reconstruction. Considering non-overlapping patches in the image compared to other proposed algorithms that apply a dictionary-based regularization, based on overlapping patches around every pixel in the image, reduces the computational complexity of the sparse approximation problem in the dictionary.

In this thesis and all the aforementioned previous works, the dictionary is fixed for all the overlapping/non-overlapping patches in the image, however images typically are non-stationary, i.e., the statistical feature properties may change from one location to the other in the image. It could be interesting to study how one can use dictionaries adopted to various statistical features of the tomographic images dependent on the location in the image. These non-stationary regions in the target image can be subdivided into stationary regions and different dictionaries for each subregion can be considered. Nevertheless, this correction needs a well known and trusted prior about the statistics of the features in the image.

Such an approach, using training images to learn a dictionary and solve a somehow big sparse approximation problem as well as looking for appropriate regularization parameters may seem computationally expensive. However a dictionary for a particular application need to be computed once and then can be used for several reconstructions. Furthermore, when time is not crucial in tomographic reconstruction, such as in material science applications, the amount of improvement in the reconstruction comparing to a fast but not reliable reconstruction technique like FBP is encouraging to use such a method.

A major contribution of this thesis is to formulate a tensor dictionary learning problem and tomographic image reconstruction around the new concepts from [63] and present algorithms to tackle such problems. It is shown that considering a tensor formulation over a matrix formulation significantly reduces the approximation error by the dictionary. As our tensor framework encourages, other imaging applications may also benefit from treating inherently two-dimensional images in their multidimensional format using the introduced tensor-tensor factorization and dictionary learning approach. So we suggest to further study the applications of the tensor dictionary in other imaging problems. In future work it would be also interesting to further study the tensor dictionary representation property using other products from the family of tensor-tensor products introduced, e.g., in [61].

In this thesis we have also studied the effects of the rotation and scale proportions of the training images to the unknown image; nonetheless, further studies are needed to understand all the difficulties and challenges of implementing such an approach in real applications.

We have focused on 2D tomographic problems; however, a more challenging problem that arises in real-world tomographic applications is in 3D. The challenge is that our training images will still be in 2D, because they typically come from pictures of slices of 3D objects. We hypothesize that the principles and model presented in this work can carry over to 3D reconstruction problems through multiplanar 2D as well, where a stack of two-dimensional reconstructed slices are acquired with the expansion of the system matrix and reformulation of the problem statement. Additional studies are necessary to investigate this hypothesis. The large-scale computing aspects will become more pronounced in 3D reconstructions.

Appendix A

ADMM

Our solution of the dictionary learning problems (4.1) and (6.4) relies on the ADMM, which has become very popular in the recent years [11]. The classic alternating direction method (ADMM) solves structured convex problems in the form of

$$\min_{x\in\mathcal{X},z\in\mathcal{Z}} f(x)+g(z) \tag{A.1}$$

$$\text{s.t. } Ax+Bz\;=\;c$$

where f and g are convex functions defined on closed subsets $\mathcal{X}$ and $\mathcal{Z}$ of a finite-dimensional space, respectively, and A, B and c are matrices and vector of appropriate sizes. We form the augmented Lagrangian associated with (A.1)

$$\mathcal{L}_\rho(x,z,y)\;=\;f(x)+g(z)+y^{\mathrm{T}}(Ax+Bz-c)+(\frac{\rho}{2})\|Ax+Bz-c\|_2^2,$$

where y is a Lagrangian multiplier vector and $\rho>0$ is a penalty parameter. ADMM performs minimization with respect to x and z alternatively, followed by the dual variable update of y, i.e., at each iteration k:

$$x^{k+1}\;:=\;\arg\min_x \mathcal{L}(x,z^k,y^k), \tag{A.2a}$$

$$z^{k+1}\;:=\;\arg\min_z \mathcal{L}(x^{k+1},z,y^k), \tag{A.2b}$$

$$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c), \tag{A.2c}$$

ADMM can be slow to converge to high accuracy, however, in practice ADMM often converges to a modest accuracy within a few tens of iterations [11].

Appendix B

Reconstruction with Tensor Dictionary Via TFOCS

The reconstruction problem (6.14) is a convex, but $\|\mathcal{C}\|_{\mathrm{sum}}$ and $\|C\|_*$ are not differentiable which rules out conventional smooth optimization techniques. The TFOCS software [8] provides a general framework for solving convex optimization problems, and the core of the method computes the solution to a standard problem of the form

$$\text{minimize} \quad l(A(x) - b) + h(x), \tag{B.1}$$

where the functions l and h are convex, A is a linear operator, and b is a vector; moreover l is smooth and h is non-smooth.

To solve problem (6.14) by TFOCS, it is reformulated as a constrained linear least squares problem:

$$\min_{\mathcal{C}} \frac{1}{2} \left\| \begin{pmatrix} \frac{1}{\sqrt{m}} A \\ \frac{\delta}{\vartheta} L \end{pmatrix} \Pi \mathtt{vec}(\mathcal{D} * \mathcal{C}) - \begin{pmatrix} b \\ 0 \end{pmatrix} \right\|_2^2 + \mu \, \varphi_\nu(\mathcal{C}) \qquad \text{s.t.} \qquad \mathcal{C} \geq 0, \tag{B.2}$$

where $\vartheta = \sqrt{2(M(M/p-1) + N(N/r-1))}$. Referring to (B.1), $l(\cdot)$ is the squared 2-norm residual and $h(\cdot) = \mu \, \varphi_\nu(\cdot)$.

The methods used in TFOCS require computation of the proximity operators of the non-smooth function h. The proximity operator of a convex function is a natural extension of the notion of a projection operator onto a convex set [25].

Let $f = \|\mathcal{C}\|_{\text{sum}} = \|C\|_{\text{sum}}$ and $g = \|C\|_*$ be defined on the set of real-valued matrices and note that $\text{dom} f \cap \text{dom}\, g \neq \emptyset$. For $Z \in \mathbb{R}^{m\times n}$ consider the minimization problem

$$\text{minimize}_X \; f(X) + g(X) + \frac{1}{2}\|X - Z\|_{\text{F}}^2 \tag{B.3}$$

whose unique solution is $X = \text{prox}_{f+g}(Z)$. While the prox operators for $\|C\|_{\text{sum}}$ and $\|C\|_*$ are easily computed, the prox operator of the sum of two functions is intractable. Although the TFOCS library includes implementations of a variety of prox operators — including norms and indicator functions of many common convex sets — implementation of prox operators of the form $\text{prox}_{f+g}(\cdot)$ is left out. Hence we compute the prox operator for $\|\cdot\|_{\text{sum}} + \|\cdot\|_*$ iteratively using a Dykstra-like proximal algorithm [25], where prox operators of $\|\cdot\|_{\text{sum}}$ and $\|\cdot\|_*$ are consecutively computed in an iterative scheme.

Let $\tau = \mu/q \geq 0$. For $f(X) = \tau\|X\|_{\text{sum}}$ and $X \geq 0$, prox_f is the one-sided elementwise shrinkage operator

$$\text{prox}_f(X)_{i,j} = \begin{cases} 0, & X_{i,j} \geq \tau \\ X_{i,j} - \tau, & |X_{i,j}| \leq \tau \\ 0, & X_{i,j} \leq -\tau \end{cases}$$

The proximity operator of $g(X) = \tau\|X\|_*$ has an analytical expression via the singular value shrinkage (soft threshold) operator

$$\text{prox}_g(X) = U\text{diag}(\sigma_i - \tau)\, V^T,$$

where $X = U\Sigma V^T$ is the singular value decomposition of X [16]. The computation of $\tau\|C\|_*$ can be done very efficiently since C is $sq \times r$ with $r \ll sq$.

The iterative algorithm which computes an approximate solution to prox_{f+g} is given in Algorithm 3. Every sequence X_k generated by Algorithm 3 converges to the unique solution prox_{f+g} of problem (B.3) [25].

Algorithm 3 Dykstra-Like Proximal Algorithm

```
**Input**: The matrix $Z$
**Output**: $\mathrm{prox}_{f+g}(Z)$
**Initialization**: Set $X_1 = Z$ and set $P_1$ and $Q_1$ to zero matrices of appropriate sizes.
**for** $k = 1, 2, \ldots$ **do**
    $Y_k = \mathrm{prox}_g(X_k + P_k)$
    $P_{k+1} = X_k + P_k - Y_k$
    $X_{k+1} = \mathrm{prox}_f(Y_k + Q_k)$
    $Q_{k+1} = Y_k + Q_k - X_{k+1}$
    **if** $\|Y_k - X_{k+1}\|_{\mathrm{F}} < 10^{-3}$ **then**
        Exit
    **end if**
**end for**
```

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[2] S. Aja-Fernandez, R. de Luis Garcia, D. Tao, and X. (Eds.) Li. *Tensors in image processing and computer vision, advances in pattern recognition.* Springer, New York, 2009.

[3] L. Bar and G. Sapiro. Hierarchical invariant sparse modeling for image analysis. *Proceedings - International Conference on Image Processing*, pages 2397–2400, 2011.

[4] Q. Barthelemy, A. Larue, A. Mayoue, D. Mercier, and J.I. Mars. Shift boolean and 2d rotation invariant sparse coding for multivariate signals. *IEEE Transactions on Signal Processing*, 60(4):1597–1611, 2012.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[6] S.R. Becker. *Practical compressed sensing: modern data acquisition and signal processing.* PhD thesis, California Institute of Technology, Pasadena, California, USA, 2011.

[7] S.R. Becker, J. Bobin, and E.J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[8] S.R. Becker, E.J. Candès, and M.C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

[9] M. Bertero and P. Boccacci. *Introduction to inverse problems in imaging.* IOP Publ, 1998.

[10] J. Bian, J.H. Siewerdsen, X. Han, E.Y. Sidky, J.L. Prince, C.A. Pelizzari, and X. Pan. Evaluation of sparse-view reconstruction from flat-panel-detector cone-beam ct. *Physics in Medicine and Biology*, 55(22):6575–6599, 2010.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

[12] J.P. Boyle and R.L. Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. *Advances in Order Restricted Statistical Inference, Lecture Notes in Statistics.*

[13] K. Braman. Third-order tensors as linear operators on a space of matrices. *Linear Algebra and Its Applications*, 433(7):1241–1253, 2010.

[14] A.M. Bruckstein, D.L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.

[15] T.M. Buzug. *Computed tomography: From photon statistics to modern cone-beam CT.* Springer, 2010.

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

[17] C.F. Caiafa and A. Cichocki. Multidimensional compressed sensing and their applications. *Willey Interdisciplinary Reviews-Data Mining and Knowledge Discovery*, 3(6):355–380, 2013.

[18] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and vision*, 20(1-2):89–97, 2004.

[19] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710, 2007.

[20] S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[21] S. Chen, H. Liu, P. Shi, and Y. Chen. Sparse representation and dictionary learning penalized image reconstruction for positron emission tomography. *Physics in Medicine and Biology*, 60(2):807–823, 2015.

[22] Y. Chen, X. Yin, L. Shi, H. Shu, L. Luo, J.-L. Coatrieux, and C. Toumoulin. Improving abdomen tumor low-dose ct images using a fast dictionary learning based processing. *Physics in Medicine and Biology*, 58(16):5803–5820, 2013.

[23] Y.-C. Chen, C.S. Sastry, V.M. Patel, P.J. Phillips, and R. Chellappa. In-plane rotation and scale invariant clustering using dictionaries. *IEEE Transactions on Image Processing*, 22(6):2166–2180, 2013.

[24] A. Cichocki, R. Zdunek, A.H. Phan, and S. Amari. *Non-negative matrix and tensor factorizations: applications to exploratory multiway data analysis and blind source separation.* Wiley, 2009.

[25] P.L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 49:185–212, 2011.

[26] D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on Information Theory*, 52(1):6–18, 2006.

[27] D.L. Donoho and I.M. Johnstone. Ideal spatial adaption by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[28] D.L. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? *Advances in Neural Information processing Systems*, 16:1141–1148, 2004.

[29] G. Duan, H. Wang, Z. Liu, J. Deng, and Y.-W. Chen. K-cpd: Learning of overcomplete dictionaries for tensor sparse coding. *IEEE 21st International Conference on Pattern Recognition (ICPR)*, pages 493–496, 2012.

[30] P.P.B. Eggermont, G.T. Herman, and A. Lent. Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear Algebra and Its Applications*, 40:37–67, 1981.

[31] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing.* Springer, 2010.

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

[33] K. Engan, S. Aase, and J. Husøy. Multi-frame compression: theory and design. *Signal Processing*, 80(10):2121–2140, 2000.

[34] V. Etter, I. Jovanovic, and M. Vetterli. Use of learned dictionaries in tomographic reconstruction. *Wavelets and Sparsity XIV*, 8138(1), 2011.

[35] A. Faridani. Introduction to the mathematics of computed tomography. In *Inside Out: Inverse Problems and Applications*, pages 1–46. MSRI Publications, 2003.

[36] M. Figueiredo, R.D. Nowak, and S.J. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):586–597, 2007.

[37] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2015.

[38] M. Golbabaee and P. Vandergheynst. Hyperspectral image compressed sensing via low-rank and joint-sparse matrix recovery. *Icassp, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 2741–2744, 2012.

[39] G.H. Golub and C.F. van Loan. *Matrix computations.* Johns Hopkins University Press, 1983.

[40] R. Gordon, R. Bender, and G.T. Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of Theoretical Biology*, 29(3):471–481, 1970.

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

[42] P. Grangeat. *Tomography.* Wiley, 2009.

[43] J. Hadamard. *Lectures on Cauchy's problem in linear partial differential equations.* Dover Publications, 1952.

[44] P.C. Hansen. Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review*, 34(4):561–580, 1992.

[45] P.C. Hansen. *Rank-deficient and discrete ill-posed problems.* SIAM Philadelphia, 1996.

[46] P.C. Hansen. *Discrete inverse problems: Insight and Algorithms (Fundamentals of Algorithms).* SIAM Philadelphia, 2010.

[47] P.C. Hansen and M. Saxild-Hansen. Air tools - a matlab package of algebraic iterative reconstruction methods. *Journal of Computational and Applied Mathematics*, 236(8):2167–2178, 2012.

[48] Horesh L. Hao, N. and M.E Kilmer. *Nonnegative tensor decomposition. In: Compressed sensing & sparse filtering.* 2014, Springer, Berlin.

[49] N. Hao, M.E. Kilmer, K. Braman, and R.C. Hoover. Facial recognition using tensor-tensor decompositions. *SIAM Journal on Imaging Sciences*, 6(1):437–463, 2013.

[50] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: Data mining, inference, and prediction.* Springer, 2009.

[51] G.T. Herman. *Fundamentals of computerized tomography: image reconstruction from projections.* Springer, 2009.

[52] K. Hotta, T. Kurita, and T. Mishima. Scale invariant face recognition method using spectral features of log-polar image. *Proceedings of the SPIE - the International Society for Optical Engineering*, 3808:33–43, 1999.

[53] P.O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5(9):1457, 2004.

[54] http://multibandtexture.recherche.usherbrooke.ca/normalized_brodatz.html. 2015.

[55] http://www.one-eighty-degrees.com/service/microstructural investigations. 2015.

[56] Y. Huang, J. Paisley, Q. Lin, X. Ding, X. Fu, and X.-P. Zhang. Bayesian nonparametric dictionary learning for compressed sensing mri. *IEEE Transactions on Image Processing*, 23(12), 2014.

[57] Belgium iMinds Vision Lab, Universiteit Antwerpen and The Netherlands CWI, Amsterdam. Astra tomography toolbox.

[58] K. Jafari-Khouzani and H. Soltanian-Zadeh. Radon transform orientation estimation for rotation invariant texture analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):1004–1008, 2005.

[59] 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. *BIT Numerical Mathematics*, 52(2):329–356, 2012.

[60] J.S Jørgensen. *Sparse image reconstruction in computed tomography.* PhD thesis, 2013.

[61] E. Kernfelda, M.E Kilmer, and Aeron S. Tensor-tensor products with invertible linear transforms. *Accepted to be published in Linear Algebra and its Applications*, 2015.

[62] HAL Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122, 2000.

[63] M.E. Kilmer, K. Braman, N. Hao, and R.C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[64] M.E. Kilmer and C.D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and Its Applications*, 435(3):641–658, 2011.

[65] T.G. Kolda and B.W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[66] V. Kolehmainen, M. Lassas, K. Niinimaäki, and S. Siltanen. Sparsity-promoting bayesian inversion. *Inverse Problems*, 28(2):025005, 2012.

[67] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2):349–396, 2003.

[68] Peter Kuchment. *The radon transform and medical imaging.* SIAM, 2014.

[69] S.J. LaRoque, E.Y. Sidky, and X. Pan. Accurate image reconstruction from few-view and limited-angle data in diffraction tomography. *Journal of The Optical Society of America A-Optics Image Science and Vision*, 25(7):1772–1782, 2008.

[70] D.D. Lee and H.S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[71] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*, pages 801–808, 2007.

[72] S. Li, L. Fang, and H. Yin. An efficient dictionary learning algorithm and its application to 3-d medical image denoising. *IEEE Transactions on Biomedical Engineering*, 59(2):417–427, 2012.

[73] H. Y. Liao and G. Sapiro. Sparse representations for limited data tomography. *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano To Macro, Proceedings*, pages 4541261, 1375–1378, 2008.

[74] B. Liu, H. Yu, S.S. Verbridge, L. Sun, and G. Wang. Dictionary-learning-based reconstruction method for electron tomography. *Scanning*, 36(4):377–383, 2014.

[75] Q. Liu, D. Liang, Y. Song, J. Luo, Y. Zhu, and W. Li. Augmented lagrangian-based sparse representation method with dictionary updating for image deblurring. *SIAM Journal on Imaging Sciences*, 6(3):1689–1718, 2013.

[76] I. Loris. On the performance of algorithms for the minimization of l1-penalized functionals. *Inverse Problems*, 25(3):035008, 2009.

[77] D.G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the IEEE International Conference on Computer Vision*, 2:1150–1157, 1999.

[78] J. Mairal. *Sparse coding for machine learning, image processing and computer vision.* PhD thesis, Ecole Normale Superieure de Cachan, 2010.

[79] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

[80] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[81] J. Mairal, G. Sapiro, and M. Elad. Multiscale sparse image representation with learned dictionaries. *IEEE International Conference on Image Processing, ICIP 2007.*

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

[83] S.G. Mallat and Z.F. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[84] A. Mirone, E. Brun, and P. Coan. A dictionary learning approach with overlap for the low dose computed tomography reconstruction and its vectorial application to differential phase tomography. *PLOS ONE*, 9(12), 2014.

[85] J.L. Mueller and S. Siltanen. *Linear and nonlinear inverse problems with practical applications.* SIAM, 2012.

[86] J.F. Murray and K. Kreutz-Delgado. Learning sparse overcomplete codes for images. *Journal of VLSI Signal Processing Systems for Signal Image and Video Technology*, 45(1-2):97–110, 2006.

[87] A.Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. *ICML Proceedings, Twenty-First International Conference on Machine Learning*, pages 615–622, 2004.

[88] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[89] B. Ophir, M. Lustig, and M. Elad. Multi-scale dictionary learning using wavelets. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):1014–1024, 2011.

[90] Batenburg K.J. Palenstijn, W.J. and J. Sijbers. The astra tomography toolbox. In *Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engineering*, pages 1139–1145. CMMSE, 2013.

[91] T. Park and G. Casella. The bayesian lasso. *Journal of The American Statistical Association*, 103(482):681–686, 2008.

[92] Y.C. Pati, R. Rezaiifar, P.S. Krishnaprasad, and A. Singh. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *Conference Record - Asilomar Conference on Signals, Systems, and Computers*, pages 40–44 vol.1, 1993.

[93] J. Radon. Über die bestimmung von funktionen durch ihre integralwerte längs gewisser mannings-faltigkeiten. *Math.-Phys.*, 69:262–277, 1917.

[94] S. Ravishankar and Y. Bresler. Mr image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Transactions on Medical Imaging*, 30(5):1028–1041, 2011.

[95] F. Roemer, G. Del Galdo, and M. Haardt. Tensor-based algorithms for learning multidimensional separable dictionaries. *Icassp, IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3963–3967, 2014.

[96] R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58, 2011.

[97] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.

[98] Soltani S. Studies of sensitivity in the dictionary learning approach to computed tomography: simplifying the reconstruction problem, rotation, and scale. *DTU Compute Technical Report 2015-4*, 2015.

[99] Soltani S., M.S. Andersen, and P.C. Hansen. Tomographic image reconstruction using training images. *submitted to Journal of Computational and Applied Mathematics*, 2015.

[100] Soltani S., M.E. Kilmer, and P.C. Hansen. A tensor-based dictionary learning approach to tomographic image reconstruction. *submitted to BIT Numerical Mathematics*, 2015.

[101] E.Y. Sidky, C.-M. Kao, and X. Pan. Accurate image reconstruction from few-views and limited-angle data in divergent-beam ct. *Journal of X-Ray Science and Technology*, 14(2):119–139, 2006.

[102] S. Siltanen, V. Kolehmainen, S. Järvenpää, J.P. Kaipio, P. Koistinen, M. Lassas, J. Pirttilä, and E. Somersalo. Statistical inversion for medical x-ray tomography with few radiographs: I. general theory. *Physics in Medicine and Biology*, 48(10):1437–1463, 2003.

[103] L.N. Smith and M. Elad. Improving dictionary learning: multiple dictionary updates and coefficient reuse. *IEEE Signal Processing Letters*, 20(1):79–82, 2013.

[104] H.H.B. Sørensen and P.C. Hansen. Multicore performance of block algebraic iterative reconstruction methods. *SIAM Journal on Scientific Computing*, 36(5):C524–46, 2014.

[105] D. Strong and T. Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems*, 19(6):S165–S187, 2003.

[106] Shengqi Tan, Yanbo Zhang, Ge Wang, Xuanqin Mou, Guohua Cao, Zhifang Wu, and Hengyong Yu. Tensor-based dictionary learning for dynamic tomographic reconstruction. *Physics in Medicine and Biology*, 60(7):2803–2818, 2015.

[107] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of The Royal Statistical Society Series B-Methodological*, 58(1):267–288, 1996.

[108] A.N. Tikhonov and V.Y. Arsenin. *Solutions of ill-posed problems.* V.H. Winston & Sons, 1977.

[109] I. Tošić, I. Jovanović, P. Frossard, M. Vetterli, and N. Duric. Ultrasound tomography with learned dictionaries. *ICASSP Proceedings, IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5502–5505, 2010.

[110] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of The IEEE*, 98(6):948–958, 2010.

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

[112] E. Van Den Berg and M.P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2009.

[113] J. Velikina, S. Leng, and G.H. Chen. Limited view angle tomographic image reconstruction via total variation minimization. *Progress in Biomedical Optics and Imaging*, 6510(2):651020, 2007.

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

[115] M. Wieczorek, J. Frikel, J. Vogel, E. Eggl, F. Kopp, P.B. Noel, F. Pfeiffer, L. Demaret, and T. Lasser. X-ray computed tomography using curvelet sparse regularization. *Medical Physics*, 42(4):1555–1565, 2015.

[116] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang. Low-dose x-ray ct reconstruction via dictionary learning. *IEEE Transactions on Medical Imaging*, 31(9):1682–1697, 2012.

[117] Y. Xu, W. Yin, Z. Wen, and Y. Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, 7(2):365–384, 2012.

[118] H. Zayyani and M. Babaie-Zadeh. Thresholded smoothed-1(0)(sl0) dictionary learning for sparse representations. *IEEE International Conference on Acoustics, Speech, and Signal Processing, Proceedings*, pages 1825–1828, 2009.

[119] R. Zdunek and A. Cichocki. Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems. *Computational Intelligence and neuroscience*, page 939567, 2008.

[120] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multi-linear data completion and denoising based on tensor-svd. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3842–3849, 2014.

[121] B. Zhao, H. Ding, Y. Lu, G. Wang, J. Zhao, and S. Molloi. Dual-dictionary learning-based iterative image reconstruction for spectral computed tomography application. *Physics in Medicine and Biology*, 57(24):–, 2012.

[122] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing, IEEE Trans Image Process*, 21(1):130–144, 2012.

[123] S. Zubair and W. Wang. Tensor dictionary learning with sparse tucker decomposition. *IEEE 18th International Conference on Digital Signal Processing*, page 6622725, 2013.