Analysis and Synthesis Sparse Modeling Methods in Image Processing

Ron Rubinstein

Analysis and Synthesis Sparse Modeling Methods in Image Processing

Research Thesis

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Ron Rubinstein

Submitted to the Senate of the Technion —
Israel Institute of Technology

Elul 5711 Haifa September 2011

The Research Thesis Was Done Under The Supervision of Prof. Michael Elad
in the Faculty of Computer Science at the Technion

Acknowledgements

This thesis has been made possible thanks to the generous help of many people, and I am delighted and honored to acknowledge them here.

My deepest gratitude goes to my teacher and mentor, Prof. Michael Elad. Miki, it has been an exceptional privilege to work with such a devoted, inspiring, knowledgeable, understanding, and insightful adviser. No words can express my appreciation for your outstanding support, invaluable advice, relentless encouragement, and unwavering guidance throughout my academic path, and for all these and more I sincerely thank you.

Special thanks go to Dr. Michael Zibulevsky, with whom it has been an incredible opportunity to collaborate. I am indebted to Dr. Zibulevsky for numerous fruitful discussions and enlightening ideas, and I dearly thank him for his generous assistance, perceptive insights, and warm companionship during my years at the Technion.

I would also like to use this opportunity and thank some of the people from whom I have learned so much over the years, and to whom I owe much of what I know today. My deepest thanks and appreciation go to Prof. Alfred Bruckstein, Prof. Ron Kimmel, Prof. Avraham Sidi, Prof. Irad Yavneh, Prof. Shmuel Peleg and Prof. Naftali Tishby, all of whom have endowed me with invaluable knowledge and have exhilarated my enthusiasm for the art.

I would further like to send my heartfelt thanks to Yaron Honen and Yana Katz from the Geometric Image Processing lab, for their exemplary technical assistance and invaluable support over the years. Additional thanks go to Prof. David Malah and Nimrod Peleg from the Signal and Image Processing Lab, for

the fruitful research and collaboration opportunities. Finally, special thanks go to our graduate studies secretary, Yardena Kolet, for her dedicated assistance and infinite support throughout my studies at the Technion.

I would like to extend my thanks to the many friends I have met during my years at the Technion, and who have made it a social and intellectual pleasure to attend. I would like to especially mention Roee Engelberg, Yaniv Hamo, Keren Ouaknine, Svetlana Raboy, and Irenne Zbarsky, each of whom has become a cherished friend, and has had a meaningful and lasting influence on my life. Particular thanks go to my friend and colleague Ori Bryt for the substantial help, constructive collaborations, and fruitful discussions, both academic and others.

It is a distinct and personal pleasure to thank my close and dear friend Jacob Kagan for the many years of devoted friendship, immeasurable assistance, insightful discussions, and thorough and single-hearted support. My sincerest gratitude and appreciation go to Jacob. One could not ask for a better friend.

Finally, this thesis would not have been possible were it not for the most devoted, supportive, and caring family. To my parents, Beverly and Israel, I wish to send my deepest gratitude, appreciation and affection for the unconditional support, persistent encouragement, inspirational advice, and endless love. You have been a guide, a motivator, a comfort, and a role model, and for all these and more I am deeply indebted to you. To my three wonderful sisters, Yael, Tali, and Tammi, thank you for your support, understanding, appreciation, and for the many joyous moments. Many thanks to my dear aunts Nitza and Nurit, for the great advice, encouragement, and affection, and to my beloved grandparents, Lucy and Elliot, for their faith in me, support, and undivided love. Sincere thanks to all my family members whom I did not name in person — my love goes to you all.

The generous financial help of the Technion and the Israeli Ministry of Science is gratefully appreciated

Dedicated to my loving and supporting family
who have always been there for me
and I hope will be there for a long, long time

Contents

List of Figures

List of Tables

Abstract

Signal models are a cornerstone of contemporary signal and image processing methodology. Of these models, analysis and synthesis sparse representation models have been particularly successful in a wide range of applications. Both models take a decompositional approach, and describe signals in terms of an underlying set, or *dictionary*, of elementary signals known as *atoms*. The analysis approach describes signals in terms of their inner products with the dictionary atoms, whereas the synthesis model takes a reverse approach and describes signals as linear combinations of atoms. The driving force behind both models is *sparsity* — the rapid decay of the representation coefficients over the dictionary. The two models have been found effective in a wide array of signal and image processing tasks, and lead to state-of-the-art results in applications such as denoising, demosaicing, compression, inpainting, upscaling, compressive sensing, and more.

This thesis studies several aspects of the analysis and synthesis modeling paradigms. We begin with the question of the *relation* between the two dictionary-based models, which arises due to the mathematical resemblance between the two. We show, through geometrical reasoning, that contrary to the mathematical similarity, the two approaches are in fact generally distinct, with a significant gap separating the two. The results of this study ignite a renewed interest in the analysis formulation, and provide several insights about the model.

In the main part of the thesis we focus on the core component of these models — the dictionary. The dictionary represents the materialization of all our knowledge about the signal behavior, and its choice determines the success of the entire model. We describe the two main disciplines of designing such dictionaries — harmonic analysis and machine learning — and discuss the recent trend of converging the two through parametric dictionaries. We develop a specific parametric dictionary which we name the *sparse dictionary*, and which provides a simple and expressive structure for designing adaptable and efficient dictionaries. Among the applications of this new structure, we describe a complete system for compressing generic images, which is unique in that it encodes each input image over a specifically-trained dictionary, sent as part of the compressed stream.

In the last part of this thesis, we return to the analysis formulation and consider the problem of dictionary training for analysis models. This is a relatively recent field, motivated by the theoretical results mentioned above, as well as the widespread success of parallel machinery for the synthesis model. We present two approaches to the training problem. The first trains a dictionary for a new ℓ^0 analysis model, which is largely motivated by the geometrical understanding of the analysis structure. The second method trains a pair of analysis and synthesis dictionaries for thresholding-based image recovery, and provides a simple and effective framework for developing image recovery processes. We find that the analysis framework thus presents a promising new field, which is well-situated to complement or compete with the synthesis approach.

List of Symbols

$\mathbf{x}$	Signal – a column vector over the real numbers
$\mathbf{y}$	Degraded signal
$\mathbf{n}$	Noise vector, typically Gaussian i.i.d.
$\boldsymbol{\Omega}$	Analysis dictionary (atoms arranged as rows)
$\mathbf{D}$	Synthesis dictionary (atoms arranged as columns)
$\boldsymbol{\Phi}, \mathbf{B}$	Base dictionaries (in the context of sparse dictionaries)
$\mathbf{A}$	Sparse atom matrix (in the context of sparse dictionaries)
$\mathbf{a}_i$	The i-th atom in a dictionary (column vector)
$\mathbf{a}_i$	The i-th column of $\mathbf{A}$ (in the context of sparse dictionaries)
$\mathbf{w}_i^T$	The i-th atom in an analysis dictionary (row vector)
$\mathbf{d}_i$	The i-th atom in a synthesis dictionary (column vector)
$\boldsymbol{\gamma}$	Sparse representation vector
$\boldsymbol{\gamma}_a$	Analysis sparse representation
$\boldsymbol{\gamma}_s$	Synthesis sparse representation
N, M	Signal lengths
M	Base dictionary size (in the context of sparse dictionaries)
L	Dictionary size / signal representation length
R	Number of training examples
d	Number of signal dimensions
k	Number of algorithm iterations
p	Atom sparsity (in the context of sparse dictionaries)
ϵ	Representation error

$R(\mathbf{x})$	Signal model (scalar penalty function)
$C(\boldsymbol{\gamma})$	Sparsity measure / representation cost function
Z	Partition function – normalizer of a distribution
$\mathbf{X}$	Matrix with signals $\mathbf{x}_i$ as its columns
$\mathbf{Y}$	Matrix with degraded signals $\mathbf{y}_i$ as its columns
$\boldsymbol{\Gamma}$	Matrix with sparse representations $\boldsymbol{\gamma}_i$ as its columns
$\boldsymbol{\gamma}_i$	The i-th column of $\boldsymbol{\Gamma}$
$\boldsymbol{\gamma}_i^T$	The i-th row of $\boldsymbol{\Gamma}$ (in context)
$\mathbf{E}$	Error or residual matrix
$\mathbf{E}_j$	Error matrix during the update of the j-th atom
$\hat{\mathbf{x}}$	Estimator of $\mathbf{x}$
$\hat{\mathbf{x}}_{\mathrm{ML}}$	Maximum-likelihood estimator of $\mathbf{x}$
$\hat{\mathbf{x}}_{\mathrm{MAP-A}}$	MAP-Analysis estimator of $\mathbf{x}$
$\hat{\mathbf{x}}_{\mathrm{MAP-S}}$	MAP-Synthesis estimator of $\mathbf{x}$
σ	Noise standard deviation
$\boldsymbol{\Sigma}$	Covariance matrix
λ	Scalar regularization parameter
$S_\lambda(\cdot)$	Scalar shrinkage operator with parameter λ
$\Psi_{\boldsymbol{\Omega}}$	MAP-Analysis defining polytope
$\Psi_{\mathbf{D}}$	MAP-Synthesis defining polytope
$\mathcal{Q}(\cdot)$	Quantization operator
$\mathbf{D}_q, \mathbf{A}_q, \boldsymbol{\Gamma}_q$	Quantized versions of $\mathbf{D}$, $\mathbf{A}$, $\boldsymbol{\Gamma}$
$\mathbf{I}$	Identity matrix
$\mathbf{I}_{N\times R}$	$N \times R$ matrix with 1's on its main diagonal and 0's elsewhere
$\mathbf{H}$	Hadamard matrix
$\overline{J}$	Complement of the set J
I_{max}	Maximum possible pixel value in an image

List of Abbreviations

AKTV	Adaptive Kernel Total Variation
BCR	Block Coordinate Relaxation
BP	Basis Pursuit
BSNR	Blurred Signal-to-Noise Ratio
CCS	Compressed Column Storage
CT	Computed Tomography
DCT	Discrete Cosine Transform
EM	Expectation-Maximization
FFT	Fast Fourier Transform
FOCUSS	Focal Underdetermined System Solver
ForWaRD	Fourier-Wavelet Regularized Deconvolution
GPCA	Generalized Principal Component Analysis
ILS-DLA	Iterative Least-Squares Dictionary Learning Algorithms
IRLS	Iterative Reweighted Least-Squares
ITD	Iteration-Tuned Dictionary
KLT	Karhunen-Loève Transform
LP	Linear Programming
LPA-ICI	Local Polynomial Approximation / Intersection of Confidence Interval
MAP	Maximum A-posteriori Probability
MDL	Minimum Description Length

ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
MOD	Method of Optimal Directions
MP	Matching Pursuit
MRF	Markov Random Field
MSE	Mean Squared Error
NP	Nondeterministic Polynomial
OMP	Orthogonal Matching Pursuit
PCA	Principal Component Analysis
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise Ratio
QP	Quadratic Programming
RMSE	Root Mean Squared Error
SMA	Sparse Matrix Approximation
SNR	Signal-to-Noise Ratio
STFT	Short-Time Fourier Transform
StOMP	Stagewise Orthogonal Matching Pursuit
SVD	Singular Value Decomposition
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Signal Models

Signal processing applications are typically concerned with only a specific subset (or family) of signals $\Omega \subset \mathbb{R}^N$ which forms the informative content. Examples of such families include natural images, facial images, fingerprints, audio recordings, video clips, medical scans, geological readings, neurological signals, and financial series, to name just a few. In practice, such a family will occupy only a small volume within the signal domain $\mathbb{R}^N$. It is this scarcity of the interesting signals which forms the core of all signal and image processing techniques, and is exploited to guide recovery, enhancement, and representation of signal data.

Signal models are a fundamental tool for facilitating this distinctiveness of the interesting signals. A signal model formulates a *mathematical description* of the family of interesting signals, which allows to distinguish them from the rest of the signal space. Indeed, due to the complexity of natural phenomena, these models are bound to remain approximate, and are subject to constant refinement. The aim of signal modeling research is to design increasingly accurate models, which faithfully capture the behavior of real signal data.

Signal models can be expressed in a variety of mathematical forms. One of the simplest and most common forms is as a penalty function

$$R(\mathbf{x}): \ \mathbb{R}^N \rightarrow \mathbb{R}^+ \ , \tag{1.1}$$

which assigns smaller penalties to signals more likely to belong to Ω. In statistical estimation theory, such a function is explained as coming from some a-priori probability distribution assumed on the signal space,

$$\mathcal{P}(\mathbf{x}) = \frac{1}{Z} \cdot e^{-R(\mathbf{x})} \ , \tag{1.2}$$

with the two related through elementary Bayesian estimation rules (see e.g., [1] which is part of this thesis). While this probabilistic interpretation is useful when employing statistical or information-theoretic signal processing methods, in this work we do not assume such an association in general. Indeed, many signal models in practical use cannot be directly related to a distribution of the form (1.2), as the resulting $\mathcal{P}(\mathbf{x})$ may not be square integrable over $\mathbb{R}^N$. Thus, we prefer to identify signal models with the definition (1.1), which allows more general constructions.

1.1.1 Applications of Signal Models

Signal models are used in a wide array of contexts. Information theory teaches us that the existence of a prior $\mathcal{P}(\mathbf{x})$ of the form (1.2) implies the ability to compress signals, with the average codeword length $E\{\ \ln(1/\mathcal{P}(\mathbf{x}))\ \}$ decreasing as $\mathcal{P}$ approaches the "true" data density $\hat{\mathcal{P}}$. Lossy compression is achieved by mapping the input signals of lower $\mathcal{P}(\mathbf{x})$ to higher probability ones. Similar techniques are sometimes used with models that do not admit to form (1.2) — such as the *sparseland* model [2] — when the model explicitly induces compact representations of its preferred signals.

Inverse problem regularization is another important use of signal models. The standard inverse problem describes the acquisition process of a measured signal $\mathbf{y} \in \mathbb{R}^M$ by transforming and distorting some origin signal $\mathbf{x} \in \mathbb{R}^N$,

$$\mathbf{y} = \mathcal{T}\mathbf{x} + \mathbf{n} \ . \tag{1.3}$$

Here, $\mathcal{T} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ is a known (not necessarily linear) transform, and $\mathbf{n} \in \mathbb{R}^M$ is the system noise. In this work we assume $\mathbf{n}$ to be white Gaussian i.i.d., though generalizations to colored Gaussian noise and other noise models often exist. The inverse problem formulation (1.3) describes a wide range of fundamental signal processing tasks, such as deconvolution, demosaicing, interpolation, super-resolution, source separation, compressive sensing, and tomography reconstruction, among others. The special choice $\mathcal{T} = \mathbf{I}$ represents the denoising problem, which is of particular interest for analysis purposes.

Recovering $\mathbf{x}$ from $\mathbf{y}$, even in the denoising case, is an impossible task without further assumptions on $\mathbf{x}$. The degradation operator $\mathcal{T}$ introduces further complexity as it is typically lossy, making its direct inversion ill-posed and highly unstable. The missing information is filled-in by the signal model, which is used to guide the solver towards solutions closer to Ω. Specifically, by penalizing undesired signals, the model $R(\mathbf{x})$ gives rise to the estimation process

$$\hat{\mathbf{x}} = \underset{\mathbf{x}'}{\text{Argmin}} \; \frac{1}{2}\|\mathbf{y} - \mathcal{T}\mathbf{x}'\|_2^2 + \lambda R(\mathbf{x}') \; , \tag{1.4}$$

where $\lambda > 0$ is a regularization parameter balancing the fidelity and regularity terms. Indeed, for a model related to a probability distribution of the form (1.2), this formulation can be interpreted as a maximum-a-posterior (MAP) estimator of $\mathbf{x}$ [1, 3, 4]. In general, though, this formulation can exist independently of such an interpretation. As can be seen, $R(\mathbf{x})$ in the above expresses all our knowledge about the set Ω, and its accuracy directly determines the success of the process.

1.1.2 Sparsity-Based Models

A central notion in the design of signal models is *sparsity.* This notion has its roots in fundamental scientific methodology (e.g., Occam's razor), and is closely related to concepts such as Minimum Description Length (MDL) [5] and Kolmogorov complexity [6]. The idea is to model signals through a *sparsifying transform* $\mathbf{x} \rightarrow \boldsymbol{\gamma}(\mathbf{x})$, where $\boldsymbol{\gamma}(\mathbf{x}) \in \mathbb{R}^L$ may have a different length than $\mathbf{x}$ (specifically, $L \geq N$). For signals in Ω, the representation $\boldsymbol{\gamma}(\mathbf{x})$ is expected to be *sparse*, in the sense that its sorted coefficients decay rapidly[1]. For signals not in Ω, the representation vectors should become denser. Indeed, the ability to define such a transform depends directly on the assumption that Ω has very small "volume" within $\mathbb{R}^N$, as clearly the set of sparse representations occupies a very small portion of $\mathbb{R}^L$. In practice, signal models rarely fully achieve these stated goals, and classify some unwanted signals as sparse, or misclassify some signals in Ω as dense; this is a main cause of artifacts and loss of information in signal recovery and compression processes.

Given the transform $\boldsymbol{\gamma}(\mathbf{x})$, the sparsity of $\boldsymbol{\gamma}$ describes the estimated likeliness of $\mathbf{x}$ belonging to Ω. Thus, a signal model $R(\mathbf{x})$ can be derived from this transform via a *sparsity measure* $C(\boldsymbol{\gamma})$, which penalizes denser representations:

$$R(\mathbf{x}) = C(\boldsymbol{\gamma}(\mathbf{x})) \ . \tag{1.5}$$

When $C(\boldsymbol{\gamma})$ forms a norm, such as the ℓ^2 norm, the model aims to decrease the overall length of $\boldsymbol{\gamma}(\mathbf{x})$, and thus penalizes mostly the large coefficients in $\boldsymbol{\gamma}$, while giving less attention to the smaller ones[2]. Alternatively, *robust* sparsity measures, which have gained substantial popularity in the past two decades, penalize more the non-vanishing small coefficients, while tolerating a limited number of large ones. Such measures are much better at capturing the *rate of decay* of a vector, and are more useful for describing modern sparsifying transforms, which are known to produce heavy-tailed coefficient distributions in natural signal data (see [8–10], and references therein). Examples of robust functions include the Huber, Cauchy, and Tukey functions, as well as the family of ℓ^p cost function with $0 \leq p \leq 1$ (see Table 1.1). The use of robust penalty functions has become increasingly prominent in many areas of statistical estimation, machine learning, and signal processing, including singular vector machines, principal component

[1] Formal measures of decay rates are established in the form of asymptotic decay bounds, but are beyond the scope of this text. An excellent reference on this topic is Mallat's book [7].

[2] Indeed the ℓ^1 norm is an exception, as it equally penalizes all magnitudes of coefficients. As such, it establishes the boundary between robust and non-robust sparsity measures.

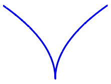
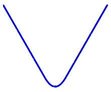
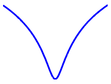
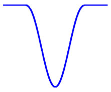
ℓ^p	$\rho(x) = \lvert x\rvert^p$	
Huber	$\rho(x) = \begin{cases} x^2/2 & \lvert x\rvert \le c \\ c(\lvert x\rvert - c/2) & \lvert x\rvert \ge c \end{cases}$	
Cauchy	$\rho(x) = \log(1 + (x/c)^2)$	
Tukey	$\rho(x) = \begin{cases} 1 - (1 - (x/c)^2)^3 & \lvert x\rvert \le c \\ 1 & \lvert x\rvert \ge c \end{cases}$	

Table 1.1: Some robust penalty functions. For all cases, $C(\boldsymbol{\gamma}) = \sum_i \rho(\gamma_i)$.

analysis, regression, clustering, and more.

1.2 Signal Modeling Using Dictionaries

With these definitions, it is clear that the careful design of the sparsifying transform is critical for the success of the model. So, how does one go about constructing a sparsifying transform? Well, a good starting point is linear operators. This approach naturally leads to the concept of a *dictionary* [11–13], which is the name given to the set of vectors, or *atoms*, describing the operator.

The dictionary is arranged as a matrix, with the atoms constituting its columns or rows. In this work we use the notations $\mathbf{D} = [\mathbf{a}_1\, \mathbf{a}_2\, \ldots\, \mathbf{a}_L] \in \mathbb{R}^{N\times L}$ and $\boldsymbol{\Omega} = [\mathbf{a}_1\, \mathbf{a}_2\, \ldots\, \mathbf{a}_L]^T \in \mathbb{R}^{L\times N}$, respectively, to distinguish the two options. When the dictionary forms a basis, it is said to be *complete*. In this case every signal has a unique representation as a linear combination of the dictionary atoms, $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$, with the linear coefficients given by $\boldsymbol{\gamma}(\mathbf{x}) = \mathbf{D}^{-1}\mathbf{x}$. The representation $\boldsymbol{\gamma}(\mathbf{x})$ can be equivalently viewed as coming from the inner products of $\mathbf{x}$ and the atoms of $\boldsymbol{\Omega} = \mathbf{D}^{-1}$, known as the *bi-orthogonal* dictionary. Some of the most well-known transforms constitute complete dictionaries, including the Fourier and DCT transforms, which sparsify uniformly smooth signals, as well as the wavelet transform, which sparsifies piecewise-smooth 1-D signals with a finite number of discontinuities [7].

1.2.1 Overcomplete Dictionaries: Analysis and Synthesis Models

Invertible dictionaries, though mathematically appealing, impose a strict limit ($L = N$) on the number of atoms in the dictionary. Consequently, complete dictionaries are limited in their ability to represent diverse natural signal behavior. Lifting this constraint, by allowing $L \geq N$, leads to more general *overcomplete* dictionary constructions, which are more descriptive than invertible dictionaries. Over the past two decades, much research has been invested in developing such dictionaries, which can increase sparsity as well as provide desirable properties such as translation and rotation invariance. Overcomplete dictionaries such as steerable pyramids [14], complex wavelets [15], curvelets [16, 17], contourlets [18, 19], surfacelets [20] and shearlets [21], as well as a wide range of trained dictionaries [22–35] are especially advantageous for multi-dimensional signal data, where invertible dictionaries lose much of their effectiveness.

In the overcomplete case, dualities of the form $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma} \;\Leftrightarrow\; \boldsymbol{\gamma} = \mathbf{\Omega}\mathbf{x}$ can no longer hold. Thus, compared to the complete case, representation with overcomplete dictionaries must be more carefully defined. Indeed, the two equivalent views of the transform in the invertible case lead to two *distinct* representation paths in the overcomplete case: the *analysis* path, where a signal $\mathbf{x}$ is represented via its inner products with the dictionary atoms,

$$\boldsymbol{\gamma}_a = \mathbf{\Omega}\mathbf{x} \; , \tag{1.6}$$

and the *synthesis* path, where the signal is represented as a linear combination of the atoms,

$$\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}_s \; . \tag{1.7}$$

In the synthesis case, further refinement is necessary due to the null space of $\mathbf{D}$, which leads to a non-unique choice of $\boldsymbol{\gamma}_s$ in (1.7). In order to obtain a well-defined representation $\boldsymbol{\gamma}_s(\mathbf{x})$ for use in (1.5), one simple approach is to utilize the Moore-Penrose pseudo-inverse dictionary $\boldsymbol{\Omega} = \mathbf{D}^+$, and select $\boldsymbol{\gamma}_s = \boldsymbol{\Omega}\mathbf{x}$. This choice essentially reduces the model to an analysis one, and is mostly used when $\mathbf{D}$ forms a *tight frame*, in which case $\boldsymbol{\Omega} = \mathbf{D}^T$ is easy to compute.

Another strategy, leading to a non-linear representation, is to choose $\boldsymbol{\gamma}_s$ as the *sparsest possible representation* based on the sparsity measure $C(\boldsymbol{\gamma})$:

$$\boldsymbol{\gamma}_s = \underset{\boldsymbol{\gamma}}{\text{Argmin}} \;\; C(\boldsymbol{\gamma}) \quad \text{Subject To} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\gamma} \; . \tag{1.8}$$

This approach assigns to each signal its highest possible likelihood according to the model, and requires more advanced machinery, developed mostly in the past fifteen or so years [7, 36]. Of specific interest is the ℓ^0 case, where $C(\boldsymbol{\gamma}) = \|\boldsymbol{\gamma}\|_0$ counts the number of non-zeros in the representation. For this case, problem (1.8) becomes the combinatorial *sparse coding* problem,

$$\boldsymbol{\gamma}_s = \underset{\boldsymbol{\gamma}}{\text{Argmin}} \;\; \|\boldsymbol{\gamma}\|_0 \quad \text{Subject To} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\gamma} \; , \tag{1.9}$$

which aims to represent $\mathbf{x}$ using the smallest number of atoms possible. This problem, known to be NP-hard in general [37], can be efficiently approximated using a wide array of algorithms, including greedy pursuits [37–41], convex relaxation [12, 42], iterative shrinkage [43–45], and others [46–48].

Another compelling choice for $C(\cdot)$ is the ℓ^1 norm $C(\boldsymbol{\gamma}) = \|\boldsymbol{\gamma}\|_1$, which provides a powerful combination of robustness and convexity. The ℓ^1 option is also touted as a stable approximation of the ℓ^0 choice [42, 49, 50]. The resulting problem is given by

$$\boldsymbol{\gamma}_s = \underset{\boldsymbol{\gamma}}{\text{Argmin}} \;\; \|\boldsymbol{\gamma}\|_1 \quad \text{Subject To} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\gamma} \; . \tag{1.10}$$

This formulation forms a convex Linear Programming (LP) problem, for which a variety of solvers are available. An interesting property of this formulation, derived from the behavior of LP problems, is that it naturally leads to a solution $\boldsymbol{\gamma}_s$ supported over a *basis* of $\mathbb{R}^n$ within $\mathbf{D}$ [12], and thus, this approach is named *Basis Pursuit* (BP).

It is worth mentioning that as an alternative to a unique representation, a stochastic approach which considers *multiple* solutions to (1.7) has also been recently proposed [51]. This approach constructs a representation $\tilde{\boldsymbol{\gamma}}_s$ which is the weighted average of several individual sparse solutions, and converges to the MMSE estimator given a noisy signal. Indeed, research in this direction is still ongoing, and is beyond the scope of this work.

1.2.2 Inverse Problem Solution Using Analysis and Synthesis Models

Dictionary-based signal models form powerful regularizers for inverse problem solution. Plugging the analysis transform in (1.4) leads to the analysis-based recovery process

$$\hat{\mathbf{x}} = \underset{\mathbf{x}'}{\mathrm{Argmin}} \, \frac{1}{2} \|\mathbf{y} - \mathcal{T}\mathbf{x}'\|_2^2 + \lambda C(\Omega\mathbf{x}') \ . \tag{1.11}$$

For the proper choice of $C(\cdot)$, this problem is convex and can be solved with standard algorithms. Of specific interest is the choice $C(\boldsymbol{\gamma}) = \|\boldsymbol{\gamma}\|_1$, as it represents the slowest growing (and hence, in a sense, the ”most robust”) convex option. For the ℓ^1 case, (1.11) becomes a Quadratic Programming (QP) problem, which is efficiently solved by interior-point methods. Algorithms based on Iterated Re-weighted Least Squares (IRLS) can also be used [52]. For specific cases, more efficient solvers exist as well [53, 54]. The analysis approach has been employed in a variety of image processing tasks, including denoising [4, 55–58], image scaling [59], tomography reconstruction [60], super-resolution [61, 62], demosaicing [62], inpainting [58], and compressed sensing [63], to name a few.

For the synthesis case, the parallel formulation leads to an optimization problem on the sparse representation $\boldsymbol{\gamma}_s$:

$$\hat{\mathbf{x}} = \mathbf{D} \cdot \underset{\boldsymbol{\gamma}_s}{\mathrm{Argmin}} \, \frac{1}{2} \|\mathbf{y} - \mathcal{T}\mathbf{D}\boldsymbol{\gamma}_s\|_2^2 + \lambda C(\boldsymbol{\gamma}_s) \ . \tag{1.12}$$

Similar to the representation problem (1.8), common choices for $C(\cdot)$ include the ℓ^0 and ℓ^1 penalty functions, among others (Table 1.1). For linear $\mathcal{T}$, the ℓ^0 case is typically solved using suitable variants of the sparse coding algorithms mentioned above. In the ℓ^1 case, problem (1.12) becomes a QP problem with efficient solvers. Alternatively, more specialized algorithms include FOCUSS [46], feature-sign search [30], and iterative thresholding methods [43–45]. The synthesis formulation has been successfully applied in a wide range of inverse problems, including image denoising [2, 31, 33, 64, 65], video denoising [31, 66], demosaicing [64, 65], inpainting [64, 67], image upscaling [68, 69], source separation [70–72], music transcription [72, 73], and tomography reconstruction [74, 75], to name just a few.

A hybrid approach, known as *thresholding* (or *shrinking*), is also in frequent use. This method, originally introduced in [76] for wavelet dictionaries, is typically employed for denoising, and arises as the *analytic* solution to (1.11) for an *orthogonal* dictionary $\mathbf{\Omega}$ and a separable penalty function $C(\boldsymbol{\gamma}) = \sum_i \rho(\boldsymbol{\gamma}_i)$. For this case, it can be shown [44] that the solution to

$$\hat{\mathbf{x}} = \operatorname*{Argmin}_{\mathbf{x}'} \frac{1}{2}\|\mathbf{y} - \mathbf{x}'\|_2^2 + \lambda C(\mathbf{\Omega}\mathbf{x}') \tag{1.13}$$

is given by

$$\hat{\mathbf{x}} = \mathbf{\Omega}^{-1} S_\lambda(\mathbf{\Omega}\mathbf{y}) \ , \tag{1.14}$$

where S_λ is an element-wise attenuation (or shrinking) of the coefficients in $\mathbf{\Omega}\mathbf{y}$, dependent on the magnitude of λ. As overcomplete and non-orthogonal dictionaries evolved, this method was burrowed for these cases as well, replacing $\mathbf{\Omega}^{-1}$ in (1.14) with $\mathbf{\Omega}^{+}$ (see e.g., [77–80]). In the overcomplete case, we can view this as a sparse approximation process over the dictionary $\mathbf{D} = \mathbf{\Omega}^{+}$, computed from the analysis coefficients $\mathbf{\Omega}\mathbf{y}$. In this sense, the process forms a type of hybrid analysis-synthesis approach. Formally, this process was later justified as constituting the first iteration of an *iterative shrinkage* method, which is known to solve the synthesis denoising problem [44, 81].

1.3 Dictionary Choice

The discussion so far has assumed that the dictionaries of the analysis and synthesis models are known. In practice, these dictionaries form the core component of these models, and embody all our knowledge of the signal domain Ω. Thus, choosing the dictionary carefully is an important and involving task, in which substantial research has been invested.

The scientific community has developed two main routes for designing dictionaries for signal modeling. The first is the *analytic* route, which derives the dictionary from a set of mathematical assumptions made on the signal family. This approach approximates the signals of interest as coming from simpler classes of mathematical functions, and designs efficient (and typically provably optimal) dictionaries for these simplified classes. The second route is the *learning* route, which infers the dictionary from signal realizations via machine-learning techniques. This approach replaces prior assumptions on the signal behavior with a training process which constructs the dictionary based on the observed signal properties. In [13], which is part of this thesis, we discuss these two options in detail, and highlight the advantages of each.

As outlined in [13], some of the most important elements of effective dictionary design include *localization*, *geometric invariance*, and *adaptivity*. Modern dictionaries typically provide localization in both the analytic and training routes. However, geometric invariance is usually better supported by analytic structures, whereas adaptivity is mostly found in training methods. Additional advantages of analytic dictionaries include algorithmic efficiency as well as compact representation. The main advantage of trained dictionaries is their ability to provide a much higher degree of specificity to the particular signal properties, allowing them to produce better results in many practical applications.

Most recently, attempts to combine the two approaches have led to the development of several *parametric* dictionary structures. These dictionary structures are controlled by a predefined set of tunable parameters, and by balancing structure and parameter count, can achieve a spectrum of complexities, invariances, and adaptivity levels. Examples of parametric dictionaries include the union-of-orthobases dictionary [27], the Generalized PCA [82], the semi-multiscale dictionary [31], the translation-invariant ILS-DLA [29], the signature dictionary [32], the hybrid Wavelet/K-SVD dictionary [35], and the sparse dictionary [33] which is part of this thesis.

1.4 Thesis Overview and Main Contributions

This thesis studies several theoretical and practical aspects of dictionary-based signal modeling. The theoretical aspects include dictionary design methodology and the analysis-synthesis relationship. The practical aspects include the sparse dictionary structure as well as algorithms for analysis and thresholding dictionary training.

1.4.1 Thesis Outline

The thesis consists of four papers and two chapters. We begin with [1], presenting the analysis and synthesis signal models and exploring their relationship in detail. We continue with [13], by considering the core component of these two models — the dictionary, and highlighting the main paradigms and concepts guiding the design of effective dictionaries. We describe the two fundamental paths — analytic and learning — used to design dictionaries, and discuss the emerging trend of fusing the two paths through parametric structures. In [33] we present a specific flexible parametric dictionary structure which we name the *sparse dictionary* structure, and discuss its benefits. A particular application of the proposed structure is discussed in [83], where a generic image compression scheme is developed and implemented, based entirely on adaptive dictionaries.

The two additional chapters in this thesis document our recent work on analysis dictionary learning. In Chapter 6 we introduce the ℓ^0 analysis model, inspired by the work in [1], and describe a training algorithm for this model which we name *Analysis K-SVD*. In Chapter 7 we focus on the thresholding framework (1.14), which we generalize to arbitrary recovery tasks, and develop a "model-less" dictionary training algorithm in which examples of origin and degraded signals replace the need for explicit knowledge of the degradation process.

In the following we review the main contributions of this thesis in more detail.

1.4.2 Analysis and Synthesis Relationship

The analysis and synthesis models described in Section 1.2.1 share a common conceptual foundation of sparsifying signal coefficients over a dictionary. The two approaches become equivalent in the invertible case, and have similar formulations in the overcomplete case. These observations have led to the conjecture that the two are closely related, see e.g. [84].

Mathematically, the gap between the two can be formulated as follows. Beginning with the analysis formulation (1.11), we define

$$\boldsymbol{\gamma}_a = \mathbf{\Omega}\mathbf{x}' \ ,$$

which implies, under a full-rank assumption on $\mathbf{\Omega}$,

$$\mathbf{x}' = \mathbf{\Omega}^+\boldsymbol{\gamma}_a \ .$$

Substituting these in (1.11) results in replacing the optimization over $\mathbf{x}'$ with an optimization over $\boldsymbol{\gamma}_a$. However, in defining $\boldsymbol{\gamma}_a = \mathbf{\Omega}\mathbf{x}'$ we constrain the optimization to only consider representations $\boldsymbol{\gamma}_a$ spanned by the columns of $\mathbf{\Omega}$. Thus, we introduce the constraint $\boldsymbol{\gamma}_a = \mathbf{\Omega}\mathbf{x}' = \mathbf{\Omega}\mathbf{\Omega}^+\boldsymbol{\gamma}_a$ in the optimization, leading to the following equivalent form of the analysis estimator:

$$\hat{\mathbf{x}} = \mathbf{\Omega}^+ \cdot \underset{\boldsymbol{\gamma}_a}{\mathrm{Argmin}} \ \frac{1}{2}\|\mathbf{y} - \mathcal{T}\mathbf{\Omega}^+\boldsymbol{\gamma}_a\|_2^2 + \lambda C(\boldsymbol{\gamma}_a) \quad \text{Subject To} \quad \mathbf{\Omega}\mathbf{\Omega}^+\boldsymbol{\gamma}_a = \boldsymbol{\gamma}_a \ . \tag{1.15}$$

As can be seen, this problem strongly resembles the synthesis structure (1.12) for the choice $\mathbf{D} = \mathbf{\Omega}^+$, except for the added constraint on $\boldsymbol{\gamma}_a$.

In [1] we perform a thorough investigation of the two paradigms, focusing on the relationship between the two. Considering their algebraic similarity, it comes as somewhat of a surprise that in reality, a large gap exists between the two models in the overcomplete case. As it turns out, the innocently-looking constraint in (1.15) can have a substantial effect on the result of the optimization. A simple demonstration is provided in Fig. 1.1, which shows how a significant gap may evolve even in a simple 2-D case.

Our work takes a geometric approach, analyzing the two models in terms of their iso-surfaces in signal space. The result of this view is the characterization of a large (exponential) number of signals on which the two formulations are bound to differ, leading to the inevitable conclusion that an equivalence between the two cannot exist. This result is general in the sense that it does not assume a specific relation (such as the pseudo-inverse) between the analysis and synthesis dictionaries, nor does it assume they have the same number of atoms (though it is assumed they have a *similar* number of atoms, e.g., up to a constant factor).

In practice, the significance of these results is two-fold. First, the realization that the two models are distinct spawns renewed interest in the analysis model, which has received less attention in recent years in favor of the synthesis model. As an example, we show in [1] a simple denoising case where the analysis option outperforms the synthesis one. Second, the improved understanding of the analysis model paves the way to further research on this model, such as analysis dictionary training, which is discussed later in this thesis.

On the other hand, the results of [1] should also be taken in perspective. Indeed, it remains possible that for specific dictionaries and specific signal families, tighter relations could be derived. Our analysis is very much a *worst-case* one, focusing on the signals for which the two approaches differ the most; other signals may

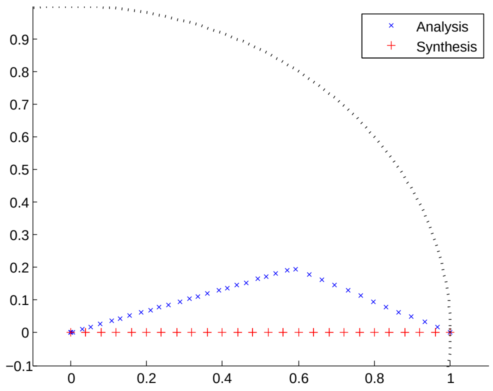


Figure 1.1: A simple 2-D example where analysis and synthesis depart. Here, $\mathbf{D} = \left(\begin{smallmatrix} 1 & 0 & \sqrt{2}^{-1} \\ 0 & 1 & \sqrt{2}^{-1} \end{smallmatrix}\right)$, $\mathbf{\Omega} = \mathbf{D}^{+}$, $\mathcal{T} = \mathbf{I}$, and $\mathbf{y} = (1,0)^T$. The penalty function is the ℓ^1 norm $C(\cdot) = \|\cdot\|_1$. The plot shows the analysis and synthesis estimates (1.11) and (1.12) for varying λ between 0 and 2, advancing from right to left. The dotted line is the unit sphere, shown for reference. As can be seen, the two estimates depart from $\mathbf{y}$ at quite different directions. The gap quickly increases, reaching a maximal difference $\|\mathbf{x}_a - \mathbf{x}_s\|_2$ of over 30% the energy of $\mathbf{x}_s$ at $\lambda \approx 0.65$.

exhibit a smaller gap. However, the fundamental conclusion remains. Specifically, our results dictate that the sets of signals for which the two approaches differ exist for *any* pair of analysis and synthesis dictionaries, and not just for specific cases.

1.4.3 Sparse Dictionaries

As discussed in Section 1.3, parametric dictionaries are gaining interest due to their ability to benefit from both the analytic and machine-learning design paradigms. The *sparse dictionary* structure presented in [33] is a particular parametric dictionary aimed at bridging this gap between the analytic and learning routes. The sparse dictionary structure suggests representing the dictionary as a composition of an *analytic* base dictionary and a sparse *trained* dictionary. In synthesis notation, this dictionary takes the form

$$\mathbf{D} = \mathbf{\Phi}\mathbf{A} \ , \tag{1.16}$$

where $\mathbf{\Phi}$ is a fixed analytic dictionary, and $\mathbf{A}$ is an adaptable sparse matrix. We note that the dictionary structure may be employed in both analysis and synthesis scenarios, though we focus on the synthesis case in [33].

The sparse dictionary is shown to achieve similar or superior estimation results to an ordinary (non-structured) trained dictionary, while providing substantial gains in complexity and generalization due to the imposed structure. In this, the dictionary structure exhibits many of the benefits of both design routes. Also, by modifying the number of non-zeros in **A**, the sparse dictionary can achieve an essentially continuous transition from analytic dictionaries (very sparse **A**) to fully unconstrained dictionaries (dense **A**). In this sense, the sparse structure is truly a ”bridge” between the two approaches.

An additional advantage of sparse dictionaries is their compact representation compared to non-structured dictionaries. This compactness makes their use feasible for compression tasks. In [83] we present an image-adaptive compression scheme which encodes an input image over a *specifically trained* dictionary sent along with the compressed stream. Such a scheme has so far been impractical due to the overhead of transmitting the dictionary. In this work we show that the scheme based on sparse dictionaries can convincingly outperform JPEG compression and approach JPEG2000 performance in some cases. Though our results do not reach state-of-the-art, this preliminary work clearly positions image-adaptive dictionaries as a plausible option for generic image compression. Directions for future improvement, including multi-scale extensions and hybrid trained and analytic dictionaries, are mentioned in [83].

Finally, the sparse dictionary structure provides a convenient framework for training parametric dictionaries with additional desired properties, by imposing specific and meaningful structures on the matrix **A**. This option is not explored in this work, but has been studied by others [35, 85].

1.4.4 Analysis and Thresholding Dictionary Learning

Following substantial achievements in synthesis dictionary training, researchers are recently gaining interest in the question of dictionary training for analysis models [86, 87]. Indeed, the formalization of the gap between the analysis and synthesis frameworks in [1] provides significant incentive for this quest, as it opens the door to an array of new opportunities with the analysis formulation. One of the first and most influential attempts to train a (non-orthogonal) dictionary for the analysis model was the pioneering work of Black and Roth [88], who trained a Markov Random Field (MRF) image prior of the form:

$$\mathcal{P}(\mathbf{x}) \sim \exp\left\{ -\sum_k \lambda^T C(\boldsymbol{\Omega}\mathbf{x}_k) \right\} \ .$$

In this expression, the sum is over all overlapping blocks $\mathbf{x}_k$ in the image $\mathbf{x}$, and $C(\boldsymbol{\gamma}) = \sum_i \rho(\boldsymbol{\gamma}_i)$ is a robust cost function with $\rho(\alpha) = \ln\left(1 + \frac{\alpha^2}{2}\right)$. The proposed training algorithm minimizes the Kullback-Leibler divergence of the learned and data distributions via a specialized gradient descent method, and shows promising results in image denoising and inpainting.

In this thesis we adopt a different approach to the analysis training problem, based on ℓ^0 sparsity. This sparsity measure allows the development of more efficient training algorithms, and leads to interesting relations with the synthesis formulation. Specifically, our methods trains *overcomplete* dictionaries, compared to the undercomplete dictionaries trained by [88].

We present two approaches to the training task. The first focuses on the recently proposed ℓ^0 *analysis model*, and is presented in Chapter 6. This new model is interested in signals which nullify a large number of coefficients in $\boldsymbol{\Omega}\mathbf{x}$, and is motivated by the observation in [1] that the favorable signals of the ℓ^1 analysis prior are orthogonal to many rows in the analyzing dictionary. In this chapter we introduce the ℓ^0 analysis model, discuss signal coding under the new model, and present the *Analysis K-SVD* algorithm for dictionary training, which is named after the original K-SVD due to the resemblance between the two. Simulations show the ability of our algorithm to successfully recover an underlying model given sparse examples, as well as discover meaningful structures in natural image data.

Next, in Chapter 7 we study a generalization of the thresholding process (1.14), which we name *analysis-synthesis thresholding* due to its use of two separate dictionaries for the analysis and synthesis stages of the estimation. The resulting process can be applied in a variety of recovery tasks, and is given by:

$$\hat{\mathbf{x}} = \mathbf{D}S_\lambda(\mathbf{\Omega y}) \ .$$

In this work we consider specifically the ℓ^0 hard thresholding case, where we can exploit the exact sparsity to develop an efficient training algorithm in the spirit of the K-SVD and Analysis K-SVD. We show that our algorithm is able to efficiently optimize the resulting target function, and present favorable recovery results for small-kernel image deblurring. Compared to traditional synthesis-based recovery methods, a notable advantage of the thresholding recovery process is its significantly lower complexity due to the low cost of the thresholding operator. Another advantage of our recovery process is its parameterless nature, as all parameters, including threshold values, are tuned during the training process. A unique property of the proposed framework is its example-based approach to the degradation modeling, which requires no explicit specification of the degradation process, and instead deduces its properties from the training data itself (which is assumed to undergo a uniform degradation). We conclude by outlining some possible directions for future research, including extensions to more complex degradation models, MRF recovery processes, and others.

Chapter 2

Analysis versus Synthesis in Signal Priors

Published in IOP Inverse Problems, 23(3):947–968, 2007.

Coauthored with Michael Elad and Peyman Milanfar.

Abstract

The concept of prior probability for signals plays a key role in the successful solution of many inverse problems. Much of the literature on this topic can be divided between analysis-based and synthesis-based priors. Analysis-based priors assign probability to a signal through various forward measurements of it, while synthesis-based priors seek a reconstruction of the signal as a combination of atom signals. The algebraic similarity between the two suggests that the two could be strongly related; however, in the absence of a detailed study, contradicting approaches have emerged. While the computationally-intensive synthesis approach is receiving ever-increasing attention and is notably preferred, other works hypothesize that the two might actually be much closer, going as far as to suggest that one can approximate the other. In this paper we describe the two prior classes in detail, focusing on the distinction between them, and our results put to

question, in fact, both these assumptions. We show that although in the simpler complete and under-complete formulations the two approaches are equivalent, in their overcomplete formulation they depart. Focusing on the ℓ^1 case, we present a novel approach for comparing the two types of priors based on high-dimensional polytopal geometry. We arrive at a series of theoretical and numerical results establishing the existence of an unbridgeable gap between the two.

2.1 Introduction

The general inverse problem seeks the recovery of an unknown signal $\mathbf{x} \in \mathbb{R}^N$ (a vector of dimension N over the real numbers) based on indirect measurements of it given in the vector $\mathbf{y} \in \mathbb{R}^M$. A typical model for describing the relation between $\mathbf{x}$ and $\mathbf{y}$ is

$$\mathbf{y} = \mathbf{T}\{\mathbf{x}\} + \mathbf{v} \ , \tag{2.1}$$

where $\mathbf{T} : \ \mathbb{R}^N \to \mathbb{R}^M$ is a (possibly non-linear) known operator, and $\mathbf{v} \in \mathbb{R}^M$ is a zero-mean white Gaussian additive noise vector (other models for the noise could also be considered, but here we restrict the discussion to the assumptions made above for simplicity). Many important problems in signal and image processing are represented using this structure: these include denoising, interpolation, scaling, super-resolution, inverse Radon transform, reconstruction from projections in general, and motion estimation, to name just few. In all these problems, the general task is an inversion of the operator $\mathbf{T}$.

Inverting the above process can be done in many different ways. When lacking any a-priori knowledge about the unknown, Maximum Likelihood (ML) estimation suggests finding the $\mathbf{x}$ that leads to the most probable set of measurements $\mathbf{y}$. We get a solution of the form

$$\begin{aligned}\hat{\mathbf{x}}_{\mathrm{ML}} &= \underset{\mathbf{x}}{\mathrm{Argmax}}\ \ \mathrm{Prob}\{\mathbf{y}\,|\,\mathbf{x}\} \\ &= \underset{\mathbf{x}}{\mathrm{Argmax}}\ \ \exp\left\{-\frac{1}{2\sigma_{\mathrm{v}}^2}\|\mathbf{y}-\mathbf{T}\{\mathbf{x}\}\|^2\right\} \\ &= \underset{\mathbf{x}}{\mathrm{Argmin}}\ \ \|\mathbf{y}-\mathbf{T}\{\mathbf{x}\}\|_2^2\ .\end{aligned}$$

As an example, if $\mathbf{T}\{\mathbf{x}\} = \mathbf{H}\mathbf{x}$, where $\mathbf{H}$ is a known degradation operator represented as a full rank matrix with more columns than rows, the ML solution amounts to the pseudo-inverse of the degrading operator, thus $\hat{\mathbf{x}}_{\mathrm{ML}} = \mathbf{H}^{+}\mathbf{y}$. For the denoising problem ($\mathbf{H} = \mathbf{I}$), ML suggests the solution $\hat{\mathbf{x}}_{\mathrm{ML}} = \mathbf{y}$, which clearly demonstrates the weakness of ML.

Generally speaking, the literature today offers through the Bayesian approach a stabilized solution to the inverse problem posed above. We concentrate on the use of the Maximum-A-posteriori Probability (MAP) estimator, which regularizes the estimation process using an assumed *prior distribution* on the signal space. Indeed, such signal priors are implicitly used in many other signal processing applications such as compression, signal decomposition, recognition, and more.

2.1.1 MAP-Analysis Approach

When studying the variety of published work in the field, two main prior types emerge. The first utilizes an analysis-based approach, deriving the probability of a signal from a set of forward transforms applied to it. Such priors form the backbone of many classic as well as more recent algorithms, and most commonly appear as regularizing elements in optimization problems or PDE methods. In this paper, we focus on a robust Gibbs-like distribution, of the form

$$\mathrm{Prob}\{\mathbf{x}\} = \mathrm{Const}\cdot\exp\{-\alpha\cdot\|\boldsymbol{\Omega}\mathbf{x}\|_p^p\}\ ,$$

where $\boldsymbol{\Omega} \in M^{[L\times N]}$ is some pre-specified matrix, and $\|\cdot\|_p^p$ is the ℓ^p norm. The term $\|\boldsymbol{\Omega}\mathbf{x}\|_p^p$ is an energy functional that is supposed to be low for highly probable signals, and higher as the signal is less probable. We refer to $\mathbf{\Omega}$ as the *analyzing operator*. Merged with the Gaussianity assumption on the additive noise, this poses the MAP recovery process as the minimization problem

$$\begin{aligned} \hat{\mathbf{x}}_{\mathrm{MAP-A}} &= \underset{\mathbf{x}}{\mathrm{Argmax}}\;\; \mathrm{Prob}\{\mathbf{x}\,|\,\mathbf{y}\} \\ &= \underset{\mathbf{x}}{\mathrm{Argmax}}\;\; \mathrm{P}\{\mathbf{y}\,|\,\mathbf{x}\}\;\mathrm{P}\{\mathbf{x}\}\,/\,\mathrm{P}\{\mathbf{y}\} \\ &= \underset{\mathbf{x}}{\mathrm{Argmin}}\;\; -\log \mathrm{P}\{\mathbf{y}\,|\,\mathbf{x}\} \;-\; \log \mathrm{P}\{\mathbf{x}\} \\ &= \underset{\mathbf{x}}{\mathrm{Argmin}}\;\; \|\mathbf{y}-\mathbf{T}\{\mathbf{x}\}\|_2^2 + \lambda\cdot\|\mathbf{\Omega x}\|_p^p \end{aligned} \tag{2.2}$$

where $\lambda = 2\alpha\sigma_{\mathrm{v}}^2$. When robust norms are used ($p < 2$ or some robust M-function [57]), an iterative algorithm is typically employed for the minimization of (2.2). Preference is generally given to $p \geq 1$ so that the overall penalty function is convex, thus guaranteeing a unique solution. We name this method the *MAP-Analysis* approach since the prior is based on a sequence of linear filters applied to the signal, essentially analyzing its behaviour.

The analysis structure is quite common in inverse problems in signal processing, image processing, and computer vision. In a typical image processing application where an image is an unknown, $\mathbf{\Omega}$ is chosen as some sort of derivative operator, promoting spatial smoothness in the image $\mathbf{x}$. As to the choice of p, choosing the ℓ^2 norm is known to lead to a simplified analytic treatment, but also known to give non-robust results (i.e. smoothing of discontinuities). Thus, recent contributions concentrate on robustness by using ℓ^p norms with $p < 2$, leading to non-linear filtering algorithms [55–57, 59, 60, 62, 89–91].

2.1.2 MAP-Synthesis Approach

The second type of prior arises from employing a synthesis-based approach. Synthesis-based methods are a more recent contribution, and stem in a large part from the Basis Pursuit method pioneered by Chen, Donoho & Saunders [12].

Suppose that a signal $\mathbf{x} \in \mathbb{R}^N$ is to be represented as a linear combination of

”building-block” atoms taken as the columns of a full-rank matrix $\mathbf{D} \in M^{[N\times L]}$, with $L \geq N$ (notice the different size compared to $\mathbf{\Omega}$). This matrix has N rows and L columns, and we refer to the columns of $\mathbf{D}$ as the *atom* signals. This leads to the linear under-determined equation set $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$, where $\boldsymbol{\gamma} \in \mathbb{R}^L$ is overcomplete. We assume for the idealized signal $\mathbf{x}$ that its representation $\boldsymbol{\gamma}$ is *sparse*, implying that only a few atoms are involved in its construction. Assuming $\mathbf{y}$ is a noisy version of this signal, then the following is the *MAP-Synthesis* option for the recovery of $\mathbf{x}$:

$$\hat{\mathbf{x}}_{\mathrm{MAP-S}} = \mathbf{D} \cdot \underset{\boldsymbol{\gamma}}{\mathrm{Argmin}} \;\; \|\mathbf{y} - \mathbf{T}\{\mathbf{D}\boldsymbol{\gamma}\}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p \; . \tag{2.3}$$

In this expression, the ℓ^p-norm with $p < 2$ seeks the sparsest representation vector $\boldsymbol{\gamma}$ that explains $\mathbf{y}$ in terms of the dictionary columns. Note that if the solution of the optimization problem is denoted as $\hat{\boldsymbol{\gamma}}$, the estimated output signal is given by $\hat{\mathbf{x}}_{\mathrm{MAP-S}} = \mathbf{D}\hat{\boldsymbol{\gamma}}$.

Synthesis-based methods have evolved rapidly over the past decade. Significant progress has been seen in the development of modern dictionaries for sparse image representation, such as the Ridgelet, Curvelet and Contourlet dictionaries [16, 18, 92]; training from example sets has also been successfully explored [28]. Parallel advancements, many of them theoretical in nature, have been achieved in the areas of sparse coding (i.e. finding sparse representations) and sparsity-based signal recovery [42, 93].

Through the MAP framework, the synthesis approach may be generalized to incomplete dictionaries. We let $\mathbf{\Gamma}_{\mathbf{x}} = \{\boldsymbol{\gamma} \mid \mathbf{x} = \mathbf{D}\boldsymbol{\gamma}\}$ denote the set of representations of $\mathbf{x}$ in $\mathbf{D}$, where $\mathbf{\Gamma}_{\mathbf{x}}$ may be infinite, empty, or a singleton. The a-priori probability assumed for $\mathbf{x}$ depends on its sparsest representation in $\mathbf{D}$. In this setting, signals not spanned by the columns of $\mathbf{D}$ are assigned a-priori probability 0.

The MAP-Synthesis prior is therefore given as a Gibbs distribution on the optimal representations:

$$\text{Prob}\{\mathbf{x}\} = \begin{cases} \text{Const} \cdot \exp\{-\alpha \cdot \|\hat{\boldsymbol{\gamma}}(\mathbf{x})\|_p^p\} & \text{if } \boldsymbol{\Gamma}_{\mathbf{x}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

where

$$\hat{\boldsymbol{\gamma}}(\mathbf{x}) = \underset{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}_{\mathbf{x}}}{\text{Arg}\min} \ \|\boldsymbol{\gamma}\|_p^p \ .$$

This prior, when plugged into the MAP formulation, leads precisely to the process described in (2.3). From a practical point of view, an iterative algorithm is required for the solution of (2.3), and there are many methods to do so effectively. For $p \geq 1$, we are guaranteed to have a unique solution.

2.1.3 Analysis versus Synthesis

Comparing the two recovery processes in (2.2) and (2.3), we see that the two describe very similar structures. The heuristic behind each remains sparsifying the representation of the signal — be this its forward projection on the basis elements, or its reconstruction as their linear combination.

How do the two methods compare? The conjecture that natural images can be effectively described as sparse combinations of atomic elements has found empirical support [94] which the analysis-based approach lacks. The concept also has clear advantages in applications such as image compression, feature extraction, content-based image retrieval and others. Furthermore, as opposed to the analysis approach, the synthesis approach has a constructive form providing an explicit description of the signals it represents, and as such, is more intuitive to interpret and design.

A different concern about the analysis approach is its capacity to benefit from the increased redundancy. As this approach requires a signal to simultaneously agree with *all* the rows of $\boldsymbol{\Omega}$, this might become impossible with a highly redundant operator, rendering the prior useless. The synthesis approach, in contrast, seems to benefit from higher redundancy, as this enriches the prior, enabling it to describe more complex types of signals.

On the other hand, the compactness promoted by the synthesis approach might also come as a weakness. In such a framework where only a small number of atoms are used to represent each signal, the significance of every atom grows enormously; any wrong choice — in a denoising scenario for instance — could potentially lead to a ”domino effect” where additional erroneous atoms are selected as compensation, deviating further from the desired description. In the analysis formulation, however, all atoms take an equal part in describing the signal, thus minimizing the dependence on each individual one, and stabilizing the recovery process.

Analysis-based methods, specifically in their robust form ($p < 2$), are a very common structure in image processing and computer vision applications. In a large part, this is because MAP-Analysis leads to a simple optimization problem, which (in the overcomplete case) is considerably easier to solve — due to the smaller dimension of the unknown — compared to a similar-sized MAP-Synthesis form. At the same time, however, a growing number of works are employing the synthesis approach for inverse problem regularization. The synthesis-based approach is attractive due to its intuitive and versatile structure, and informally, is widely considered to provide superior results. This recent trend is strengthened by a wealth of theoretical and practical advancements, making the synthesis approach both more appealing and computationally tractable [42, 93, 95].

Nonetheless, MAP-Synthesis remains a prohibitive option in many cases. This has led several works to seek alternative approaches over direct minimization. One option which has been proposed is the use of an analysis-based method to approximate the synthesis-based one, as is done in [84] where the analysis operator is taken as the pseudo-inverse of the synthesis dictionary. This approach has only been partially justified, however, leaving the question of its generality much unattended.

2.1.4 This Paper's Contribution

As can be seen from the discussion, the literature to-date is highly ambivalent in respect to the two regularization approaches. The extensive research of the synthesis-based methods implicitly suggests MAP-Synthesis is superior to MAP-Analysis — especially considering the huge gap in complexity between the two structures. At the same time, other works, building on the algebraic similarity presented in the next section, hypothesize that the two are actually much closer, in fact close enough to approximate one another [84].

In light of these developments, it is our goal in this paper to clarify the *distinction* between the two approaches, and shed some light on the conceptual and technical gaps between them. We show that indeed for specific cases the two approaches are equivalent, utilizing a pseudo-inverse relation between the analysis operator and synthesis dictionary. Such is the case for the square and under-complete formulations, as well as for the ℓ^2 (i.e. $p = 2$) choice. However, as we go to the general overcomplete formulation ($L > N$), we find that the equivalence between the two MAP options breaks. Concentrating on the $p = 1$ case, often favoured due to its convexity and robustness, we provide theoretical as well as numerical results indicating that the two methods are fundamentally distinct. Our results break, in fact, *both* of the above common assumptions: first in establishing the gap between the two approaches, and second by presenting simulations where the analysis approach actually supersedes its synthesis counterpart.

This paper is organized as follows. Section 2.2 describes the square and under-determined cases, where the two methods exhibit almost complete equivalence. In Section 2.3 we turn to discuss the overcomplete case, focusing on the ℓ^1 choice. Taking a geometrical viewpoint, we construct the theoretical model describing the gap between the two methods, and discuss some consequences of this model.

Simulation results are provided in Section 2.4, and Section 2.5 concludes with a summary of the claims made in the paper.

2.2 The Square and Under-Determined Cases

We begin by showing that in the (under-)determined case (i.e., $L \leq N$), the two methods are practically equivalent.

Theorem 2.1. Square Non-Singular Case – Complete Equivalence. *MAP-Analysis and MAP-Synthesis are equivalent if MAP-Analysis utilizes a square and non-singular analyzing operator $\mathbf{\Omega}$. The equivalent MAP-Synthesis method is obtained for the dictionary $\boldsymbol{D} = \mathbf{\Omega}^{-1}$.*

Proof. We start with the MAP-Analysis approach as posed in equation (2.2). Since $\mathbf{\Omega}$ is square and non-singular, defining $\mathbf{\Omega}\mathbf{x} = \boldsymbol{\gamma}$ leads to $\mathbf{x} = \mathbf{\Omega}^{-1}\boldsymbol{\gamma}$. Putting this into (2.2), we get an alternative optimization problem with $\boldsymbol{\gamma}$ replacing $\mathbf{x}$ as unknown,

$$\hat{\mathbf{x}} = \mathbf{\Omega}^{-1} \cdot \underset{\boldsymbol{\gamma}}{\text{Argmin}} \;\; \|\mathbf{y} - \mathbf{T}\{\mathbf{\Omega}^{-1}\boldsymbol{\gamma}\}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p \, ,$$

and the equivalence to the MAP-Synthesis method in (2.3) is evident. Likewise, starting from the MAP-Synthesis formulation and using the same argument, we can obtain a MAP-Analysis one — and thus the two methods are equivalent. □

The generalization of Theorem 2.1 for the $L \leq N$ case requires more care, and is only true for the denoising ($\mathbf{T} = \mathbf{I}$) case. Before stating the theorem, we point out that complete equivalence cannot be guaranteed in this case due to the property of MAP-Synthesis to only produce results in the column-span of $\mathbf{D}$, while MAP-Analysis poses no such restriction. Nevertheless, the following theorem represents both conceptually and computationally a complete equivalence between the two, as knowing the solution to either one immediately fixes the solution to the other. We arrive at the following result, whose proof is postponed to the appendix:

Theorem 2.2. Under-Complete Denoising Case – Near-Equivalence. *MAP-Analysis denoising with a full-rank analyzing operator $\boldsymbol{\Omega} \in M^{[L \times N]}$ $(L \leq N)$ is nearly-equivalent to MAP-Synthesis with the dictionary $\boldsymbol{D} = \boldsymbol{\Omega}^{+}$. This is expressed by the relation $\hat{\boldsymbol{x}}_{\mathrm{MAP-A}} = \hat{\boldsymbol{x}}_{\mathrm{MAP-S}} + \boldsymbol{y}^{\boldsymbol{D}_{\perp}}$, with $\boldsymbol{y}^{\boldsymbol{D}_{\perp}}$ representing the component of the input orthogonal to the columns of $\boldsymbol{D}$.*

(Proof in 2.A.)

We also see that when the input is in the column-span of $\mathbf{D}$ (as in the square non-singular case), we obtain $\hat{\mathbf{x}}_{\mathrm{MAP-A}} = \hat{\mathbf{x}}_{\mathrm{MAP-S}}$.

2.3 The Over-Determined Case

We have seen that the two methods are practically equivalent for the $L \leq N$ case. Our main interest however is in the overcomplete $(L > N)$ case, advocated strongly by the Basis Pursuit approach. A natural starting point for analyzing the overcomplete case is the pseudo-inverse relation, which, as we have just seen, successfully achieves equivalence in the (under-)complete case. We assume hereon that $\boldsymbol{\Omega}$ has full column rank, and hence $\boldsymbol{\Omega}^{+}\boldsymbol{\Omega} = I$. Beginning with the MAP-Analysis formulation in (2.2), we let $\boldsymbol{\Omega}\mathbf{x} = \boldsymbol{\gamma}$. Since $\boldsymbol{\Omega}^{+}\boldsymbol{\Omega} = I$, recovering $\mathbf{x}$ from $\boldsymbol{\gamma}$ is done by $\mathbf{x} = \boldsymbol{\Omega}^{+}\boldsymbol{\gamma}$. However, in replacing the unknown from $\mathbf{x}$ to $\boldsymbol{\gamma}$ we must add the constraint that $\boldsymbol{\gamma}$ is spanned by the columns of $\boldsymbol{\Omega}$, due to its definition (this can be represented by the constraint $\boldsymbol{\Omega}\boldsymbol{\Omega}^{+}\boldsymbol{\gamma} = \boldsymbol{\gamma}$). Thus we obtain the following equivalent MAP-Analysis form:

$$\hat{\mathbf{x}}_{\mathrm{MAP-A}} = \boldsymbol{\Omega}^{+} \cdot \underset{\boldsymbol{\gamma}:\ \boldsymbol{\Omega}\boldsymbol{\Omega}^{+}\boldsymbol{\gamma}=\boldsymbol{\gamma}}{\mathrm{Argmin}} \; \|\mathbf{y} - \mathbf{T}\{\boldsymbol{\Omega}^{+}\boldsymbol{\gamma}\}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p \,. \tag{2.5}$$

Comparing this to (2.3), we see that if the MAP-Synthesis solution (with $\mathbf{D} = \boldsymbol{\Omega}^{+}$) satisfies the constraint $\boldsymbol{\Omega}\boldsymbol{\Omega}^{+}\boldsymbol{\gamma} = \boldsymbol{\gamma}$, then omitting it in (2.5) has no effect, and both approaches arrive at the same solution. However, in the general case this constraint is not satisfied, and thus the two methods lead to different results.

An interesting observation is that while the representation solutions could differ vastly, the final estimators $\hat{\mathbf{x}} = \mathbf{\Omega}^{+}\hat{\gamma}$ in both might be very similar; this is because in multiplying by $\mathbf{\Omega}^{+}$ we null-out content not in the column-span of $\mathbf{\Omega}$, essentially satisfying the constraint. However, as we will see, this does not turn out to close the gap between the two methods. The exception to this is the non-robust ℓ^2 case, in which equivalence still holds.

Theorem 2.3. Over-Complete Case – Equivalence for $p = 2$. *MAP-Analysis with a full-rank analyzing operator* $\mathbf{\Omega} \in M^{L \times N}$ *(*$L > N$*) is equivalent to MAP-Synthesis with* $\boldsymbol{D} = \mathbf{\Omega}^{+}$ *for* $p = 2$.

Proof. From (2.5) the proof is trivial. When $p = 2$, the unknown $\boldsymbol{\gamma}$ can be assumed to be the sum of two parts, $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{\mathbf{\Omega}} + \boldsymbol{\gamma}^{\mathbf{\Omega}_\perp}$, where $\boldsymbol{\gamma}^{\mathbf{\Omega}}$ comes from the column-span of $\mathbf{\Omega}$, and $\boldsymbol{\gamma}^{\mathbf{\Omega}_\perp}$ from the orthogonal subspace. The second penalty term ($\|\boldsymbol{\gamma}\|_2^2$) clearly prefers $\boldsymbol{\gamma}^{\mathbf{\Omega}_\perp}$ to be zero; as to the first term ($\|\mathbf{y} - \mathbf{T}\{\mathbf{\Omega}^{+}\boldsymbol{\gamma}\}\|_2^2$), $\boldsymbol{\gamma}^{\mathbf{\Omega}_\perp}$ has no impact on it as it is nulled-out by $\mathbf{\Omega}^{+}$. Thus, $\boldsymbol{\gamma}^{\mathbf{\Omega}_\perp}$ that violates the constraint in $\boldsymbol{\gamma}$ is chosen as zero, and the two methods coincide. □

2.3.1 MAP-Analysis and MAP-Synthesis in ℓ^1

From this point on we consider the two MAP methods with $p = 1$. The ℓ^1 choice is essentially the "meeting point" between the analysis and synthesis approaches, which prefer $p \geq 1$ and $0 \leq p \leq 1$ respectively. The use of the ℓ^1 norm in signal and image recovery has received considerable attention beginning at the late 1980's, with the adoption of robust statistics by the signal processing community. Probably most notable of the analysis-based methods is the Total-Variation approach [55], [1] with some additional examples including [60, 62, 90, 91]. Classical synthesis-based methods include the Basis Pursuit method [12] and the Lasso [96].

[1] Total variation takes a "true" MAP-Analysis form only in the 1D case.

For the ℓ^1 choice, we have the following forms of the two recovery processes:

$$\begin{aligned}\hat{\mathbf{x}}_{\mathrm{MAP-A}} &= \underset{\mathbf{x}}{\mathrm{Argmin}}\ \|\mathbf{y}-\mathbf{T}\{\mathbf{x}\}\|_2^2+\lambda\cdot\|\mathbf{\Omega x}\|_1\\ \hat{\mathbf{x}}_{\mathrm{MAP-S}} &= \mathbf{D}\cdot\underset{\boldsymbol{\gamma}}{\mathrm{Argmin}}\ \|\mathbf{y}-\mathbf{T}\{\mathbf{D}\boldsymbol{\gamma}\}\|_2^2+\lambda\cdot\|\boldsymbol{\gamma}\|_1\ .\end{aligned}$$

The ℓ^1 option is a favourable choice for these methods due to its combination of convexity, robustness, as well as proximity to ℓ^0 in the synthesis case [42, 95].

Looking at the two MAP formulations, we see that both depend on a weighting parameter λ to control the regularizing element; for $\lambda = 0$ both reproduce the ML estimator, and as $\lambda \to \infty$ they deviate from it until finally converging to 0. However, the rate at which this occurs may vary substantially between the two methods, and hence this parametrization is inconvenient for our purposes. To overcome this, we propose the following reformulations of the two problems:

$$\begin{aligned}\hat{\mathbf{x}}_{\mathrm{MAP-A}}(a) &= \underset{\mathbf{x}}{\mathrm{Argmin}}\ \|\mathbf{\Omega x}\|_1 \quad \text{Subject To} \quad \|\mathbf{y}-\mathbf{T}\{\mathbf{x}\}\|_2\le a\\ \hat{\mathbf{x}}_{\mathrm{MAP-S}}(a) &= \mathbf{D}\cdot\underset{\boldsymbol{\gamma}}{\mathrm{Argmin}}\ \|\boldsymbol{\gamma}\|_1 \quad \text{Subject To} \quad \|\mathbf{y}-\mathbf{T}\{\mathbf{D}\boldsymbol{\gamma}\}\|_2\le a\ .\end{aligned}$$

These formulations are conceptually simpler, with a directly controlling the deviation from the ML estimator. The original MAP target functions are essentially the Lagrangian functionals of these constrained versions (with λ representing the inverse of the Lagrange multiplier), and thus the two forms are equivalent.

2.3.2 A Geometrical Viewpoint

The above formulations have a simple geometrical interpretation, which provides an interesting way of comparing the two MAP approaches. The solutions of both problems are obviously confined to the same region of ”radius” a about $\mathbf{y}$ (this is true as we assume $\mathbf{D}$ to be full-rank); we also assume this region does not include the origin, otherwise the solution is trivially zero. Considering MAP-Analysis first, the level-sets of its target function $f_{\mathrm{A}}(\mathbf{x}) = \|\mathbf{\Omega x}\|_1$ are a collection of concentric, centro-symmetric polytopes $\{\mathbf{x} \mid \|\mathbf{\Omega x}\|_1 \le c\}$. Graphically, the solution can be obtained by taking a small level-set $\{\|\mathbf{\Omega x}\|_1 \le c\}$ about the origin, and gradually inflating it (by increasing c) until it first encounters the region $\{\|\mathbf{y}-\mathbf{T}\{\mathbf{x}\}\|_2 \leq a\}$. The point of intersection constitutes the solution to the MAP-Analysis problem, as there cannot be a point in this region having a smaller value of $\|\mathbf{\Omega x}\|_1$.

As to MAP-Synthesis, a similar process may be described using the collection of concentric, centro-symmetric polytopes $\mathbf{D}\cdot\{\boldsymbol{\gamma} \mid \|\boldsymbol{\gamma}\|_1 \leq c\}$[2]. This is reasoned as follows: consider the set $\mathbf{D}\cdot\{\|\boldsymbol{\gamma}\|_1 \leq c\}$ where c is small enough such that this set does not intersect the region $\{\|\mathbf{y} - \mathbf{T}\{\mathbf{x}\}\|_2 \leq a\}$. Then for any $\mathbf{x}$ in this region, there does not exist a representation $\boldsymbol{\gamma}$ satisfying $\|\boldsymbol{\gamma}\|_1 \leq c$, or in other words, any representation as $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$ must satisfy $\|\boldsymbol{\gamma}\|_1 > c$. This, of course, is true for any c which is small enough; therefore if we inflate this set (by enlarging c) until it first touches the region at the value $\hat{c}$, then for the intersection point $\hat{\mathbf{x}} = \mathbf{D}\hat{\boldsymbol{\gamma}}$ we know it has a representation satisfying $\|\hat{\boldsymbol{\gamma}}\|_1 = \hat{c}$, whereas for any $c < \hat{c}$ the signals within the region have no such representation, and hence $\hat{\mathbf{x}}$ must be the MAP-Synthesis solution.

Conveniently, for both MAP methods these ”inflations” are performed via simple scaling: we have $\{\|\mathbf{\Omega x}\|_1 \leq c\} = c\cdot\{\|\mathbf{\Omega x}\|_1 \leq 1\}$ and $\mathbf{D}\{\|\boldsymbol{\gamma}\|_1 \leq c\} = c\cdot\mathbf{D}\{\|\boldsymbol{\gamma}\|_1 \leq 1\}$. This implies that given the canonical *MAP defining polytopes* $\Psi_{\mathbf{\Omega}} := \{\|\mathbf{\Omega x}\|_1 \leq 1\}$ and $\Phi_{\mathbf{D}} := \mathbf{D}\cdot\{\|\boldsymbol{\gamma}\|_1 \leq 1\}$, the inflation processes are fully defined, and so are the MAP solutions; in fact, specifying these polytopes is completely equivalent to specifying $\mathbf{\Omega}$ or $\mathbf{D}$, respectively. We find that the behaviour of each of the methods is governed exclusively by the geometry of a single high-dimensional polytope, providing us with the basis for comparing the two methods.[3] We therefore continue by characterizing the geometry of these two polytopes.

Before continuing, we briefly review some elementary polytope terminology.

[2] Note that these sets exist in *signal space*, and have the explicit form $\{\mathbf{x} \mid \exists\boldsymbol{\gamma},\ \mathbf{x} = \mathbf{D}\boldsymbol{\gamma} \wedge \|\boldsymbol{\gamma}\|_1 \leq c\}$.

[3] In fact, the same arguments hold for any ℓ^p formulation, replacing the ℓ^1-norms in the definitions of $\Psi_{\mathbf{\Omega}}$ and $\Phi_{\mathbf{D}}$ with the proper ℓ^p-norms. However, analyzing these defining shapes for a general p is a difficult task, and thus we restrict ourselves to the ℓ^1 case.

Given an N-dimensional polytope, its boundary is an $(N-1)$-dimensional manifold; each of the polytope's *facets* is an $(N-1)$-dimensional surface constituting one segment of this manifold. A facet may also be referred to as an *(N-1)-dimensional face*. Similarly, the boundary of each facet consists of *(N-2)-dimensional faces* — and so on. A polytope's vertices, edges and ridges are its faces of dimensions 0, 1 and 2, respectively.

The MAP-Analysis Defining Polytope.

The MAP-Analysis defining polytope is a level set of the MAP-Analysis target function, $f_{\text{A}}(\mathbf{x}) = \|\mathbf{\Omega x}\|_1$:

$$\Psi_{\mathbf{\Omega}} = \{\mathbf{x} \mid \|\mathbf{\Omega x}\|_1 \leq 1\} \ .$$

Applying the gradient operator to f_{A}, we find that the normal to this surface satisfies

$$\mathbf{n}(\mathbf{x}) \ \propto \ \nabla f_{\text{A}}(\mathbf{x}) = \mathbf{\Omega}^T \text{sign}(\mathbf{\Omega x}) \ .$$

Evidently $\mathbf{n}(\mathbf{x})$ is defined for any $\mathbf{x}$ in which all coordinates of $\mathbf{\Omega x}$ are non-zero; where one or more of these vanishes, $\mathbf{n}(\mathbf{x})$ exhibits a discontinuity arbitrarily filled-in by the sign function. $\mathbf{n}(\mathbf{x})$ is therefore (as expected) piecewise-smooth. Intuitively, consider the signals $\mathbf{x}$ on the boundary of the defining polytope, then the facets correspond to the locations where $\mathbf{n}(\mathbf{x})$ is smooth, whereas the other faces correspond to where $\mathbf{n}(\mathbf{x})$ is discontinuous. The discontinuities in $\mathbf{n}(\mathbf{x})$ obviously result from $\mathbf{x}$ being orthogonal to rows in $\mathbf{\Omega}$; the following claim, whose proof is provided in the appendix, relates the face dimension to the rank of these rows:

Claim 2.1. *Let $\boldsymbol{x} \in \partial\Psi_{\mathbf{\Omega}}$ (the boundary of the defining polytope), and let k denote the rank of the rows in $\mathbf{\Omega}$ to which $\boldsymbol{x}$ is orthogonal to. Then $\boldsymbol{x}$ resides strictly within a face of dimension $(N-k-1)$ of the MAP-Analysis defining polytope.*

(Proof in 2.B.)

We use the term *strictly within a face* to indicate a signal located in the interior of a face, in the sense that there exists a finite ϵ-ball about it — of the same dimension as the face — entirely contained within this face (note that this also covers signals that are vertices, who reside strictly within themselves). Also, as opposed to standard residence, strict residence is *unique*, as the faces are considered open rather than closed, and thus do not overlap.

The claim implies that to obtain a vertex of $\Psi_{\mathbf{\Omega}}$, we choose $N-1$ linearly-independent rows in $\mathbf{\Omega}$, determine their 1D null-space $\mathbf{v}$ and normalize such that $\|\mathbf{\Omega}\mathbf{v}\|_1 = 1$ (note that this defines two antipodal vertices). Edges are similarly obtained by choosing $N-2$ linearly-independent rows, and taking any properly normalized signal in their 2D null-space. This leads to an immediate conclusion concerning the *vertex complexity* of the MAP-Analysis defining polytope, as its vertex count is equal to the number of possible choices of $N-1$ linearly-independent rows in $\mathbf{\Omega}$. In the worst-case, this may reach an exponential $\binom{L}{N-1}$, and in fact, this is a *tight bound* for the worst-case. As an example, assume the rows of $\mathbf{\Omega}$ are chosen such that their directions $\{\hat{\mathbf{w}}_i\}$ are uniformly distributed on the unit sphere. Under these conditions, the probability of any set of $N-1$ rows to be dependent *vanishes* for all practical purposes, and thus we obtain that for this randomized case the expected number of MAP-Analysis vertices achieves $\Theta\binom{L}{N-1}$. Obviously this is also the tight bound for the worst-case vertex count.

An interesting observation is that the MAP-Analysis defining polytope exhibits a highly regular structure. For instance, consider the set of edges associated with some choice of $N-2$ independent rows from $\mathbf{\Omega}$. Letting $\{\mathbf{u}, \mathbf{v}\}$ span their 2D null-space, these edges are obtained as any linear combination of the two (for instance of the form $\mathbf{x} = \cos(\theta)\mathbf{u} + \sin(\theta)\mathbf{v}$), properly normalized to ensure $\|\mathbf{\Omega}\mathbf{x}\|_1 = 1$. It follows that this set of edges forms a closed *edge-loop* of the polytope; the planar edge loop consists of consecutive edges, all existing on a common plane. We conclude that the edges of $\Psi_{\mathbf{\Omega}}$ are arranged in ”loops” about the origin, each loop associated with a choice of $N-2$ independent rows from $\mathbf{\Omega}$. Similar arguments generalize to higher-dimensional regularities, corresponding to the choices of $N-k$ independent rows from $\mathbf{\Omega}$ for $k>2$.

Finally, the organized structure is also found in a highly regular *neighbourliness pattern*. Since every vertex is obtained as the null-space of some $N-1$ rows from $\mathbf{\Omega}$, and each choice of $N-2$ of these defines an edge loop passing through this vertex, we have that each vertex of $\Psi_{\mathbf{\Omega}}$ is incident to *exactly* $N-1$ edge loops, and consequently, every vertex of $\Psi_{\mathbf{\Omega}}$ has precisely $2(N-1)$ neighbours.

The MAP-Synthesis Defining Polytope.

The MAP-Synthesis defining polytope is given by

$$\Phi_{\mathbf{D}} = \mathbf{D}\cdot\{\boldsymbol{\gamma}\mid\|\boldsymbol{\gamma}\|_1\leq 1\}\ .$$

It is a known result that this polytope is obtained as the convex hull of the columns of $\mathbf{D}$ and $-\mathbf{D}$; a proof is brought in the appendix for completeness:

Claim 2.2. *The MAP-Synthesis defining polytope $\Phi_{\boldsymbol{D}}=\boldsymbol{D}\cdot\{\|\boldsymbol{\gamma}\|_1\leq 1\}$ is obtained as the convex hull of $\{\pm\boldsymbol{d}_i\}_{i=1\ldots L}$, where $\{\boldsymbol{d}_i\}$ are the columns of $\boldsymbol{D}$.*

(Proof in 2.C.)

The claim simply states that the vertices of the MAP-Synthesis defining polytope are those columns of $\pm\mathbf{D}$ which cannot be represented as a convex combination of any other columns (and their antipodes); the other faces are the convex combinations of neighbouring vertices. A vertex can therefore be represented as $\mathbf{v}=\mathbf{D}\boldsymbol{\gamma}$ where $\boldsymbol{\gamma}$ has a single non-zero element $\gamma_i=\pm1$, and a point on an edge can be represented similarly with $\boldsymbol{\gamma}$ having two non-vanishing elements γ_i,γ_j satisfying $|\gamma_i|+|\gamma_j|=1$. In general, a point on a k-dimensional face will have a representation $\mathbf{x}=\mathbf{D}\boldsymbol{\gamma}$ with $\boldsymbol{\gamma}$ having $k+1$ non-vanishing elements, and $\|\boldsymbol{\gamma}\|_1=1$. We emphasize that this is *not* a sufficient condition, so a signal $\mathbf{x}=\mathbf{D}\boldsymbol{\gamma}$ synthesized from a sparse representation $\boldsymbol{\gamma}$ might not reside on a low-dimensional face if the corresponding columns of $\pm\mathbf{D}$ are not neighbours, or do not constitute polytope vertices.

An immediate implication of Claim 2.2 concerns the redundancy of certain atom signals in $\mathbf{D}$. From the claim, it is clear that any column of $\mathbf{D}$ residing strictly within the convex hull of the remaining columns has absolutely no effect on the MAP-Synthesis defining polytope — and thus can be removed.

Corollary 2.1. *Let $\boldsymbol{d}_k$ be a column of $\boldsymbol{D}$ which is obtained as a convex combination of the remaining columns and their antipodes, $\{\pm\boldsymbol{d}_i\}_{i=1..\hat{k}..L}$. Then the MAP-Synthesis problem obtained by removing $\boldsymbol{d}_k$ from $\boldsymbol{D}$ is equivalent to the original one.*

Redundant columns in $\mathbf{D}$ can be safely removed without altering the MAP-Synthesis solution, and by locating these we may be able to prune the dictionary, generally obtaining a simpler formulation. The problem of determining whether some vector $\mathbf{x}$ is a convex combination of the set $\{\mathbf{y}_i\}$ can be formulated as a linear-programming (LP) problem, and thus locating all redundant columns in $\mathbf{D}$ requires L executions of LP. As an alternative to removal, we may choose to elongate the redundant atom such that it becomes a vertex of the MAP-Synthesis defining polytope, and thus expressed by the prior. However, increasing a dictionary atom may have the effect of assimilating a different one into the convex hull. One simple method to ensure none of the columns in $\mathbf{D}$ are redundant is to normalize them to a *fixed length* (see section 2.3.3 below and specifically Claim 2.3).

2.3.3 Consequences of the Geometrical Viewpoint

The geometrical analysis leads to some important consequences concerning the two MAP methods. In this section we describe a few of these conclusions.

The Analysis-Synthesis Gap.

From the geometrical viewpoint, we find that contrary to the algebraic similarity, the analysis and synthesis structures are actually very different. As we have seen, the two polytopal structures asymptotically differ in their vertex counts. A parallel difference exists in the neighbourliness properties of these polytopes; since every vertex has a linear number of neighbours in the MAP-Analysis case (while their total number is exponential) it follows that the probability of any two vertices to be neighbours approaches 0 as $N \to \infty$. In contrast, Donoho [97] has recently shown that for MAP-Synthesis polytopes, the probability of any 2 (non-antipodal) vertices to be neighbours approaches 1 as $N \to \infty$.[4] We find that while MAP-Analysis polytopes feature very *large* numbers of vertices with very *low* neighbourliness, MAP-Synthesis polytopes exhibit *low* vertex counts and very *high* neighbourliness. Combined with the high regularity of the MAP-Analysis polytopes, we see that the two approaches actually describe very different structures. These theoretical gaps indeed translate to very concrete behavioural differences between the two methods, and this will be shown in the experiments section.

MAP-Synthesis as a Superset of MAP-Analysis.

An interesting consequence of the geometrical description is that any ℓ^1 MAP-Analysis estimator may be reformulated as an equivalent MAP-Synthesis one. This is accomplished by simply taking all the MAP-Analysis defining polytope vertices — one of each antipodal pair — and setting them as the MAP-Synthesis dictionary atoms. Since both methods will have the same defining polytope, they will be completely equivalent. This establishes the generality of MAP-Synthesis over MAP-Analysis in ℓ^1:

[4] The dictionary is assumed to be of linear size in N, as well as to fulfill certain randomness conditions; see Theorem 1 in [97].

Theorem 2.4. Over-Complete ℓ^1 Case – Generality of MAP-Synthesis. *For any ℓ^1 MAP-Analysis form with full-rank analyzing operator $\mathbf{\Omega}$ ($L \geq N$), there exists a dictionary $\boldsymbol{D}(\mathbf{\Omega})$ describing an equivalent ℓ^1 MAP-Synthesis form. The reverse is not true.*

The reverse direction fails due to the strict regularity imposed on the MAP-Analysis defining polytopes. Since this regularity does not apply to MAP-Synthesis, it may clearly describe structures not represented in the MAP-Analysis form.

The actual equivalence transform presented here has little practical value; except for the special case of $N = 2$, where the size of $\mathbf{D}(\mathbf{\Omega})$ will be equal to (or even smaller than) that of $\mathbf{\Omega}^T$, the size of $\mathbf{D}(\mathbf{\Omega})$ will generally grow exponentially. Nonetheless, the theorem describes a definite one-way relationship between the two formulations: the synthesis formulation is clearly more general than the analysis one, with indeed a *vast collection* of MAP-Synthesis priors unrepresented by the stricter MAP-Analysis form.

MAP Principal Signals.

The constructive nature of MAP-Synthesis provides a good understanding of the signals which are most "favoured" by this prior; in essence, these are the dictionary atoms and their sparse combinations. The parallel entities for the MAP-Analysis prior, however, are difficult to derive using algebraic tools. The geometric interpretation enables us to define these qualitative terms in a precise manner, and give a description of the MAP-Analysis counterparts of the synthesis atoms.

Roughly speaking, we consider a signal to be favoured by some prior when this prior is capable of recovering the signal well given deteriorated versions of it; intuitively, these should be the signals with maximal a-priori probability. However, we observe that both MAP structures are energy-dependent; therefore, the most probable signals for both are simply the zero signal and its immediate neighbourhood. Moreover, the intuition itself here is not entirely accurate: a highly probable signal will not be well-recovered if there exists a near-by signal with even higher probability.

To resolve this, we confine ourselves to a *fixed-energy* sphere; on this sphere we seek the most effectively recovered signals by the specific MAP method. Since the recovery is a local process, we will further be interested in the *local* maxima of the distribution on this sphere rather than the global ones. Our line of thought can be described as follows. Consider an energy-preserving denoising process, where the denoised solution is post-processed by re-normalizing it to the magnitude of the input (thus eliminating its decay to zero caused by the low-energy preference of the prior). Under these conditions, the MAP estimation essentially searches the neighbourhood of the input on the fixed-energy sphere, outputting a higher-probability (and presumably less noisy) signal near the input. A signal will therefore be well-recovered when its prior probability is maximal relative to a significant enough part of its neighbourhood on the fixed-energy sphere. Specifically, the *local maxima* of the distribution will be the most effectively recovered signals on the sphere.

Reducing w.l.o.g. to the unit sphere, we refer to the local maxima of the distribution as the *principal signals* of the distribution. Formally,

Definition 2.1. *Let Prob{$\boldsymbol{x}$} be any MAP-Analysis or MAP-Synthesis distribution. Then the principal signals of this distribution are defined as the* ***local*** *maxima of the optimization problem*

$$\underset{\mathbf{x}}{\text{Argmax}}\ \ \text{Prob}\{\mathbf{x}\} \qquad \text{Subject To} \quad \|\mathbf{x}\|_2 = 1\ .$$

As we will soon see, in the synthesis case these signals are tightly related to the MAP-Synthesis dictionary atoms.

The geometry of the MAP defining polytope directly dictates the behaviour of the distribution on the unit sphere, and consequently the locations of the principal

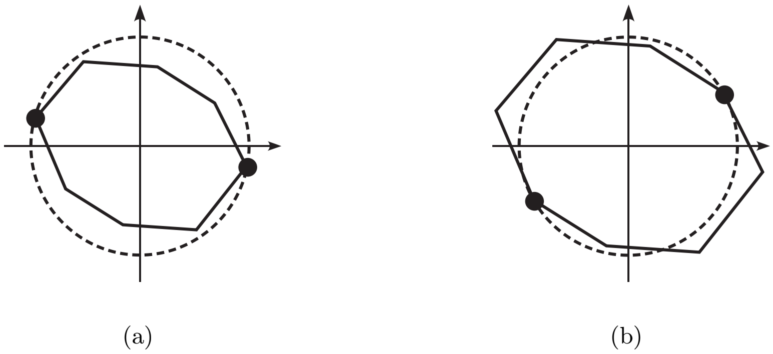


Figure 2.1: Principal signals and the MAP defining polytope. The dotted circles denote the unit sphere in 2D signal space. The two polygons are different scales of the same MAP defining polytope. (a) A principal signal, intersected by a vertex of the defining polytope. (b) A vertex which is not a principal signal.

signals on it. For both priors, the boundaries of the defining polytopes define iso-surfaces of signals with equal a-priori probability; these have the form $r \cdot \partial\Psi_{\mathbf{\Omega}}$ or $r \cdot \partial\Phi_{\mathbf{D}}$ — where $r \in \mathbb{R}^+$ is a non-negative scaling factor — and for increasing r, represent decreasingly probable signals. Beginning with such an iso-surface $r \cdot \partial\Psi_{\mathbf{\Omega}}$ or $r \cdot \partial\Phi_{\mathbf{D}}$, with small enough r such that it is entirely bounded by the unit sphere, then as r is increased, the surface intersects the sphere at decreasingly probably locations, until finally completely enclosing it. Clearly, to be a local maximum a signal must be intersected by the inflating iso-surfaces before its surrounding neighbourhood. Consequently, such a local maximum is intersected by an extreme point – a *vertex* – of the polytope. We conclude that the MAP principal signals project to *vertices* of the MAP defining polytope.

We immediately point out, however, that projection onto a vertex is only a *necessary* condition for principality, as demonstrated in Figure 2.1. Simulation results show a dramatic difference in the recovery performance of principal vs. non-principal polytope vertices.

For a vertex to be principal, it must be maximally distant from the origin relative to all the directions about it on the boundary of the defining polytope. Luckily, determining this only requires examining those directions from the vertex to its one-dimensional incident edges (this follows from the fact that for any scalar function, the convex combination of a set of descent directions is also a descent direction).

In the case of MAP-Synthesis, its defining polytope vertices are a subset of the dictionary atoms, hence the principal signals are a subset of these atoms. However, not all atoms constitute polytope vertices, and only a few of these are actually principal. Furthermore, determining which of the atoms are vertices is a difficult task, and so is the task of determining the incident edges of each vertex. However, given an atom $\mathbf{d}$, a simple work-around to determine its principality is to examine *all* line segments connecting $\mathbf{d}$ with the remaining atoms and their antipodes. If $\mathbf{d}$ is found to be maximally distant relative to all these line segments, clearly it is a vertex as well as a principal signal; on the other hand, if $\mathbf{d}$ is found not to be maximal relative to some segment, it immediately follows that it is not principal.

In practice, many MAP-Synthesis dictionaries have their atoms *normalized* to a fixed length. As we mentioned earlier (without proof) this ensures that all the atoms constitute defining polytope vertices. However, for such dictionaries, a stronger claim can be made: indeed, when the atoms are normalized, they all constitute principal signals of the MAP distribution. We have the following result, whose proof is provided in the appendix:

Claim 2.3. Principal Signals of MAP-Synthesis with a Normalized Dictionary. *Let $\boldsymbol{D}$ be a MAP-Synthesis dictionary with fixed-energy columns. Then the dictionary atoms coincide with the principal signals of the MAP-Synthesis prior.*

(Proof in 2.D.)

In the general case, however, the MAP-Synthesis principal signals remain a subset of the dictionary atoms. Since dictionaries in practice are commonly normalized, this distinction is not usually made. Nevertheless, when the dictionary atoms are not normalized, the difference in recovery performance can be substantial; while the principal signals are truly ”favoured” by the prior, other atoms might not be at all.

In the MAP-Analysis case, the distinction becomes more significant. The number of MAP-Analysis vertices is exponentially large, and empirical evidence suggests that most of these are non-principal and not well-recovered. Unfortunately, we are not currently aware of any simple analytical method for characterizing the MAP-Analysis principal signals. Nonetheless, these signals can be generated by computer. For the simulations in this paper we used a simple traversal algorithm for locating these signals; this enabled us to produce large sets of MAP-Analysis principal signals and study their behaviour.

Our traversal algorithm locates one principal signal at a time. Beginning with some initial vertex $\mathbf{v}$, we examine its incident edge-loops, and for each loop, we determine $\mathbf{u}$ such that $\{\mathbf{v}, \mathbf{u}\}$ orthogonally span the plane in which the loop exists. Assuming a small enough ϵ, $\mathbf{v}$'s infinitesimal neighbours on this edge loop can be approximated by $\mathbf{v}_+ = (\mathbf{v} + \epsilon\mathbf{u})/\|\mathbf{\Omega}(\mathbf{v} + \epsilon\mathbf{u})\|_1$ and $\mathbf{v}_- = (\mathbf{v} - \epsilon\mathbf{u})/\|\mathbf{\Omega}(\mathbf{v} - \epsilon\mathbf{u})\|_1$, where the normalization is applied to ensure $\|\mathbf{\Omega}\mathbf{v}_+\|_1 = \|\mathbf{\Omega}\mathbf{v}_-\|_1 = 1$. By comparing the ℓ^2 norms of $\mathbf{v}$, $\mathbf{v}_+$ and $\mathbf{v}_-$, we determine whether $\mathbf{v}$ is maximal relative to its two incident edges on this edge loop. Now, if $\mathbf{v}$ is found to be maximal relative to all its incident edges, it is a principal signal. Otherwise, it is not maximal relative to some incident edge. In this case we replace it with a vertex with larger ℓ^2-norm from the violating edge loop (in our implementation, we choose the one with largest ℓ^2-norm in the loop), and continue the traversal. This swapping continues until a local maximum is encountered, providing one MAP-Analysis principal signal. The entire process is then repeated using a new vertex as a starting point.

2.4 Numerical Results

The geometrical viewpoint reveals a large gap between the two formulations in the over-determined ℓ^1 case. In this section we provide some simulation results, demonstrating this theoretical gap.

2.4.1 Synthetic Experiments

The following synthetic experiments demonstrate how the gap can be easily brought to an extreme even in a simple case. To obtain these results we compared the two methods on their most favourable signals: their principal signals.

For the experiment, we selected the pseudo-inverse relation between the dictionary and analysis operator; this is a natural choice for bridging the two methods, however in reality, it may lead to very different behaviours of the two methods. We selected the 128×256 *Identity-Hadamard* dictionary $\mathbf{D} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{H} \end{bmatrix}$ and its pseudo-inverse $\mathbf{\Omega} = \mathbf{D}^T = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{H} \end{bmatrix}^T$ as the synthesis dictionary and analysis operator. This is an interesting choice as the two feature the same two-ortho structure, and furthermore $\mathbf{D}$ is a near-optimal Grassmanian frame, making it favourable for MAP-Synthesis methods [98, 99].

The dictionary size immediately limits the number of distinct MAP-Synthesis principal signals to a mere 256. In contrast, MAP-Analysis boasts an enormous number of them: our traversal algorithm easily produced 10 000 such signals. What's more, our program was designed to reject new signals if these resided in a radius of < 0.1 from any existing principal signal; however, after 10 000 generated signals, the rejection rate remained negligible, suggesting that the true number of such signals is much greater (with an only known upper bound of order $\binom{L}{N-1} = \binom{256}{127} \approx 10^{75}$). These are obviously impressive numbers compared to the modest number of MAP-Synthesis principal signals.

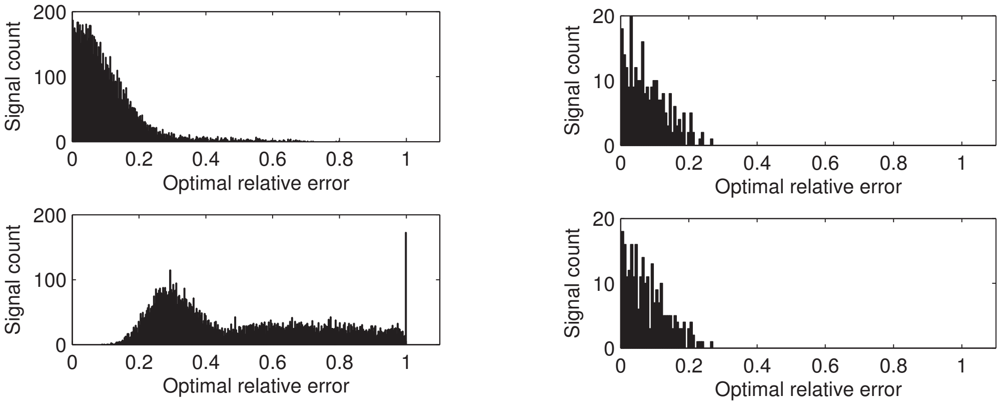
An interesting point in this experiment is that the MAP-Synthesis principal signals in our case all double as MAP-Analysis principal signals. To sharpen the comparison, we therefore generated additional sets of preferable MAP-Synthesis signals, which we obtained on low-dimensional faces of the MAP-Synthesis defining polytope (i.e., sparse combinations of atoms). For the experiment, we generated 1000 signals on 2D faces, 1000 on 3D faces, and so on up to 12D faces.

To quantify the performance of a specific method on a set of signals, we generated noisy versions of the signals in the set, and applied the method (in its energy-preserving form), with varying a values, to each of the contaminated signals. We then selected, for each signal individually, the optimal a value a_{opt} and its associated relative error $err_{opt} = \|\widehat{\mathbf{x}}_{\mathrm{MAP}}(a_{opt}) - \mathbf{x}\|_2 / \|\mathbf{y} - \mathbf{x}\|_2$ to represent the performance of the method on this signal. We collected the optimal errors for all signals in the set, and these were used to characterize the performance of the method on the entire set.

Figures 2.2-2.4 summarize the results. The first two present histograms of the optimal errors obtained on the principal signal sets and the MAP-Synthesis 2D and 3D signal sets. The final figure summarizes the results for all 12 sets of MAP-Synthesis signals.

The results demonstrate several points. First, we see that each method is indeed successful in recovering its own sets of principal signals; this agrees with the predictions of the geometrical model. Also interesting is the fact that the two methods exhibit comparable performance when evaluated each on their own set of principal signals; this observation is particularly evident from Figure 2.2(b), where the signals are simultaneously principal to both MAP-Analysis and MAP-Synthesis.

On the other hand, the results also depict a clear disparity between the two methods. We see that MAP-Analysis completely fails in recovering the MAP-Synthesis favourable signals, while MAP-Synthesis performs notably poorly compared to MAP-Analysis on its massive number of principal signals. The results also illustrate the asymptotical nature of gap between the two approaches in the



(a)

(b)

Figure 2.2: Denoising MAP principal signals. (a) Results for MAP-Analysis principal signal (10 000 examples): distributions of optimal errors obtained using MAP-Analysis (above) and MAP-Synthesis (below). (b) The same for MAP-Synthesis principal signals (256 examples).

number of principal signals each one accepts.

The acute inconsistencies lead to the inevitable conclusion that the pseudo-inverse relation does not bridge between the two methods. Moreover, we see here that the difference in complexity between the two structures has a strong expression in practice, indicative of an inherent gap between the two formulations. Though the experiment specifically utilizes the pseudo-inverse relation, the gap depicted here cannot be associated to this specific choice; indeed, any reasonably-sized MAP-Synthesis dictionary will be limited in the number of favourable signals it can accommodate, and consequently in its ability to handle the large number of MAP-Analysis principal signals. In the other direction, any attempt to adapt a MAP-Analysis prior to a given set of MAP-Synthesis signals is bound to give rise to an enormous number of additional (unwanted) favourable signals.

2.4.2 Real-World Experiments

In this section we present some comparative denoising results obtained for actual image data. For these experiments we selected the *overcomplete DCT* transform; this transform partitions the image into overlapping blocks, and applies to each

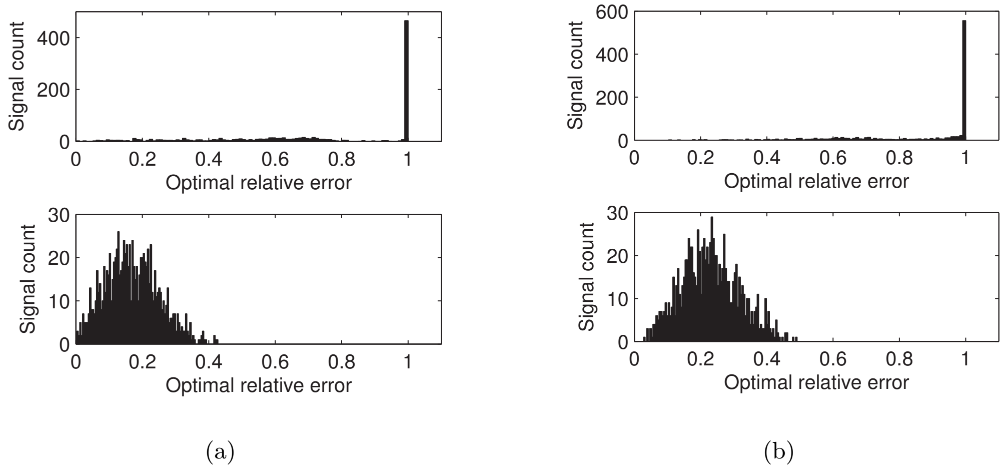


Figure 2.3: Denoising signals on low-dimensional MAP-Synthesis faces. (a) Results for signals on 2D faces (1000 examples): distributions of optimal errors obtained using MAP-Analysis (above) and MAP-Synthesis (below). (b) The same for signals on 3D faces (1000 examples).

block a unitary DCT transform. The overcomplete DCT transform constitutes a tight frame when all image pixels are covered by an equal number of blocks. Our experiments used 8 × 8 blocks, with a shift of either 1, 2 or 4 pixels between neighbouring blocks. We also used shifts of 8 pixels (i.e. no overlap, leading to a unitary transform) as reference. Boundary cases were handled by assuming periodicity, ensuring the tight frame condition.

Since the transform is tight, the synthesis dictionary was simply taken as the transpose of the analysis operator, leading to a dictionary constructed of 8 × 8 DCT bases in all possible shifts over the image domain. Motivations for choosing this transform include: (1) The transform is widely used in image processing, and has been employed in both analysis and synthesis frameworks; (2) it is a tight frame, and has an efficient implementation; and (3) it is highly redundant, whilst offering a convenient way for controlling its redundancy (specifically, 4× for a shift size of 4, 16× for a shift size of 2, and 64× for a shift size of 1).

We ran the experiments on a collection of standard test images, including *Lenna*, *Barbara* and *Mandrill*. Each of these was downscaled to a size of 128×128

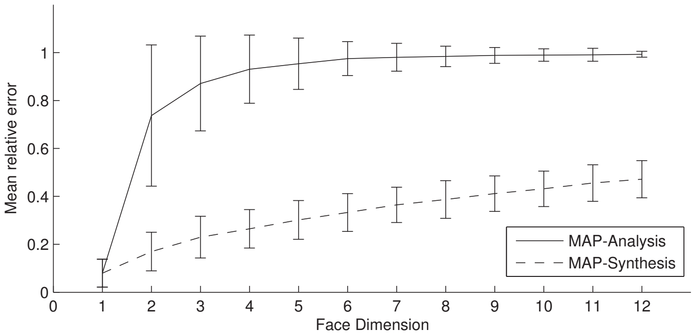
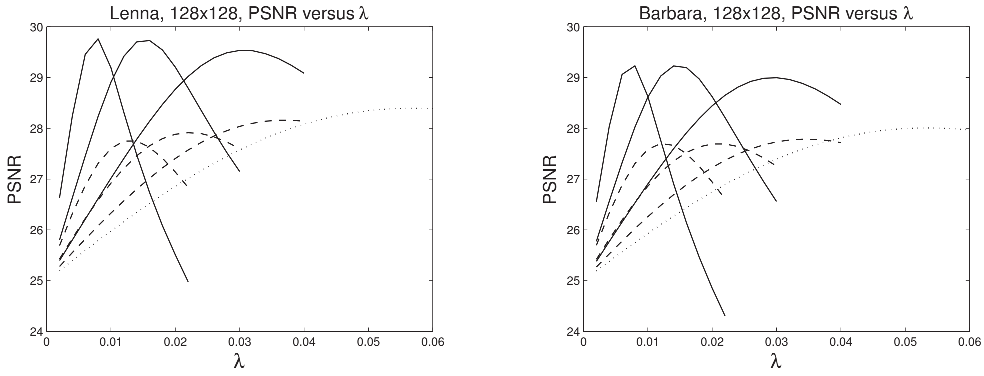


Figure 2.4: Denoising MAP-Synthesis highly recoverable signals. The graphs show the mean optimal errors obtained versus the MAP-Synthesis face dimension; error bars correspond to the standard deviation of the errors.

to reduce computation costs. We added white Gaussian noise to each source image, producing 25dB PSNR inputs. Each input was denoised using both MAP-Analysis and MAP-Synthesis with varying λ values, and the output PSNR was determined for each value.

The results for *Lenna* and *Barbara* are shown in Figure 2.5. The results for *Mandrill* were similar. As can be seen in the figures, the results are quite surprising: MAP-Analysis actually beats MAP-Synthesis — in a convincing way — in every test. Compared to the baseline unitary transform (dotted line), where both methods coincide, MAP-Analysis (solid) shows a significant gain when introducing overcompleteness, which slightly improves as the redundancy increases; in contrast, MAP-Synthesis (dashed) shows slightly *degraded* performance as the overcompleteness is increased. As a consequence, the distance between the two methods grows with the redundancy.

The experiments presented here were also carried out using the Contourlet transform [18], which has a 4:3 redundancy factor. In these experiments the two methods led to almost identical outputs, an outcome which conforms with the low redundancy of the transform. Interestingly, however, the picture remained the same: in all tests, MAP-Analysis actually showed a small edge over MAP-



(a)

(b)

Figure 2.5: Image denoising using the redundant DCT transform. Solid lines, left to right: MAP-Analysis with block shifts of 1, 2 and 4 pixels; dashed lines, left to right: MAP-Synthesis with block shifts of 1,2 and 4 pixels; dotted line: MAP-Analysis/MAP-Synthesis with a block shift of 8 pixels (unitary transform). Images are of size 128×128. (a) Results for *Lenna* (b) Results for *Barbara*. Images downloaded from `http://www.wikipedia.com`, and downscaled using bilinear interpolation.

Synthesis.

The reasons for the superiority of MAP-Analysis in the denoising scenario require further study; however, in our context we see that the gap indeed exists, and can become dramatic even in practical situations. One possible explanation for this could be the advantage of MAP-Analysis discussed in Section 2.1.3: since MAP-Analysis utilizes all its filters simultaneously to support the recovery process, it may be more robust in the presence of noise compared to MAP-Synthesis, whose compact representation may be unstable when noise is introduced, leading to recovery errors. A different possibility is that the high overcompleteness in MAP-Synthesis, rather than positively enriching its descriptiveness, leads to a reverse effect where the dictionary becomes ”too descriptive”, representing a wide range of undesirable signals. This effect does not apply to MAP-Analysis where increasing the number of filters still requires the signal to agree with all existing ones.

2.5 Conclusions: Analysis versus Synthesis Revisited

We began our discussion presenting two popular MAP-based methods for inverse problem regularization — the MAP-Analysis and the MAP-Synthesis approaches — and showing the algebraical similarity between the two. We saw that the two are equivalent in the square non-singular case as well as in the under-complete denoising case; however, in the overcomplete case the two methods were shown to depart. We concentrated on the interesting ℓ^1 case, and found that the geometrical structures underlying the two exhibited very different properties. This perspective has led to a generality relation of MAP-Synthesis over MAP-Analysis, as well as to the characterization of the MAP-Analysis parallels of the MAP-Synthesis atoms.

The geometrical model does not provide a definite answer to the question of *who is better*. It does, however, shed some light on the real gap that exists between the two approaches, a gap which is not evident from the algebra alone. We have used the geometrical model to locate those signals where the gap is expected to be the largest, leading us to the results of the synthetic experiments; we saw that for these signals the gap indeed becomes large. The experiments also demonstrated the asymptotical nature of the difference between the two structures in their number of principal signals. Our real-world experiments showed that this gap exists not only in theory, and, no less important, that MAP-Synthesis should not be a-priori considered to be superior to MAP-Analysis.

Our results are not to be interpreted as a recommendation for this method or another. The synthetic experiments indicate that each of the methods is successful, only on *different sets of signals*. The real-world experiments, which demonstrated a significant advantage to MAP-Analysis, should be regarded as a sample case rather than a conclusion. MAP-Synthesis remains advantageous in its simplicity of dictionary design, and we further emphasize that the interesting ℓ^0 MAP-Synthesis case, though generally close to the ℓ^1 case, has not been treated. Nonetheless, as MAP-Analysis is *significantly* simpler to solve, our results come to emphasize that despite the recent blossom of MAP-Synthesis methods, both approaches are still worthy candidates for inverse problem regularization. The question of which will actually be better for a specific application and family of signals, remains open.

Acknowledgements

The authors would like to thank Prof. David L. Donoho for the enlightening discussions and ideas which helped in developing the presented work. This research was supported by the USA-Israel binational science foundation (BSF) grant #2004199.

2.A Equivalence in the Undercomplete Case

Theorem 2.2. Under-Complete Denoising Case – Near-Equivalence. *MAP-Analysis denoising with a full-rank analyzing operator $\mathbf{\Omega} \in M^{[L \times N]}$ $(L \leq N)$ is nearly-equivalent to MAP-Synthesis with the dictionary $\boldsymbol{D} = \mathbf{\Omega}^+$. This is expressed by the relation $\hat{\boldsymbol{x}}_{\text{MAP-A}} = \hat{\boldsymbol{x}}_{\text{MAP-S}} + \boldsymbol{y}^{\boldsymbol{D}\perp}$, with $\boldsymbol{y}^{\boldsymbol{D}\perp}$ representing the component of the input orthogonal to the columns of $\boldsymbol{D}$.*

Proof. In the following, we assume the relation $\mathbf{D} = \mathbf{\Omega}^+$; we additionally assume that $\mathbf{\Omega}$ has full row-rank (equivalently, that $\mathbf{D}$ has full column-rank), and thus $\mathbf{D} = \mathbf{\Omega}^T(\mathbf{\Omega}\mathbf{\Omega}^T)^{-1}$ and $\mathbf{\Omega}\mathbf{D} = I$. We introduce the notation $\mathbf{z} = \mathbf{z}^{\mathbf{D}} + \mathbf{z}^{\mathbf{D}\perp}$ to denote the (single) decomposition of a signal $\mathbf{z}$ to the part $\mathbf{z}^{\mathbf{D}}$ in the column-span of $\mathbf{D}$ and the part $\mathbf{z}^{\mathbf{D}\perp}$ in the orthogonal subspace.

We begin with the MAP-Analysis formulation in (2.2):

$$\hat{\mathbf{x}}_{\text{MAP-A}} = \underset{\mathbf{x}}{\text{Argmin}} \ \ \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \cdot \|\mathbf{\Omega}\mathbf{x}\|_p^p \ .$$

Decomposing in respect to the column-span of $\mathbf{D}$, we obtain

$$\begin{aligned}\hat{\mathbf{x}}_{\text{MAP-A}} &= \underset{\mathbf{x}^{\mathbf{D}},\mathbf{x}^{\mathbf{D}\perp}}{\text{Argmin}} \|\mathbf{y}^{\mathbf{D}} + \mathbf{y}^{\mathbf{D}\perp} - \mathbf{x}^{\mathbf{D}} - \mathbf{x}^{\mathbf{D}\perp}\|_2^2 + \ \lambda \cdot \|\mathbf{\Omega}(\mathbf{x}^{\mathbf{D}} + \mathbf{x}^{\mathbf{D}\perp})\|_p^p \\ &= \underset{\mathbf{x}^{\mathbf{D}},\mathbf{x}^{\mathbf{D}\perp}}{\text{Argmin}} \|\mathbf{y}^{\mathbf{D}} - \mathbf{x}^{\mathbf{D}}\|_2^2 + \|\mathbf{y}^{\mathbf{D}\perp} - \mathbf{x}^{\mathbf{D}\perp}\|_2^2 + \ \lambda \cdot \|\mathbf{\Omega}\mathbf{x}^{\mathbf{D}} + \mathbf{\Omega}\mathbf{x}^{\mathbf{D}\perp}\|_p^p \ .\end{aligned}$$

We note that $\mathbf{z}$ is orthogonal to the columns of $\mathbf{D}$ iff it is orthogonal to the rows of $\mathbf{\Omega}$: since $\mathbf{\Omega\Omega}^T$ is invertible, we have $0 = \mathbf{D}^T\mathbf{z} \iff 0 = (\mathbf{\Omega\Omega}^T)\mathbf{D}^T\mathbf{z} = (\mathbf{\Omega\Omega}^T)(\mathbf{\Omega\Omega}^T)^{-1}\mathbf{\Omega z} = \mathbf{\Omega z}$. This implies $\mathbf{\Omega x}^{\mathbf{D}\perp} = 0$, leading to

$$\hat{\mathbf{x}}_{\mathrm{MAP-A}} = \underset{\mathbf{x}^{\mathbf{D}},\mathbf{x}^{\mathbf{D}\perp}}{\mathrm{Argmin}} \ \|\mathbf{y}^{\mathbf{D}} - \mathbf{x}^{\mathbf{D}}\|_2^2 + \|\mathbf{y}^{\mathbf{D}\perp} - \mathbf{x}^{\mathbf{D}\perp}\|_2^2 + \lambda \cdot \|\mathbf{\Omega x}^{\mathbf{D}}\|_p^p \,.$$

Obviously any solution to this problem will satisfy $\hat{\mathbf{x}}^{\mathbf{D}\perp} = \mathbf{y}^{\mathbf{D}\perp}$, as there is no additional penalty term for $\mathbf{x}^{\mathbf{D}\perp}$. Therefore the MAP-Analysis problem reduces to an optimization problem for $\hat{\mathbf{x}}^{\mathbf{D}}_{\mathrm{MAP-A}}$:

$$\hat{\mathbf{x}}^{\mathbf{D}}_{\mathrm{MAP-A}} \ = \ \underset{\mathbf{x}^{\mathbf{D}}}{\mathrm{Argmin}} \ \|\mathbf{y}^{\mathbf{D}} - \mathbf{x}^{\mathbf{D}}\|_2^2 + \lambda \cdot \|\mathbf{\Omega x}^{\mathbf{D}}\|_p^p \,.$$

Signals $\mathbf{x}^{\mathbf{D}}$ spanned by the columns of $\mathbf{D}$ have a representation as $\mathbf{x}^{\mathbf{D}} = \mathbf{D}\boldsymbol{\gamma}$. We can thus reformulate the above as an optimization on $\boldsymbol{\gamma}$, leading to

$$\begin{aligned}\hat{\mathbf{x}}^{\mathbf{D}}_{\mathrm{MAP-A}} &= \mathbf{D} \cdot \underset{\boldsymbol{\gamma}}{\mathrm{Argmin}} \ \|\mathbf{y}^{\mathbf{D}} - \mathbf{D}\boldsymbol{\gamma}\|_2^2 + \lambda \cdot \|\mathbf{\Omega D}\boldsymbol{\gamma}\|_p^p \\ &= \mathbf{D} \cdot \underset{\boldsymbol{\gamma}}{\mathrm{Argmin}} \ \|\mathbf{y}^{\mathbf{D}} - \mathbf{D}\boldsymbol{\gamma}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p\end{aligned}$$

We see that the solution to $\hat{\mathbf{x}}^{\mathbf{D}}_{\mathrm{MAP-A}}$ comes from a MAP-Synthesis structure with $\mathbf{D} = \mathbf{\Omega}^+$, and applied to $\mathbf{y}^{\mathbf{D}}$. We conclude by showing that $\mathbf{y}^{\mathbf{D}}$ in this formulation may be replaced with $\mathbf{y}$. We do this using similar arguments to those applied above, in a reverse manner:

$$\begin{aligned}\hat{\mathbf{x}}^{\mathbf{D}}_{\mathrm{MAP-A}} &= \mathbf{D} \cdot \underset{\boldsymbol{\gamma}}{\mathrm{Argmin}} \ \|\mathbf{y}^{\mathbf{D}} - \mathbf{D}\boldsymbol{\gamma}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p \\ &= \mathbf{D} \cdot \underset{\boldsymbol{\gamma}}{\mathrm{Argmin}} \ \|\mathbf{y}^{\mathbf{D}} - \mathbf{D}\boldsymbol{\gamma}\|_2^2 + \|\mathbf{y}^{\mathbf{D}\perp}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p \\ &= \mathbf{D} \cdot \underset{\boldsymbol{\gamma}}{\mathrm{Argmin}} \ \|\mathbf{y}^{\mathbf{D}} + \mathbf{y}^{\mathbf{D}\perp} - \mathbf{D}\boldsymbol{\gamma}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p \\ &= \mathbf{D} \cdot \underset{\boldsymbol{\gamma}}{\mathrm{Argmin}} \ \|\mathbf{y} - \mathbf{D}\boldsymbol{\gamma}\|_2^2 + \lambda \cdot \|\boldsymbol{\gamma}\|_p^p\end{aligned}$$

Summing up, for the (under-)determined case, and with the relation $\mathbf{D} = \mathbf{\Omega}^+$, we have shown that given a signal $\mathbf{y} = \mathbf{y}^{\mathbf{D}} + \mathbf{y}^{\mathbf{D}\perp}$, the MAP-Analysis solution and the MAP-Synthesis solution are related by $\hat{\mathbf{x}}_{\mathrm{MAP-A}} = \hat{\mathbf{x}}_{\mathrm{MAP-S}} + \mathbf{y}^{\mathbf{D}\perp}$, as claimed. □

2.B MAP-Analysis Defining Polytope

Lemma 2.1. Facets of the MAP-Analysis Defining Polytope. *Let $\boldsymbol{x} \in \partial\Psi_{\boldsymbol{\Omega}}$, where $\Psi_{\boldsymbol{\Omega}}$ is the MAP-Analysis defining polytope $\{\boldsymbol{x} \mid \|\boldsymbol{\Omega x}\|_1 \leq 1\}$. If $\boldsymbol{\Omega x}$ has no vanishing elements, then $\boldsymbol{x}$ resides strictly within a facet ($N-1$-dimensional face) of the MAP-Analysis defining polytope.*

Proof. Let $f_{\mathrm{A}}(\mathbf{x}) = \|\boldsymbol{\Omega}\mathbf{x}\|_1$ (the MAP-Analysis target function), and assume $\boldsymbol{\Omega}\mathbf{x}$ has no vanishing elements; then $\nabla f_{\mathrm{A}}(\mathbf{x}) = \boldsymbol{\Omega}^T \mathrm{sign}(\boldsymbol{\Omega}\mathbf{x})$, and is defined at $\mathbf{x}$. Also, since all elements of $\boldsymbol{\Omega}\mathbf{x}$ are finite and non-zero, there exists a ball $\mathcal{B}_\epsilon(\mathbf{x})$ around $\mathbf{x}$ such that for all $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x})$, $\boldsymbol{\Omega}\mathbf{x}$ has no vanishing elements. Now consider the intersection $\partial\Psi_{\boldsymbol{\Omega}} \cap \mathcal{B}_\epsilon(\mathbf{x})$: this is a neighbourhood of $\mathbf{x}$ on the boundary of the defining polytope, and for all $\mathbf{x}$ in it, $\boldsymbol{\Omega}\mathbf{x}$ has no zero coordinates. From continuity of $\boldsymbol{\Omega}\mathbf{x}$, we conclude that none of its coordinates change sign within this neighbourhood, so for all $\mathbf{x}$ in it, $\mathrm{sign}(\boldsymbol{\Omega}\mathbf{x}) = \mathrm{sign}(\boldsymbol{\Omega}\mathbf{x})$ and also $\nabla f_{\mathrm{A}}(\mathbf{x}) = \nabla f_{\mathrm{A}}(\mathbf{x})$. As the defining polytope is a level-set of f_{A}, ∇f_{A} (where defined) designates the direction of the normal to this polytope. We therefore have a finite neighbourhood of $\mathbf{x}$ on the boundary of the polytope where the normal is fixed, and thus $\mathbf{x}$ must reside strictly within a facet of this polytope. □

We now bring the proof of Claim 2.1, generalizing the above lemma.

Claim 2.1. Faces of the MAP-Analysis Defining Polytope. *Let $\boldsymbol{x} \in \partial\Psi_{\boldsymbol{\Omega}}$, and let k denote the rank of the rows in $\boldsymbol{\Omega}$ to which $\boldsymbol{x}$ is orthogonal to. Then $\boldsymbol{x}$ resides strictly within a face of dimension $(N-k-1)$ of the MAP-Analysis defining polytope.*

Proof. Assume a signal $\mathbf{x} \in \partial\Psi_{\boldsymbol{\Omega}}$. Let $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ orthonormally span the rows in $\boldsymbol{\Omega}$ to which $\mathbf{x}$ is orthogonal to, and let $\{\mathbf{u}_1, \ldots, \mathbf{u}_{N-k}\}$ span their complementary space. We denote $\mathcal{U} = \mathrm{Span}\{\mathbf{u}_i\}$ and $\mathcal{W} = \mathrm{Span}\{\mathbf{w}_j\}$. Clearly $\mathbf{x} \in \mathcal{U}$, from orthogonality to $\{\mathbf{w}_j\}$.

First, we consider the space $\mathcal{U}$. Any vector $\mathbf{v} \in \mathcal{U}$ may be written as $\mathbf{v} = \mathbf{U}\boldsymbol{\alpha}(\mathbf{v})$, where $\mathbf{U} = [\mathbf{u}_1 \mid \ldots \mid \mathbf{u}_{N-k}]$ is an $N \times (N-k)$ matrix, and $\boldsymbol{\alpha}(\mathbf{v}) = \mathbf{U}^T\mathbf{v}$. Since $\mathbf{v} \in \mathcal{U}$, it is orthogonal to all the rows in $\boldsymbol{\Omega}$ to which $\mathbf{x}$ is orthogonal to; therefore, letting $\widehat{\boldsymbol{\Omega}}$ be the matrix obtained by discarding these rows from $\boldsymbol{\Omega}$, then for any $\mathbf{v} \in \mathcal{U}$, we have $\|\boldsymbol{\Omega}\mathbf{v}\|_1 = \|\widehat{\boldsymbol{\Omega}}\mathbf{v}\|_1$. Note that since we assume $\boldsymbol{\Omega}$ is full rank, then after removing from it the rows whose span is $\mathcal{W}$, the remaining rows of $\widehat{\boldsymbol{\Omega}}$ still span *at least* the complement space $\mathcal{U}$.

Now, define $\mathcal{W} = \widehat{\boldsymbol{\Omega}}\mathbf{U}$; we have $\|\boldsymbol{\Omega}\mathbf{v}\|_1 = \|\widehat{\boldsymbol{\Omega}}\mathbf{v}\|_1 = \|\widehat{\boldsymbol{\Omega}}\mathbf{U}\boldsymbol{\alpha}(\mathbf{v})\|_1 = \|\mathcal{W}\boldsymbol{\alpha}(\mathbf{v})\|_1$ for any $\mathbf{v} \in \mathcal{U}$. Multiplying $\widehat{\boldsymbol{\Omega}}$ to the left of $\mathbf{U}$ is essentially an orthogonal projection of its rows on the subspace $\mathcal{U}$; since the rows of $\widehat{\boldsymbol{\Omega}}$ span $\mathcal{U}$, the rank of the result must be equal to that of $\mathbf{U}$. Therefore the rank of $\mathcal{W}$ is $(N-k)$, so it must have at least this number of rows, and is thus an overcomplete analysis operator on the $\boldsymbol{\alpha}$-space.

Since $\mathbf{x} \in \mathcal{U}$, all the equalities above hold for $\mathbf{x}$. Specifically, $\|\mathcal{W}\boldsymbol{\alpha}(\mathbf{x})\|_1 = \|\boldsymbol{\Omega}\mathbf{x}\|_1 = 1$, so by definition $\boldsymbol{\alpha}(\mathbf{x}) \in \partial\Psi_{\mathcal{W}}$. In other words, $\boldsymbol{\alpha}(\mathbf{x})$ must reside on the boundary of the defining polytope corresponding to the $(N-k)$-dimensional MAP-Analysis problem for the $\boldsymbol{\alpha}$-space with operator $\mathcal{W}$. We further know that $\widehat{\boldsymbol{\Omega}}\mathbf{x}$ has no vanishing elements, since all such elements have been removed, so $\mathcal{W}\boldsymbol{\alpha}(\mathbf{x}) = \widehat{\boldsymbol{\Omega}}\mathbf{x}$ has no vanishing elements. We have thus established all the conditions of Lemma 2.1 for $\boldsymbol{\alpha}(\mathbf{x})$, and it follows that $\boldsymbol{\alpha}(\mathbf{x})$ resides strictly within a facet of the $(N-k)$-dimensional polytope $\Psi_{\mathcal{W}}$.

Given this, we know there exists an $(N-k-1)$-dimensional ball about $\boldsymbol{\alpha}(\mathbf{x})$ such that this ball is entirely contained within the boundary of $\Psi_{\mathcal{W}}$. By applying $\mathbf{U}$ to the points of this ball, we orthonormally inject it to the N-dimensional signal space, obtaining an $(N-k-1)$-dimensional ball about $\mathbf{x} = \mathbf{U}\boldsymbol{\alpha}(\mathbf{x})$. This ball resides entirely on the boundary of $\Psi_{\boldsymbol{\Omega}}$, since for any signal $\mathbf{x} = \mathbf{U}\boldsymbol{\alpha}(\mathbf{x})$ in this ball, $\mathbf{x} \in \mathcal{U}$ and so $\|\boldsymbol{\Omega}\mathbf{x}\|_1 = \|\mathcal{W}\boldsymbol{\alpha}(\mathbf{x})\|_1 = 1$. Evidently, we have an $(N-k-1)$-dimensional ball about $\mathbf{x}$, residing entirely on the boundary of the defining polytope, therefore $\mathbf{x}$ must reside on a face of dimension *at least* $(N-k-1)$ of this polytope. To conclude the proof, we show this residence is strict; in other words, we prove that there does not exist a ball of higher dimension about $\mathbf{x}$ residing entirely within the polytope's boundary.

Consider a d-dimensional ball about $\mathbf{x}$, contained entirely within the boundary of the defining polytope; then for any point $\mathbf{x}+\mathbf{e}$ in this ball, the point $\mathbf{x}-\mathbf{e}$ is also in the ball. Now, write $\mathbf{e}$ as

$$\mathbf{e} = \sum_i a_i \mathbf{u}_i + \sum_j b_j \mathbf{w}_j \ ,$$

where $\{\mathbf{u}_i\}$ and $\{\mathbf{w}_j\}$ are the orthonormal bases as defined above. Since both points are on the boundary of $\Psi_{\mathbf{\Omega}}$, we have $\|\mathbf{\Omega}\mathbf{x}\|_1 = \|\mathbf{\Omega}(\mathbf{x}+\mathbf{e})\|_1 = \|\mathbf{\Omega}(\mathbf{x}-\mathbf{e})\|_1 = 1$. Written explicitly, these expand to

$$\mathbf{\Omega}(\mathbf{x} \pm \mathbf{e}) = \mathbf{\Omega}\left[\mathbf{x} \pm \left(\sum a_i \mathbf{u}_i + \sum b_j \mathbf{w}_j\right)\right] \ .$$

Since $(\mathbf{x} \pm \sum a_i \mathbf{u}_i) \in \mathcal{U}$, all vanishing coefficients in $\mathbf{\Omega}\mathbf{x}$ also vanish in $\mathbf{\Omega}\left(\mathbf{x} \pm \sum a_i \mathbf{u}_i\right)$. As to the second part, assume by contradiction that $\sum b_j \mathbf{w}_j \in \mathcal{W}$ is non-zero. Clearly the same coefficients cannot all vanish in $\mathbf{\Omega}\sum b_j \mathbf{w}_j$, as the corresponding rows in $\mathbf{\Omega}$ span $\mathcal{W}$. Therefore adding or subtracting $\mathbf{\Omega}\mathbf{e}$ to $\mathbf{\Omega}\mathbf{x}$ necessarily *increases* the absolute-value-sum of these coefficients. On the other hand, the entire ℓ^1 norm of $\mathbf{\Omega}(\mathbf{x} \pm \mathbf{e})$ remains fixed; so, for the remainder of the coefficients, the addition or subtraction of $\mathbf{\Omega}\mathbf{e}$ must strictly *reduce* their absolute-value-sum. However, this may not occur simultaneously for both addition *and* subtraction. Therefore, the only resolution to this is to require $b_j \equiv 0$ for all j, implying that necessarily $\mathbf{e} \in \mathcal{U}$. Thus, we have limited the dimension of the ball about $\mathbf{x}$ to $N-k$ (the dimension of $\mathcal{U}$). Finally, $\mathbf{x} \in \mathcal{U}$, but clearly $\|\mathbf{\Omega}(\mathbf{x}+\delta\mathbf{x})\|_1 \neq \|\mathbf{\Omega}\mathbf{x}\|_1$ for any $\delta \neq 0$. So $\mathbf{e}$ cannot be proportional to $\mathbf{x}$, and hence the ball about $\mathbf{x}$ must be of dimension less than $\mathcal{U}$. We conclude that $d \leq (N-k-1)$, so $\mathbf{x}$ can reside strictly within a face of dimension no more than $(N-k-1)$. Since we have already shown the existence of such a face, we conclude that $\mathbf{x}$ resides strictly within an $(N-k-1)$-dimensional face of the MAP-Analysis defining polytope, as claimed. □

2.C MAP-Synthesis Defining Polytope

Claim 2.2. Geometry of the MAP-Synthesis Defining Polytope. *The MAP-Synthesis defining polytope $\Phi_{\boldsymbol{D}} = \boldsymbol{D}\{\|\boldsymbol{\gamma}\|_1 \leq 1\}$ is obtained as the convex hull of $\{\pm\boldsymbol{d}_i\}_{i=1\ldots L}$, where $\{\boldsymbol{d}_i\}$ are the columns of $\boldsymbol{D}$.*

Proof. For the proof we note that $\mathbf{d}_i = \mathbf{D}\mathbf{e}_i$, where $\{\mathbf{e}_i\}$ is the standard basis of $\mathbb{R}^L$. We introduce the notation $\mathcal{CH}\{\mathbf{v}_i\}$ to denote the convex hull of the set $\{\mathbf{v}_i\}$.

$\mathcal{CH}\{\mathbf{d_i}\} \subseteq \mathbf{\Phi_D}$: We have $\pm\mathbf{e}_i \in \{\|\boldsymbol{\gamma}\|_1 \leq 1\}$ for all i, and therefore $\pm\mathbf{d}_i = \mathbf{D}(\pm\mathbf{e}_i) \in \mathbf{D}\{\|\boldsymbol{\gamma}\|_1 \leq 1\} = \Phi_{\mathbf{D}}$. Since $\Phi_{\mathbf{D}}$ is convex, it must also contain the convex hull of $\{\pm\mathbf{d}_i\}$.

$\mathbf{\Phi_D} \subseteq \mathcal{CH}\{\mathbf{d_i}\}$: Let $\mathbf{x} \in \Phi_{\mathbf{D}}$, then there exists a representation $\boldsymbol{\gamma}$ such that $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$, where $\|\boldsymbol{\gamma}\|_1 \leq 1$. Since $\boldsymbol{\gamma} \in \{\|\boldsymbol{\gamma}\|_1 \leq 1\}$, it is a convex combination of $\{\pm\mathbf{e}_i\}$, and can be written as $\boldsymbol{\gamma} = \sum_i \big\{a_i\mathbf{e}_i + b_i(-\mathbf{e}_i)\big\}$. This implies $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma} = \sum_i \Big\{a_i\mathbf{d}_i + b_i(-\mathbf{d}_i)\Big\}$, so $\mathbf{x}$ is a convex combination of $\{\pm\mathbf{d}_i\}$, and as such exists in their convex hull. □

2.D MAP-Synthesis with a Normalized Dictionary

Lemma 2.2. *Let $\mathcal{P}$ be a polytope with fixed-length vertices, i.e., for all vertices $\boldsymbol{v}$ of $\mathcal{P}$, $\|\boldsymbol{v}\|_2 = c$ for some constant c. Then for every non-vertex point $\boldsymbol{p}$ on the boundary of the polytope, $\|\boldsymbol{p}\|_2 < c$.*

Proof. Consider a facet φ of $\mathcal{P}$, defined by the vertices $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. This facet constitutes the intersection of some $(n-1)$-dimensional hyperplane with the polytope. Now, consider the ℓ^2-norm function $f(\mathbf{x}) = \|\mathbf{x}\|_2$, constrained to this plane. The iso-surfaces of f on this plane are a set of concentric ellipsoids about some central point of minimal ℓ^2-norm. Since $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ are of a fixed length, they all reside on the same ellipsoid. The facet φ, which is the convex hull of $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$, must thus exist entirely within this ellipsoid by definition of the convex hull as the minimal convex set containing $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. This implies that for every $\mathbf{p} \in \varphi$, $\|\mathbf{p}\|_2 \leq c$.

To obtain sharp inequality, we assume by contradiction that $\|\mathbf{p}\|_2 = c$ while $\mathbf{p}$ is not a vertex. Since $\mathbf{p}$ is not a vertex, there exist two points $\mathbf{p}_1, \mathbf{p}_2 \in \varphi$ such that $\mathbf{p}$ resides on the line connecting $\mathbf{p}_1$ and $\mathbf{p}_2$. However, examining the function f, we have the following observation: for any point in space, advancing from it in two opposite directions will always lead to *at least* one direction of increase in f; this is due to the fact that when constrained to an infinite line, f always achieves a single minimum and no maximum on the line. This implies that at least one of $\mathbf{p}_1$ and $\mathbf{p}_2$ will have ℓ^2-norm larger than c, leading to a contradiction. Hence necessarily $\|\mathbf{p}\|_2 < c$. □

Claim 2.3. Principal Signals of MAP-Synthesis with a Normalized Dictionary. *Let $\boldsymbol{D}$ be a MAP-Synthesis dictionary with fixed-energy columns. Then the dictionary atoms coincide with the principal signals of the MAP-Synthesis prior.*

Proof. From Lemma 2.2, the proof is trivial. Let us denote the length of the dictionary atoms by c. Then for any atom $\mathbf{d}$, it follows that it must be a vertex as $\|\mathbf{d}\|_2 = c$. Now, assume by contradiction that $\mathbf{d}$ is non-principal; therefore there exists a direction from $\mathbf{d}$ on the boundary of the defining polytope such that the distance from the origin increases. However this means that if we advance from $\mathbf{d}$ in this direction a short enough distance, we will obtain a non-vertex point on the polytope boundary whose length is larger than c, contradicting the previous lemma. We conclude that $\mathbf{d}$ must be a principal signal. □

Chapter 3

Dictionaries for Sparse Representation Modeling

Published in Proceedings of the IEEE, 98(6):1045–1057, 2010.

Coauthored with Alfred M. Bruckstein and Michael Elad.

Abstract

Sparse and redundant representation modeling of data assumes an ability to describe signals as linear combinations of a few atoms from a pre-specified dictionary. As such, the choice of the dictionary that sparsifies the signals is crucial for the success of this model. In general, the choice of a proper dictionary can be done using one of two ways: (i) building a sparsifying dictionary based on a mathematical model of the data, or (ii) learning a dictionary to perform best on a training set. In this paper we describe the evolution of these two paradigms. As manifestations of the first approach, we cover topics such as wavelets, wavelet packets, contourlets, and curvelets, all aiming to exploit 1-D and 2-D mathematical models for constructing effective dictionaries for signals and images. Dictionary learning takes a different route, attaching the dictionary to a set of examples it is supposed to serve. From the seminal work of Field and Olshausen, through the MOD, the

K-SVD, the Generalized PCA and others, this paper surveys the various options such training has to offer, up to the most recent contributions and structures.

3.1 Introduction

The process of digitally sampling a natural signal leads to its representation as the sum of Delta functions in space or time. This representation, while convenient for the purposes of display or playback, is mostly inefficient for analysis tasks. Signal processing techniques commonly require more meaningful representations which capture the useful characteristics of the signal — for recognition, the representation should highlight salient features; for denoising, the representation should efficiently separate signal and noise; and for compression, the representation should capture a large part of the signal with only a few coefficients. Interestingly, in many cases these seemingly different goals align, sharing a core desire for *simplification*.

Representing a signal involves the choice of a *dictionary*, which is the set of elementary signals – or *atoms* – used to decompose the signal. When the dictionary forms a basis, every signal is uniquely represented as the linear combination of the dictionary atoms. In the simplest case the dictionary is orthogonal, and the representation coefficients can be computed as inner products of the signal and the atoms; in the non-orthogonal case, the coefficients are the inner products of the signal and the dictionary inverse, also referred to as the bi-orthogonal dictionary.

For years, orthogonal and bi-orthogonal dictionaries were dominant due to their mathematical simplicity. However, the weakness of these dictionaries — namely their limited expressiveness — eventually outweighed their simplicity. This led to the development of newer *overcomplete* dictionaries, having more atoms than the dimensions of the signal, which promised to represent a wider range of signal phenomena.

The move to overcomplete dictionaries was done cautiously, in an attempt to minimize the loss of favorable properties offered by orthogonal transforms. Many dictionaries formed *tight frames*, which ensured that the representation of the signal as a linear combination of the atoms could still be identified with the inner products of the signal and the dictionary. Another approach, manifested by the *Best Basis* algorithm, utilized a specific dictionary structure which essentially allowed it to serve as a pool of atoms from which an *orthogonal* sub-dictionary could be efficiently selected.

Research on *general* overcomplete dictionaries mostly commenced over the past decade, and is still intensely ongoing. Such dictionaries introduce an intriguing ambiguity in the definition of a signal representation. We consider the dictionary $\mathbf{D} = [\mathbf{d}_1\, \mathbf{d}_2 \ldots \mathbf{d}_L] \in \mathbb{R}^{N\times L}$, where the columns constitute the dictionary atoms, and $L \geq N$. Representing a signal $\mathbf{x} \in \mathbb{R}^N$ using this dictionary can take one of two paths — either the *analysis* path, where the signal is represented via its inner products with the atoms,

$$\boldsymbol{\gamma}_a = \mathbf{D}^T\mathbf{x} \ , \tag{3.1}$$

or the *synthesis* path, where it is represented as a linear combination of the atoms,

$$\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}_s \ . \tag{3.2}$$

The two definitions coincide in the complete case ($L = N$), when the analysis and synthesis dictionaries are bi-orthogonal. In the general case, however, the two may dramatically differ.

The synthesis approach poses yet another interesting question: when $\mathbf{D}$ is overcomplete, the family of representations $\boldsymbol{\gamma}_s$ satisfying (3.2) is actually *infinitely large*, with the degrees of freedom identified with the null-space of $\mathbf{D}$. This allows us to seek the most informative representation of the signal with respect to some cost function $C(\boldsymbol{\gamma})$:

$$\boldsymbol{\gamma}_s = \underset{\boldsymbol{\gamma}}{\text{Argmin}}\; C(\boldsymbol{\gamma}) \quad \text{Subject To} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\gamma} \ . \tag{3.3}$$

Practical choices of $C(\boldsymbol{\gamma})$ promote the *sparsity* of the representation, meaning that we want the sorted coefficients to decay quickly. Solving (3.3) is thus commonly referred to as *sparse coding*. We can achieve sparsity by choosing $C(\gamma)$ as some robust penalty function, which we loosely define as a function that is tolerant to large coefficients but aggressively penalizes small non-zero coefficients. Examples include the Huber function [100] as well as the various ℓ^p cost functions with $0 \leq p \leq 1$.

The two options (3.1) and (3.2), and specifically the problem (3.3), have been extensively studied over the past few years. This in turn has led to the development of new signal processing algorithms which utilize general overcomplete transforms. However, in going from theory to practice, the challenge of *choosing* the proper dictionary for a given task must be addressed. Earlier works made use of traditional dictionaries, such as the Fourier and wavelet dictionaries, which are simple to use and perform adequately for 1-dimensional signals. However, these dictionaries are not well equipped for representing more complex natural and high-dimensional signal data, and new and improved dictionary structures were sought.

A variety of dictionaries have been developed in response to the rising need. These dictionaries emerge from one of two sources — either a *mathematical model* of the data, or a *set of realizations* of the data. Dictionaries of the first type are characterized by an analytic formulation and a fast implicit implementation, while dictionaries of the second type deliver increased flexibility and the ability to adapt to specific signal data. Most recently, there is a growing interest in dictionaries which can mediate between the two types, and offer the advantages of both worlds. Such structures are just beginning to emerge, and research is still ongoing.

In this paper we present the fundamental concepts guiding modern dictionary design, and outline the various contributions in the field. In Section 3.2 we take a historical viewpoint, and trace the evolution of dictionary design methodology from the early 1960's to the late 1990's, focusing on the conceptual advancements. In Sections 3.3 and 3.4 we overview the state-of-the art techniques in both analytic and trained dictionaries. We summarize and conclude in Section 3.5.

3.2 A History of Transform Design

3.2.1 Signal Transforms: The Linear Era

Signal transforms have been around for as long as signal processing has been conducted. In the 1960's, early signal processing researchers gave significant attention to linear time-invariant operators, which were simple and intuitive processes for manipulating analog and digital signals. In this scenery, the Fourier transform naturally emerged as the basis which diagonalizes these operators, and it immediately became a central tool for analyzing and designing such operators. The transform gained tremendous popularity with the introduction of the Fast Fourier Transform (FFT) in 1965 by Cooley and Tukey [101], which provided its numerical appeal.

The Fourier basis describes a signal in terms of its global frequency content, as a combination of orthogonal waveforms

$$\mathcal{F} = \left\{ \phi_n(x) = e^{inx} \right\}_{n \in \mathbb{Z}} \ .$$

A signal is approximated in this basis by projecting it onto the K lowest frequency atoms, which has a strong smoothing and noise-reducing effect. The Fourier basis is thus efficient at describing uniformly *smooth* signals. However, the lack of localization makes it difficult to represent *discontinuities*, which generate large coefficients over all frequencies. Therefore, the Fourier transform typically produces oversmooth results in practical applications. For finite signals, the Fourier transform implicitly assumes a periodic extension of the signal, which introduces a discontinuity at the boundary. The Discrete Cosine Transform (DCT) is the result of assuming an anti-symmetric extension of the signal, which results in continuous boundaries, and hence in a more efficient approximation. Since the DCT has the added advantage of producing non-complex coefficients, it is typically preferred in practical applications; see Fig. 3.1 for some 2-D DCT atoms.

Signal approximation in the Fourier basis was soon recognized as a specific instance of *linear approximation*: given a basis $\{\boldsymbol{\phi}_n\}_{n=0}^{N-1}$ of $\mathbb{R}^N$, a signal $\mathbf{x} \in \mathbb{R}^N$ is linearly approximated by projecting it onto a *fixed* subset of $K < N$ basis elements

$$\mathbf{x} \approx \sum_{n \in I_K} (\boldsymbol{\psi}_n^T \mathbf{x}) \boldsymbol{\phi}_n \ , \tag{3.4}$$

where $\{\boldsymbol{\psi}_n\}_{n=0}^{N-1}$ is in general the bi-orthogonal basis ($\boldsymbol{\psi}_n = \boldsymbol{\phi}_n$ in the orthonormal case). The process is an under-complete linear transform of $\mathbf{x}$, and, with the right choice of basis, can achieve *compaction* — the ability to capture a significant part of the signal with only a few coefficients. Indeed, this concept of *compaction* will later be replaced with *sparsity*, though the two are closely related [102].

Optimizing compaction was a major driving force for the continued development of more efficient representations. During the 1970's and 1980's, a new and very appealing source of compaction was brought to light: *the data itself.* The focus was on a set of statistical tools developed during the first half of the century, known as the Karhunen-Loève Transform (KLT) [7, 103], or Principal Component Analysis (PCA) [104]. The KLT is a linear transform which can be adapted to represent signals coming from a certain known distribution. The adaptation process fits a low-dimensional subspace to the data which minimizes the ℓ^2 approximation error. Specifically, given the data covariance matrix $\boldsymbol{\Sigma}$ (either known or empirical), the KLT atoms are the first K eigenvectors of the eigenvalue decomposition of $\boldsymbol{\Sigma}$,

$$\boldsymbol{\Sigma} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T \ .$$

From a statistical point of view, this process models the data as coming from a low-dimensional Gaussian distribution, and thus is most effective for Gaussian data. Fig. 3.1 shows an example of the KLT basis trained from a set of image patches. The DCT basis, shown in the same figure, is regarded as a good approximation of the KLT for natural image patches when a non-adaptive transform is required.

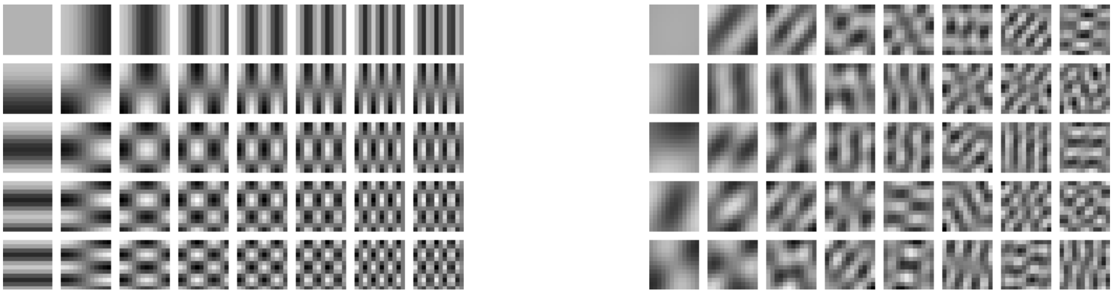


Figure 3.1: Left: a few 12×12 DCT atoms. Right: the first 40 KLT atoms, trained using 12×12 image patches from *Lena*.

Compared to the Fourier transform, the KLT is superior (by construction) in terms of representation efficiency. However, this advantage comes at the cost of a non-structured and substantially more complex transform. As we will see, this tradeoff between *efficiency* and *adaptivity* continues to play a major role in modern dictionary design methodology as well.

3.2.2 Non-Linear Revolution and Elements of Modern Dictionary Design

In statistics research, the 1980's saw the rise of a new powerful approach known as *robust statistics.* Robust statistics advocates sparsity as a key for a wide range of recovery and analysis tasks. The idea has its roots in classical Physics, and more recently in Information Theory, and promotes simplicity and conciseness in guiding phenomena descriptions. Motivated by these ideas, the 1980's and 1990's were characterized by a search for sparser representations and more efficient transforms.

Increasing sparsity required departure from the linear model, towards a more flexible *non-linear* formulation. In the non-linear case, each signal is allowed to use a different set of atoms from the dictionary in order to achieve the best approximation. Thus, the approximation process becomes

$$\mathbf{x} \approx \sum_{n \in I_K(\mathbf{x})} c_n \boldsymbol{\phi}_n \ , \tag{3.5}$$

where $I_K(\mathbf{x})$ is an index set adapted to each signal individually (we refer the reader to [7, 105] for a more thorough discussion of this topic).

The non-linear view paved the way to the design of newer, more efficient transforms. In the process, many of the fundamental concepts guiding modern dictionary design were formed. Following the historic time line, we trace the emergence of the most important modern dictionary design concepts, which were mostly formed during the last two decades of the 20th century.

Localization: To achieve sparsity, transforms required better localization. Atoms with concentrated supports allow more flexible representations based on the local signal characteristics, and limit the effects of irregularities, which are observed to be the main source of large coefficients. In this spirit, one of the first structures to be used was the Short Time Fourier Transform (STFT) [106], which emerges as a natural extension to the Fourier transform. In the STFT, the Fourier transform is applied locally to (possibly overlapping) portions of the signal, revealing a *time-frequency* (or space-frequency) description of the signal. An example of the STFT is the JPEG image compression algorithm [107], which is based on this concept.

During the 1980's and 1990's, the STFT was extensively researched and generalized, becoming more known as the *Gabor* transform — named in homage of Dennis Gabor, who first suggested the time-frequency decomposition back in 1946 [108]. Gabor's work was independently rediscovered in 1980 by Bastiaans [109] and Janssen [110], who studied the fundamental properties of the expansion.

A basic 1-D Gabor dictionary consists of windowed waveforms

$$\mathcal{G} = \left\{ \phi_{n,m}(x) = w(x - \beta m) e^{i 2\pi \alpha n x} \right\}_{n,m \in \mathbb{Z}} ,$$

where $w(\cdot)$ is a low-pass window function localized at 0 (typically a Gaussian), and α and β control the time and frequency resolutions of the transform. Much of the mathematical foundations of this transform were laid out during the late 1980's by Daubechies, Grossman and Meyer [111, 112] who studied the transform from the angle of frame theory, and by Feichtinger and Gröchenig [113–115] who employed a generalized group-theoretic point of view. Study of the discrete version of the transform and its numerical implementation followed in the early 1990's, with notable contributions by Wexler and Raz [116] and by Qian and Chen [117].

In higher dimensions, more complex Gabor structures were developed which add *directionality*, by varying the orientation of the sinusoidal waves. This structure gained substantial support from the work of Daugman [118, 119], who discovered oriented Gabor-like patterns in simple-cell receptive fields in the visual cortex. These results motivated the deployment of the transform to image processing tasks, led by works such as Daugman [120] and Porat and Zeevi [121]. Today, practical uses of the Gabor transform are mainly in analysis and detection tasks, as a collection of directional filters. Fig. 3.2 shows some examples of 2-D Gabor atoms of various orientations and sizes.

Multi-Resolution: One of the most significant conceptual advancements achieved in the 1980's was the rise of *multi-scale* analysis. It was realized that natural signals, and images specifically, exhibited meaningful structures over many scales, and could be analyzed and described particularly efficiently by multi-scale constructions. One of the simplest and best known such structures is the *Laplacian pyramid*, introduced in 1984 by Burt and Adelson [122]. The Laplacian pyramid represents an image as a series of difference images, where each one corresponds to a different scale and roughly a different frequency band.

In the second half of the 1980's, though, the signal processing community was particularly excited about the development of a new very powerful tool, known as *wavelet analysis* [7, 123, 124]. In a pioneering work from 1984, Grossman and Morlet [125] proposed a signal expansion over a series of translated and dilated versions of a single elementary function, taking the form

$$\mathcal{W} = \left\{ \phi_{n,m}(x) = \alpha^{n/2} f(\alpha^n x - \beta m) \right\}_{n,m \in \mathbb{Z}} \ .$$

This simple idea captivated the signal processing and harmonic analysis communities, and in a series of influential works by Meyer, Daubechies, Mallat and others [111, 112, 126–131], an extensive wavelet theory was formalized. The theory was formulated for both the continuous and discrete domains, with a complete mathematical framework relating the two. A significant breakthrough came from Meyer's work in 1985 [126], who found that unlike the Gabor transform (and contrary to common belief) the wavelet transform could be designed to be *orthogonal* while maintaining stability — an extremely appealing property to which much of the initial success of the wavelets can be attributed to.

Specifically of interest to the signal processing community was the work of Mallat and his colleagues [129–131] which established the wavelet decomposition as a multi-resolution expansion and put forth efficient algorithms for computing it. In Mallat's description, a multi-scale wavelet basis is constructed from a pair of localized functions referred to as the *scaling function* and the *mother wavelet*, see Fig. 3.3. The scaling function is a low frequency signal, and along with its translations, spans the coarse approximation of the signal. The mother wavelet is a high frequency signal, and with its various scales and translations spans the signal detail. In the orthogonal case, the wavelet basis functions at each scale are critically sampled, spanning precisely the new detail introduced by the finer level.

Non-linear approximation in the wavelet basis was shown to be optimal for piecewise-smooth 1-D signals with a finite number of discontinuities, see e.g., [130]. This was a striking finding at the time, realizing that this is achieved without prior detection of the discontinuity locations. Unfortunately, in higher dimensions the wavelet transform loses its optimality; the multi-dimensional transform is a simple separable extension of the 1-D transform, with atoms supported over rectangular regions of different sizes (see Fig. 3.3). This separability makes the transform simple to apply, however the resulting dictionary is only effective for signals with *point* singularities, while most natural signals exhibit elongated *edge* singularities. The JPEG2000 image compression standard, based on the wavelet transform, is

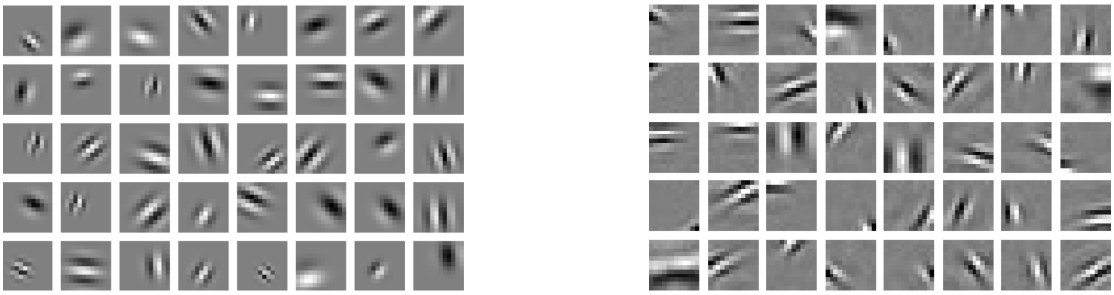


Figure 3.2: Left: a few 12×12 Gabor atoms at different scales and orientations. Right: a few atoms trained by Olshausen and Field (extracted from [94]).

indeed known for its *ringing* (smoothing) artifacts near edges.

Adaptivity: Going to the 1990's, the desire to push sparsity even further, and describe increasingly complex phenomena, was gradually revealing the limits of approximation in orthogonal bases. The weakness was mostly associated with the small and fixed number of atoms in the dictionary — dictated by the orthogonality — from which the optimal representation could be constructed. One option to obtain further sparsity was thus to adapt *the transform atoms themselves* to the signal content.

One of the first such structures to be proposed was the *wavelet packet* transform, introduced by Coifman, Meyer and Wickerhauser in 1992 [132]. The transform is built upon the success of the wavelet transform, adding adaptivity to allow finer tuning to the specific signal properties. The main observation of Coifman *et al.* was that the wavelet transform enforced a very specific time-frequency structure, with high frequency atoms having small supports and low frequency atoms having large supports. Indeed, this choice has deep connections to the behavior of real natural signals; however, for specific signals, better partitionings may be possible. The wavelet packet dictionary essentially unifies all dyadic time-frequency atoms which can be derived from a specific pair of scaling function and mother wavelet, so atoms of different frequencies can come in an array of time supports. Out of this large collection, the wavelet packet transform allows to efficiently select an optimized *orthogonal* sub-dictionary for any given signal, with the standard wavelet basis being just one of an exponential number of options. The process was thus named by the authors a *Best Basis* search. The wavelet packet transform is, by definition, at least as good as wavelets in terms of coding efficiency. However, we note that the multi-dimensional wavelet packet transform remains a separable and non-oriented transform, and thus does not generally provide a substantial improvement over wavelets for images.

Geometric Invariance and Overcompleteness: In 1992, Simoncelli *et al.* [14] published a thorough work advocating a dictionary property they termed *shiftability*, which describes the invariance of the dictionary under certain geometric deformations, e.g., translation, rotation or scaling. Indeed, a well known weakness of the wavelet transform is its strong translation-sensitivity, as well as rotation-sensitivity in higher dimensions. The authors concluded that achieving these properties required abandoning orthogonality in favor of *overcompleteness*, since the critical number of atoms in an orthogonal transform was simply insufficient. In the same work, the authors developed an overcomplete *oriented* wavelet transform — the *steerable wavelet transform* — which was based on their previous work on steerable filters and consisted of localized 2-D wavelet atoms in many orientations, translations and scales.

For the basic 1-D wavelet transform, translation-invariance can be achieved by increasing the sampling density of the atoms. The *stationary wavelet transform*, also known as the undecimated or non-subsampled wavelet transform, is obtained from the orthogonal transform by eliminating the sub-sampling and collecting *all* translations of the atoms over the signal domain. The algorithmic foundation for this was laid by Beylkin in 1992 [133], with the development of an efficient algorithm for computing the undecimated transform. The stationary wavelet transform was indeed found to substantially improve signal recovery compared to orthogonal wavelets, and its benefits were independently demonstrated

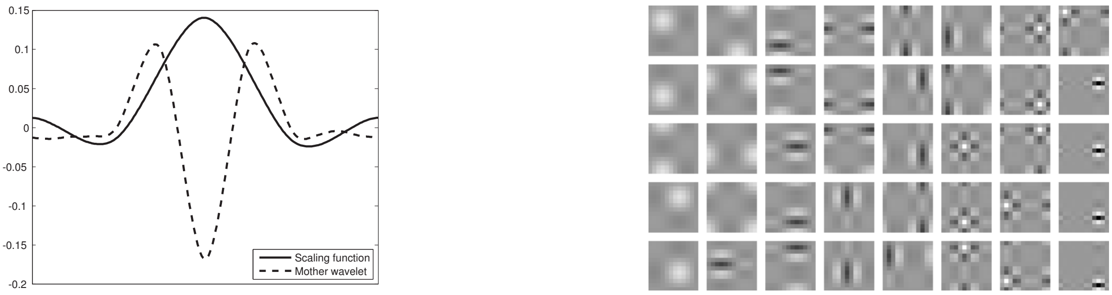


Figure 3.3: Left: Coiflet 1-D scaling function (solid) and mother wavelet (dashed). Right: some 2-D separable Coiflet atoms.

in 1995 by Nason and Silverman [134] and Coifman and Donoho [135].

3.2.3 From Transforms to Dictionaries

By the second half of the 1990's, most of the concepts for designing effective transforms were laid out. At the same time, a conceptual change of a different sort was gradually taking place. In their seminal work from 1993, Mallat and Zhang [11] proposed a novel sparse signal expansion scheme based on the selection of a small subset of functions from a general overcomplete *dictionary* of functions. Shortly after, Chen, Donoho and Saunders published their influential paper on the *Basis Pursuit* [136], and the two works signalled the beginning of a fundamental move from *transforms* to *dictionaries* for sparse signal representation. An array of works since has formed a wide mathematical and algorithmic foundation of this new field, and established it as a central tool in modern signal processing [137].

The seemingly minor terminological change enclosed the idea that a signal was allowed to have *more than one description* in the representation domain, and that selecting the best one depended on the task. Moreover, it de-coupled the processes of *designing* the dictionary and *coding* the signal: indeed, given the dictionary — the collection of elemental signals — different cost functions could be proposed in (3.3), and different coding methods could be applied.

The first dictionaries to be used in this way were the existing transforms —

such as the Fourier, wavelet, STFT, and Gabor transforms, see e.g., [11, 136]. As an immediate consequence, the move to a dictionary-based formalism provided the benefit of constructing *dictionary mergers*, which are the unions of several simpler dictionaries; these were proposed by Chen, Donoho and Saunders in [136], and provide a simple way to increase the variety of features representable by the dictionary.

3.2.4 Higher Dimensional Signals

The variety of dictionaries developed through the mid-1990's served one-dimensional signals relatively well. However, the dictionaries for multi-dimensional signal representation were still unsatisfying. Particularly frustrating, for instance, was the common knowledge that 2-D piecewise-smooth signals could be described much more efficiently using a simple piecewise-linear approximation over an adaptive triangle grid, than using any existing dictionary [7, 16].

In 1998, Donoho developed the *wedgelet* dictionary for 2-D signal representation [138], which bears some resemblance to the adaptive triangulation structure. The wedgelet dictionary consists of constant-valued, axis-aligned squares, bisected by straight lines, and spanning many sizes and locations. Donoho showed that this dictionary is optimal for piecewise-constant images with regular edge discontinuities, and provided a quick (though non-optimal) approximation technique. The elegant wedgelet construction, though too simplistic for many tasks, was adopted and generalized by several researchers, leading to such structures as wavelet-wedgelets hybrids (*wedgeprints*) [139], piecewise-linear wedgelets (*platelets*) [140], and higher-dimensional wedgelets (*surflets*) [141].

In parallel to the wedgelet transform, Candès and Donoho introduced the *ridgelet* transform as a multi-dimensional extension of the wavelet transform [92]. A ridgelet atom is a translated and dilated wavelet in one direction, and fixed in the orthogonal directions (similar to a plane wave). The transform is proven to be optimal for piecewise-smooth functions with plane discontinuities. Indeed, the basic ridgelet dictionary is unsuitable for natural signals due its lack of localization. However, with proper localization and multi-scale extension, the dictionary forms the core of the much more powerful *curvelet* transform [16, 77], introduced by the authors soon after, and which provides a comprehensive framework for representing multi-dimensional signals. Similar efforts led to the development of the *contourlet*, *shearlet*, and other transforms, which are described in more detail in the next section.

3.2.5 Analytic versus Trained Dictionaries

The dictionaries described so far all roughly fall under the umbrella of *Harmonic Analysis*, which suggests modeling interesting signal data by a more simple class of *mathematical functions*, and designing an efficient representation around this model. For example, the Fourier dictionary is designed around smooth functions, while the wavelet dictionary is designed around piecewise-smooth functions with point singularities. The dictionaries of this sort are characterized by an analytic formulation, and are usually supported by a set of optimality proofs and error rate bounds. An important advantage of this approach is that the resulting dictionary usually features a fast implicit implementation which does not involve multiplication by the dictionary matrix. On the other hand, the dictionary can only be *as successful as its underlying model*, and indeed, these models tend to be over-simplistic compared to the complexity of natural phenomena.

Through the 1980's and 1990's, *Machine Learning* techniques were rapidly gaining interest, and promised to confront this exact difficulty. The basic assumption behind the learning approach is that the structure of complex natural phenomena can be more accurately extracted *directly from the data* than by using a mathematical description. One direct benefit of this is that a finer adaptation to specific instances of the data becomes possible, replacing the use of generic models.

A key contribution to the area of dictionary learning was provided by Olshausen and Field in 1996 [94]. In their widely celebrated paper, the authors trained a dictionary for sparse representation of small image patches collected from a number of natural images. With relatively simple algorithmic machinery, the authors were able to show a remarkable result — the trained atoms they obtained were incredibly similar to the mammalian simple-cell receptive fields, which until then were only weakly explained via Gabor filters. The finding was highly motivating to the sparse representation community, as it demonstrated that the single assumption of sparsity could account for a fundamental biological visual behavior. Also, the results demonstrated the potential in example-based methods to uncover elementary structures in complex signal data.

The experiments of Olshausen and Field inspired a series of subsequent works aimed at improving the example-based training process. Towards the end of the 1990's, these works mostly focused on statistical training methods, which model the examples as random independent variables originating from a sparse noisy source. With $\mathbf{X} = [\mathbf{x}_1\,\mathbf{x}_2 \ldots \mathbf{x}_n]$ denoting the data matrix, the statistical approach suggests seeking for the dictionary which either maximizes the likelihood of the data $P(\mathbf{X}|\mathbf{D})$ (*Maximum Likelihood* estimation), e.g., [24], or maximizes the posterior probability of the dictionary $P(\mathbf{D}|\mathbf{X})$ (*Maximum A-Posterior* estimation), e.g., [25]. The resulting optimization problems in these works are typically solved in an Expectation-Maximization (EM) fashion, alternating estimation of the sparse representations and the dictionary; earlier works employed gradient descent or similar methods for both tasks, while later ones employ more powerful sparse-coding techniques for the estimation of the sparse representations.

3.3 Analytic Dictionaries — State-of-the-Art

Recent advances in analytic dictionary design have mostly focused on the move to two and higher dimensions. Multi-dimensional signals are significantly more complex than one-dimensional ones due to the addition of *orientation*. Also, the elementary singularities become *curves* — or *manifolds* in general — rather than points, and thus have a much more complex geometry to trace. In order to handle these complex signals, new transforms that are both localized and oriented have been developed.

Analytic dictionaries are typically formulated as *tight frames*, meaning that $\mathbf{D}\mathbf{D}^T\mathbf{x} = \mathbf{x}$ for all $\mathbf{x}$, and therefore the dictionary transpose can be used to obtain a representation over the dictionary. The analytic approach then proceeds by analyzing the behavior of the filter-set $\mathbf{D}^T\mathbf{x}$, and establishes decay rates and error bounds.

The tight frame approach has several advantages. Analyzing the behavior of $\mathbf{D}^T$ as an analysis operator seems easier than deriving sparsity bounds in a synthesis framework, and indeed, results obtained for the analysis formulation also induce upper bounds for the synthesis formulation. Another benefit is that — when formulated carefully — the algorithms for both analysis and synthesis operators become nearly reversals, simplifying algorithm design. Finally, the tight frame approach is beneficial in that it simultaneously produces a useful structure for both the analysis and synthesis frameworks, and has a meaningful interpretation in both.

Sparse-coding in this case is typically done by computing the analysis coefficients $\mathbf{D}^T\mathbf{x}$, and passing them through a non-linear shrinking operator. This method has the advantage of providing a simple and efficient way to achieve sparse representations over the dictionary, though it is worth noting that from a pure synthesis point of view, this process is sub-optimal, and one might benefit from employing a more advanced sparse-coding technique, e.g., an *iterated shrink-*

age technique [43], directly to the expansion coefficients. Recent efforts in this direction have led Yaghoobi *et al.* [142] to propose a parameter tuning method for analytic dictionaries, which may further improve their performance in sparse-coding processes.

3.3.1 Curvelets

The curvelet transform was introduced by Candès and Donoho in 1999 [16], and was later refined into its present form in 2003 [17]. When published, the transform astonished the harmonic analysis community by achieving what was then believed to be only possible with adaptive representations: it could represent 2-D piecewise-smooth functions with smooth curve discontinuities at an (essentially) optimal rate.

The curvelet transform is formulated as a *continuous* transform, with discretized versions developed for both formulations [17, 77, 143]. Each curvelet atom is associated with a specific location, orientation and scale. In the 2-D case, a curvelet atom is roughly supported over an elongated elliptical region, and is oscillatory along its width and smooth along its length, see Fig. 3.4. The curvelet atoms are characterized by their specific anisotropic support, which obeys a parabolic scaling law $width \sim length^2$. As it turns out, this property is useful for the efficient representation of smooth curves [144], and indeed several subsequent transforms follow this path. In higher dimensions, the curvelet atoms become flattened ellipsoids, oscillatory along their short direction and smooth along the other directions [17, 143, 145].

3.3.2 Contourlets

The curvelet transform offers an impressively solid continuous construction and exhibits several useful mathematical properties. However, its discretization turns out to be challenging, and the resulting algorithms are relatively complicated.

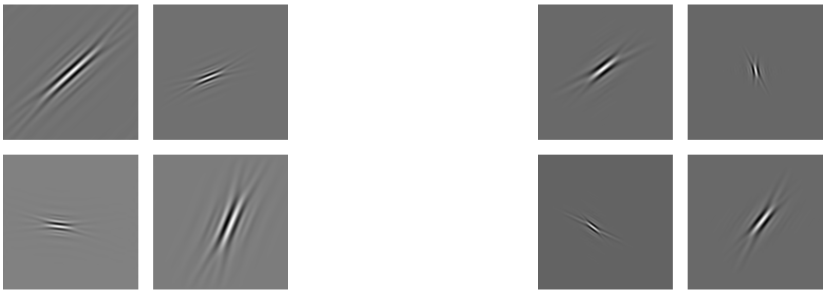


Figure 3.4: Some curvelet atoms (left) and contourlet atoms (right). Both represent the second version of the corresponding transform.

Also, current discretizations have relatively high redundancies, which makes them more costly to use and less applicable for tasks like compression.

With this in mind, Do and Vetterli proposed the *contourlet* transform in 2002 [18, 146] as an alternative to the 2-D curvelet transform. The transform was later refined in 2006 by Lu and Do [19], and a multi-dimensional version, named *surfacelets*, was also recently introduced [20].

The contourlet transform shares many of the characteristics of the curvelet transform, including localization, orientation, and parabolic scaling. However, as opposed to curvelets, the contourlets are defined *directly in the discrete domain*, and thus have a native and simple construction for discrete signals. Also, the standard contourlet transform has much lower redundancy, approximately in the range $[1.3, 2.3]$ for the second-generation implementation [19], compared to $[2.8, 7.2]$ for second-generation curvelets [17].

The contourlet transform implementation is based on a pyramidal band-pass decomposition of the image followed by a directional filtering stage. The resulting oriented atoms are elongated and oscillatory along their width, with some visual resemblance to the curvelet atoms (see Fig. 3.4). The main appeal of the transform is due to its simple discrete formulation, its low complexity and reduced redundancy. It should be noted, though, that while the transform is well suited for tasks such as compression, its aggressive sub-sampling has been noted to lead to artifacts in signal reconstruction, in which case a translation-invariant version of the transform is preferred [79, 147]; indeed, this option significantly increases redundancy and complexity, though the simpler structure of the transform remains.

3.3.3 Bandelets

The bandelet transform was proposed in 2005 by Le Pennec and Mallat [148], with a second version introduced soon after by Peyré and Mallat [149]. The bandelet transform represents one of the most recent contributions in the area of *signal-adaptive transforms*, and as such it differs fundamentally from the non-adaptive curvelet and contourlet transforms.

The idea behind the bandelet construction is to exploit geometric regularity in the image — specifically edges and directional phenomena — in order to fit a specifically optimized set of atoms to the image. The original bandelet construction operates in the spatial domain, and is based on an adaptive subdivision of the image to dyadic regions according to the local complexity; in each region, a set of skewed wavelets is matched to the image flow, in such a way that the wavelet atoms essentially ”wrap-around” the edges rather than cross them. This process significantly reduces the number of large wavelet coefficients, as these typically emerge from the interaction of a wavelet atom and a discontinuity.

The resulting set of atoms forms a (slightly) overcomplete set, which is specifically tailored for representing the given image. In the second bandelet construction, which is formulated in the wavelet domain, the transform is further refined to produce an *orthogonal* set. In terms of dictionaries, the bandelet transform selects a set of atoms from a nearly infinite set, and in fact discretization is the main source for limiting the size of this set. This is as opposed to the wavelet packet transform, for instance, where the complete set of atoms is not much larger than the signal dimension. See Fig. 3.5 for an example of bandelets.

3.3.4 Other Analytic Dictionaries

Many additional analytic transforms have been developed during the past decade, some of which we mention briefly. The *complex wavelet transform* [15, 150] is an oriented and near-translation-invariant high-dimensional extension of the wavelet transform, achieved through the utilization of *two* mother wavelets satisfying a specific relationship between them. Similar to the original wavelet transform, the complex wavelet transform is efficient and simple to implement, and the added phase information delivers orientation sensitivity and other favorable properties. The *shearlet* transform [21, 80, 151] is a recently proposed alternative to curvelets, which utilizes structured shear operations rather than rotations to control orientation. Similar to curvelets, the shearlet transform is based on a comprehensive continuous mathematical construction, and it shares many of the properties of the curvelet transform while providing some attractive new features. See Fig. 3.6 for some examples of complex wavelet and shearlet atoms.

Recent adaptive dictionaries include the *directionlet* transform [152], which is a discrete transform which constructs oriented and anisotropic wavelets based on local image directionality, utilizing a specialized directional grouping of the grid points for its numerical implementation. The *grouplet* transform [153] is a multi-scale adaptive transform which essentially generalizes Haar wavelets to arbitrary supports, based on image content regularity; when applied in the wavelet domain, the transform bears some resemblance to the second-generation bandelet transform, and thus is referred to as *grouped bandelets*.

3.4 Dictionary Training — State-of-the-Art

Dictionary training is a much more recent approach to dictionary design, and as such, has been strongly influenced by the latest advances in sparse representation theory and algorithms. The most recent training methods focus on ℓ^0 and ℓ^1

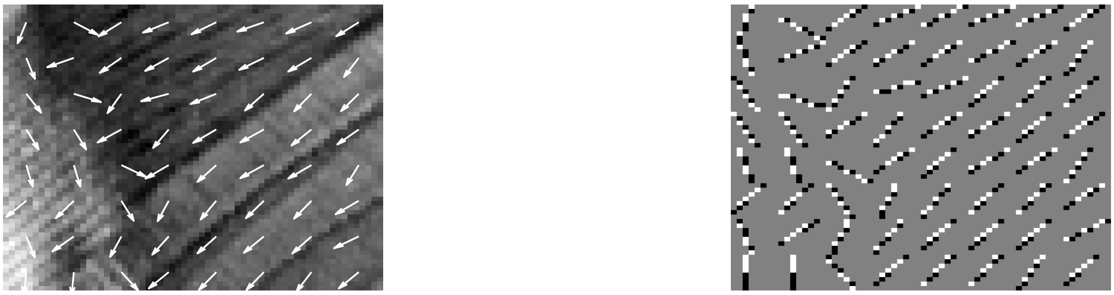


Figure 3.5: Left: the flow in a specific image region. Right: some bandelet atoms adapted to the region. Note how the 1-D wavelets are skewed to follow edges.

sparsity measures, which lead to simple formulations and enable the use of recently developed efficient sparse-coding techniques [38, 39, 41, 43, 46, 136].

The main advantage of trained dictionaries is that they lead to state-of-the-art results in many practical signal processing applications. The cost — as in the case of the KLT — is a dictionary with no known inner structure or fast implementation. Thus, the most recent contributions to the field employ *parametric* models in the training process, which produce structured dictionaries, and offer several advantages. A different development, which we do not discuss here, is the recent advancement in *online* dictionary learning [34, 154], which allows training dictionaries from very large sets of examples, and is found to accelerate convergence and improve the trained result.

3.4.1 Method of Optimal Directions

The Method of Optimal Directions (MOD) was introduced by Engan *et al.* in 1999 [23, 155], and was one of the first methods to implement what is known today as a *sparsification process*. Given a set of examples $\mathbf{X} = [\mathbf{x}_1 \; \mathbf{x}_2 \; \ldots \; \mathbf{x}_n]$, the goal of the MOD is to find a dictionary $\mathbf{D}$ and a sparse matrix $\mathbf{\Gamma}$ which minimize the representation error,

$$\operatorname*{Argmin}_{\mathbf{D},\mathbf{\Gamma}} \|\mathbf{X} - \mathbf{D}\mathbf{\Gamma}\|_F^2 \quad \text{Subject To} \quad \|\boldsymbol{\gamma}_i\|_0 \le T \quad \forall i \;, \tag{3.6}$$

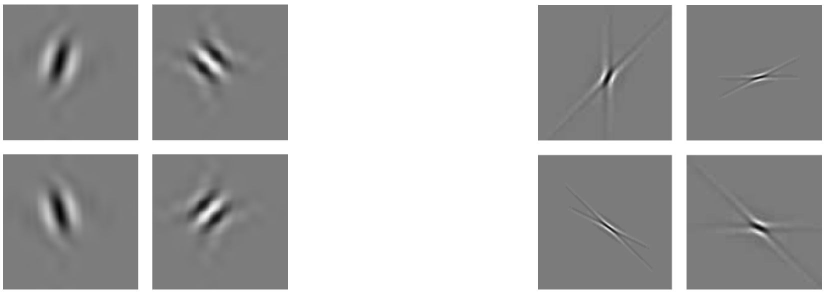


Figure 3.6: Left: a few complex wavelet atoms (real part). Right: a few shearlets.

where $\{\boldsymbol{\gamma}_i\}$ represent the columns of $\boldsymbol{\Gamma}$, and the ℓ^0 sparsity measure $\|\cdot\|_0$ counts the number of non-zeros in the representation. The resulting optimization problem is combinatorial and highly non-convex, and thus we can only hope for a local minimum at best. Similar to other training methods, the MOD alternates sparse-coding and dictionary update steps. The sparse-coding is performed for each signal individually using any standard technique. For the dictionary update, (3.6) is solved via the analytic solution of the quadratic problem, given by $\mathbf{D} = \mathbf{X}\boldsymbol{\Gamma}^+$ with $\boldsymbol{\Gamma}^+$ denoting the Moore-Penrose pseudo-inverse.

The MOD typically requires only a few iterations to converge, and is overall a very effective method. The method suffers, though, from the relatively high complexity of the matrix inversion. Several subsequent works have thus focused on reducing this complexity, leading to more efficient methods.

3.4.2 Union of Orthobases

Training a union-of-orthobases dictionary was proposed in 2005 by Lesage *et al.* [27] as a means of designing a dictionary with reduced complexity and which could be more efficiently trained. The process also represents one of the first attempts at training a *structured* overcomplete dictionary — a tight frame in this case. The model suggests training a dictionary which is the concatenation of k orthogonal bases, so $\mathbf{D} = [\mathbf{D}_1\,\mathbf{D}_2 \ldots \mathbf{D}_k]$ with the $\{\mathbf{D}_i\}$ unitary matrices. Sparse-coding over this dictionary can be performed efficiently through a Block

Coordinate Relaxation (BCR) technique [47].

A drawback of this approach is that the proposed model itself is relatively restrictive, and in practice it does not perform as well as more flexible structures. Interestingly, there is a close connection between this structure and the more powerful *Generalized PCA* model, described next. The GPCA also arises from a union of orthogonal spaces model, though it deviates from the classical sparse representation paradigm. Identifying such relations could thus prove valuable in enabling a merge between the two forces.

3.4.3 Generalized PCA

Generalized PCA, introduced in 2005 by Vidal, Ma and Sastry [82], offers a different and very interesting approach to overcomplete dictionary design. The GPCA view is basically an extension of the original PCA formulation, which approximates a set of examples by a low-dimensional subspace. In the GPCA setting, the set of examples is modeled as the union of *several* low-dimensional subspaces — perhaps of unknown number and variable dimensionality — and the algebraic-geometric GPCA algorithm determines these subspaces and fits orthogonal bases to them.

The GPCA viewpoint differs from the sparsity model described in (3.2), as each example in the GPCA setting is represented using only one of the subspaces; thus, atoms from different subspaces cannot jointly represent a signal. This property has the advantage of limiting over-expressiveness of the dictionary, which characterizes other overcomplete dictionaries; on the other hand, the dictionary structure may be too restrictive for more complex natural signals.

A unique property of the GPCA is that as opposed to other training methods, it can detect the *number* of atoms in the dictionary in certain settings. Unfortunately, the algorithm may become very costly this way, especially when the amount and dimension of the subspaces increases. Indeed, intriguing models arise by merging the GPCA viewpoint with the classical sparse representation viewpoint: for instance, one could easily envision a model generalizing (3.6) where several distinct dictionaries are allowed to co-exists, and every signal is assumed to be sparse over exactly one of these dictionaries.

3.4.4 The K-SVD Algorithm

The desire to efficiently train a generic dictionary for sparse signal representation led Aharon, Elad and Bruckstein to develop the K-SVD algorithm in 2005 [28]. The algorithm aims at the same sparsification problem as the MOD (3.6), and employs a similar block-relaxation approach. The main contribution of the K-SVD is that the dictionary update, rather than using a matrix inversion, is performed atom-by-atom in a simple and efficient process. Further acceleration is provided by updating both the current atom and its associated sparse coefficients simultaneously. The result is a fast and efficient algorithm which is less demanding than the MOD.

The K-SVD algorithm takes its name from the Singular-Value-Decomposition (SVD) process that forms the core of the atom update step, and which is repeated K times, as the number of atoms. For a given atom k, the quadratic term in (3.6) is rewritten as

$$\|\mathbf{X} - \sum_{j \neq k} \mathbf{d}_j \boldsymbol{\gamma}_j^T - \mathbf{d}_k \boldsymbol{\gamma}_k^T\|_F^2 = \|\mathbf{E}_k - \mathbf{d}_k \boldsymbol{\gamma}_k^T\|_F^2 \ , \tag{3.7}$$

where $\{\boldsymbol{\gamma}_j^T\}$ are the *rows* of $\mathbf{\Gamma}$, and $\mathbf{E}_k$ is the residual matrix. The atom update is obtained by minimizing (3.7) for $\mathbf{d}_k$ and $\boldsymbol{\gamma}_k^T$ via a simple rank-1 approximation of $\mathbf{E}_k$. To avoid introduction of new non-zeros in $\mathbf{\Gamma}$, the update process is performed using only the examples whose current representations use the atom $\mathbf{d}_k$. Fig. 3.7 shows an example of a K-SVD trained dictionary for 2-D image patch representation.

In practice, the K-SVD is an effective method for representing small signal patches. However, the K-SVD, as well as the MOD, suffer from a few common weaknesses. The high non-convexity of the problem means that the two methods will get caught in local minima or even saddle points. Also, the result of the training is a non-structured dictionary which is relatively costly to apply, and therefore these methods are suitable for signals of relatively small size. In turn, in recent years several *parametric* dictionary training methods have begun to appear, and aim to address these issues by importing the strengths of analytic dictionaries to the world of example-based methods.

3.4.5 Parametric Training Methods

There are several motivations for training a parametric dictionary. By reducing the number of free parameters and imposing various desirable properties on the dictionary, we can accelerate convergence, reduce the density of local minima, and assist in converging to a better solution. A smaller number of parameters also improves generalization of the learning process and reduces the number of examples needed. Another advantage of the parameterization is that the dictionary will typically have a more compact representation, and may lend itself to a more efficient implementation. Finally, with the proper structure, a parameterized dictionary may be designed to represent infinite or arbitrary-sized signals. Several parametric dictionary structures have been recently proposed, and in the following we mention a few examples.

Translation-Invariant Dictionaries: Given a dictionary for a fixed-size signal patch, a dictionary for an arbitrary-sized signal can be constructed by collecting all the translations of the trained atoms over the signal domain and forming a large translation-invariant dictionary. Several training methods for such structures have been proposed in recent years. Blumensath and Davies [156] employed statistical training methodology to design dictionaries for arbitrary time series representation; Jost *et al.* [157] developed a learning process based on a sequential computation of the dictionary atoms, promoting de-correlation of the trained atoms; and the MOD has been extended by Engan *et al.* [29] to translation-invariant and op-

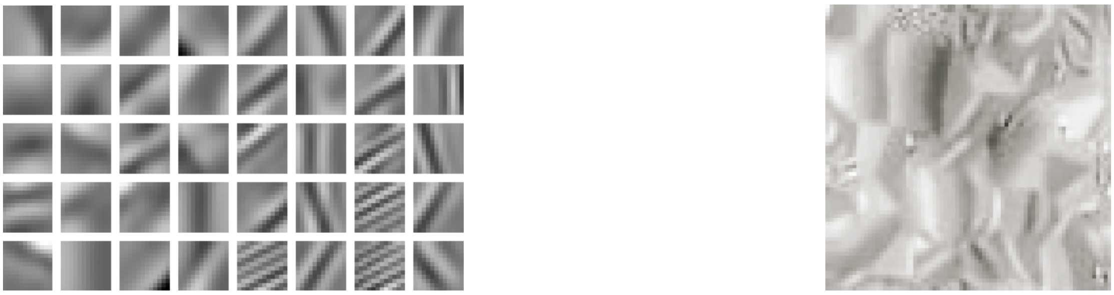


Figure 3.7: Left: atoms from a K-SVD dictionary trained on 12×12 image patches from *Lena*. Right: a signature dictionary, trained on the same image.

tionally linearly-constrained dictionary training, which they successfully applied to electrocardiogram (ECG) recordings.

A very different approach to translation-invariance was recently proposed by Aharon and Elad in [32]. In the 2-D case, their proposed *signature dictionary* is a small image in which each $N \times N$ sub-block constitutes an atom. Thus, assuming a periodic extension, an $M \times M$ signature dictionary stores M^2 atoms in a compact structure. Compared to the previous methods, this approach does not aim to produce a dictionary for arbitrary-sized signals, and instead, describes an interesting form of invariance at the block level. Indeed, a possible extension of this model could allow extraction of variable-sized atoms from the signature image, though this option remains for future research. An example of a trained signature dictionary is shown in Fig. 3.7.

Multiscale Dictionaries: Training dictionaries with multi-scale structures is an exciting and challenging option which has only been partially explored. In [26], Sallee and Olshausen proposed a pyramidal wavelet-like signal expansion, generated from the dilations and translations of a set of elementary small trained patches. The training method learns the elementary patches as well as a statistical model of the coefficients. In simulations, the structure is found to compete favorably with other pyramidal-based transforms. While the results of this method seem slightly constrained by the small number of elementary functions trained, it is likely to substantially benefit from increasing the overcompleteness and employing some more advanced sparse-coding machinery.

A different and interesting contribution in this direction is the semi-multiscale extension of the K-SVD introduced in 2008 by Mairal, Sapiro and Elad [31]. The semi-multiscale structure is obtained by arranging several fixed-sized learned dictionaries of different scales over a dyadic grid. The resulting structure is found to deliver a pronounced improvement over the single-scale K-SVD dictionary in applications such as denoising and inpainting, producing nearly state-of-the-art denoising performance. The main significance of this work, though, is the potential it demonstrates in going to multi-scale learned structures. Such results are highly encouraging, and motivate further research into multi-scale training models.

Sparse Dictionaries: One of the most recent contributions to the field of parametric dictionaries, specifically aimed at merging the advantages of trained and analytic dictionaries, was recently presented by Rubinstein, Zibulevsky and Elad [33]. Their proposed *sparse dictionary* structure takes the form $\mathbf{D} = \mathbf{BA}$, where $\mathbf{B}$ is some fixed analytic dictionary with a fast computation, and $\mathbf{A}$ is a sparse matrix. Thus, the dictionary is compactly expressed and has a fast implementation, while adaptivity is provided through the matrix $\mathbf{A}$. Also, the parameterization is shown to improve learning generalization and to reduce the training set size. Thus, the training method can be used to learn larger dictionaries than the MOD or K-SVD, e.g., for large image patches, or 3-D signal patches. Nonetheless, we note that the sparse dictionary structure, as most other models, remains targeted at fixed-size signals. Indeed, further work is required to design more general dictionary models which will truly capture the benefits of both analytic and example-based worlds.

3.5 Conclusions

Dictionary design has significantly evolved over the past decades, beginning with simple orthogonal transforms and leading to the complex overcomplete analytic and trained dictionaries now defining the state-of-the-art. Substantial conceptual advancement has been made in understanding the elements of an efficient dictionary design — most notably adaptivity, multi-scale, geometric invariance, and overcompleteness. However, with a wealth of tools already developed, much work remains to be done; indeed, the various components have yet to be neatly merged into a single efficient construct. Many future research directions have been mentioned in the text, and demonstrate the viability and vividness of the field as well as the large number of challenges that still await. Of specific interest, we highlight the strong need for a multi-scale structured dictionary learning paradigm, as well as methods to use such dictionaries in applications, which will clearly be the focus of much research in the near future.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable and enlightening comments, which substantially enhanced the final result.

Images in this paper were generated using several software packages. The authors would like to acknowledge the writers of these packages and thank them for their contribution and support of reproducible research. *In order of appearance*: Images of the curvelet transform were generated using the CurveLab toolbox curtesy of Candès, Demanet, Donoho and Ying (`http://www.curvelet.org`); images of the contourlet transform were generated using the SurfBox toolbox curtesy of Y. M. Lu (`http://lcav.epfl.ch/~lu`); images related to the bandelet transform were generated using the Bandelet Toolbox curtesy of G. Peyré (`http://www.cmap.polytechnique.fr/~peyre/bandelets`); images of the complex wavelet transform were generated using the wavelet software curtesy of Cai, Li and Selesnick (`http://taco.poly.edu/WaveletSoftware`). The authors would further like to thank G. Kutyniok and W. Lim for their assistance in obtaining images of the shearlet transform.

Chapter 4

Learning Sparse Dictionaries for Sparse Signal Approximation

Published in IEEE Transactions on Signal Processing, 58(3):1553–1564, 2010.

Coauthored with Michael Zibulevsky and Michael Elad.

Abstract

An efficient and flexible dictionary structure is proposed for sparse and redundant signal representation. The proposed *sparse dictionary* is based on a sparsity model of the dictionary atoms over a base dictionary, and takes the form $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$ where $\mathbf{\Phi}$ is a fixed base dictionary and $\mathbf{A}$ is sparse. The sparse dictionary provides efficient forward and adjoint operators, has a compact representation, and can be effectively trained from given example data. In this, the sparse structure bridges the gap between implicit dictionaries, which have efficient implementations yet lack adaptability, and explicit dictionaries, which are fully adaptable but non-efficient and costly to deploy. In this paper we discuss the advantages of sparse dictionaries, and present an efficient algorithm for training them. We demonstrate the advantages of the proposed structure for 3-D image denoising.

4.1 Introduction

Sparse representation of signals over redundant dictionaries [11, 12, 137] is a rapidly evolving field, with state-of-the-art results in many fundamental signal and image processing tasks [2, 31, 65, 66, 70, 74, 84, 158]. The basic model suggests that natural signals can be compactly expressed, or efficiently approximated, as a linear combination of prespecified *atom signals*, where the linear coefficients are *sparse* (i.e., most of them zero). Formally, letting $\mathbf{x} \in \mathbb{R}^N$ be a column signal, and arranging the atom signals as the columns of the *dictionary* $\mathbf{D} \in \mathbb{R}^{N \times L}$, the sparsity assumption is described by the following *sparse approximation* problem, for which we assume a sparse solution exists:

$$\hat{\boldsymbol{\gamma}} = \operatorname*{Argmin}_{\boldsymbol{\gamma}} \|\boldsymbol{\gamma}\|_0^0 \quad \text{Subject To} \quad \|\mathbf{x} - \mathbf{D}\boldsymbol{\gamma}\|_2 \leq \epsilon \ . \tag{4.1}$$

In this expression, $\hat{\boldsymbol{\gamma}}$ is the *sparse representation* of $\mathbf{x}$, ϵ is the error tolerance, and the function $\|\cdot\|_0^0$, loosely referred to as the *ℓ^0-norm*, counts the non-zero entries of a vector. Though known to be NP-hard in general [37], the above problem is relatively easy to *approximate* using a wide variety of techniques [38–40, 42, 44–48].

A fundamental consideration in employing the above model is the *choice* of the dictionary $\mathbf{D}$. The majority of literature on this topic can be categorized into two basic approaches: the *analytic* approach and the *learning-based* approach. In the first approach, a mathematical model of the data is formulated, and an analytic construction is developed to efficiently represent the model. This generally leads to dictionaries that are highly structured and have a fast numerical implementation. We refer to these as *implicit* dictionaries as they are described by their algorithm rather than their explicit matrix. Dictionaries of this type include Wavelets [7], Curvelets [16], Contourlets [18], Shearlets [21], Complex Wavelets [15], and Bandelets [148], among others.

The second approach suggests using machine learning techniques to infer the dictionary from a set of examples. In this case, the dictionary is typically represented as an explicit matrix, and a training algorithm is employed to adapt the matrix coefficients to the examples. Algorithms of this type include PCA and Generalized PCA [82], the Method of Optimal Directions (MOD) [23], the K-SVD [28], and others. Advantages of this approach are the much finer-tuned dictionaries they produce compared to the analytic approaches, and their significantly better performance in applications. However, this comes at the expense of generating an unstructured dictionary, which is more costly to apply. Also, complexity constraints limit the size of the dictionaries that can be trained in this way, and the dimensions of the signals that can be processed.

In this paper, we present a novel dictionary structure that bridges some of the gap between these two approaches, gaining the benefits of both. The structure is based on a sparsity model of the dictionary atoms over a known *base dictionary*. The new parametric structure leads to a simple and flexible dictionary representation which is both adaptive and efficient. Advantages of the new structure include low complexity, compact representation, stability under noise and reduced overfitting, among others.

4.1.1 Related Work

The idea of training dictionaries with a specific structure has been proposed in the past, though research in this direction is still in its early stages. Much of the work so far has focused specifically on developing adaptive *Wavelet* transforms, as in [159–162]. These works attempt to adapt various parameters of the Wavelet transform, such as the mother wavelet or the scale and dilation operators, to better suit specific given data.

More recently, an algorithm for training unions of orthonormal bases was proposed in [27]. The suggested dictionary structure takes the form

$$\mathbf{D} = [\ \mathbf{D}_1\ \mathbf{D}_2\ \ldots\ \mathbf{D}_k\ ]\ , \tag{4.2}$$

where the $\mathbf{D}_i$'s are unitary sub-dictionaries. The structure has the advantage of offering efficient sparse-coding via Block Coordinate Relaxation (BCR) [47], and its training algorithm is simple and relatively efficient. However, the dictionary model itself is relatively restrictive, and its training algorithm shows somewhat weak performance. Furthermore, the structure does not lead to quick forward and adjoint operators, as the dictionary itself remains explicit.

A different approach is proposed in [31], where a semi-multiscale structure is employed. The dictionary model is a concatenation of several scale-specific dictionaries over a dyadic grid, leading (in the 1-D case) to the form:

$$\mathbf{D} = \left(\begin{array}{c|cc|cccc|c} & \mathbf{D}_2 & & \mathbf{D}_3 & & & & \\ \mathbf{D}_1 & & & & \mathbf{D}_3 & & & \cdots \\ & & & & & \mathbf{D}_3 & & \\ & & \mathbf{D}_2 & & & & \mathbf{D}_3 & \end{array} \right). \tag{4.3}$$

The multiscale structure is shown to provide excellent results in applications such as denoising and inpainting. Nonetheless, the explicit nature of the dictionary is maintained along with most of the drawbacks of such dictionaries. Indeed, the use of sparse dictionaries to replace the explicit ones in (4.3) is an exciting option for future study.

Another recent contribution is the *signature dictionary* proposed in [32]. According to the suggested model, the dictionary is described via a compact *signature image*, with each sub-block of this image constituting an atom of the dictionary.[1] The advantages of this structure include near-translation-invariance, reduced overfitting, and faster sparse-coding when utilizing spatial relationships between neighboring signal blocks. On the other hand, the small number of parameters in this model — one coefficient per atom — also makes this dictionary more restrictive than other structures. Indeed, the sparse dictionary model proposed in this paper enhances the dictionary expressiveness by increasing the number of parameters

[1] Indeed, both fixed and variable-sized sub-blocks can be considered, though in [32] mostly fixed-sized blocks are studied.

per atom from 1 to $p > 1$, while maintaining other favorable properties of the dictionary.

4.1.2 Paper Organization

This paper is organized as follows. We begin in Section 4.2 with a description of the dictionary model and its advantages. In Section 4.3 we consider the task of training the dictionary from examples, and present an efficient algorithm for doing so. Section 4.4 analyzes and quantifies the complexity of sparse dictionaries, and compares it to other dictionary forms. Simulation results are provided in Section 4.5. We summarize and conclude in Section 4.6.

4.1.3 Notation

• Bold uppercase letters designate matrices ($\mathbf{M}$, $\mathbf{\Gamma}$), and bold lowercase letters designate column vectors ($\mathbf{v}$, $\boldsymbol{\gamma}$). The columns of a matrix are referenced using the corresponding lowercase letter, e.g. $\mathbf{M} = [\,\mathbf{m}_1 \,|\, \ldots \,|\, \mathbf{m}_n\,]$; the elements of a vector are similarly referenced using standard-type letters, e.g. $\mathbf{v} = (v_1, \ldots, v_n)^T$. The notation $\mathbf{0}$ is used to denote the zero vector, with its length inferred from the context.

• Given a single index $I = i_1$ or an ordered sequence of indices $I = (i_1, \ldots, i_k)$, we denote by $\mathbf{M}_I = [\,\mathbf{m}_{i_1} \,|\, \ldots \,|\, \mathbf{m}_{i_k}\,]$ the sub-matrix of $\mathbf{M}$ containing the columns indexed by I, *in the order in which they appear in* I. For vectors we similarly denote the sub-vector $\mathbf{v}_I = (v_{i_1}, \ldots, v_{i_k})^T$. We use the notation $\mathbf{M}_{I,J}$, with J a second index or sequence of indices, to refer to the sub-matrix of $\mathbf{M}$ containing the rows indexed by I and the columns indexed by J, in their respective orders. This notation is used for both access and assignment, so if $I = (2, 4, 6, \ldots, n)$, the statement $\mathbf{M}_{I,j} := \mathbf{0}$ means nullifying the even-indexed entries in the j-th row of $\mathbf{M}$.

4.2 Sparse Dictionaries

4.2.1 Motivation

Selecting a dictionary for sparse signal representation involves balancing between two elementary and seemingly competing considerations. The first is the *complexity* of the dictionary, as the dictionary forward and adjoint operators form the dominant components of most sparse-coding techniques, and these in turn form the core of all sparsity-based signal processing methods. Indeed, techniques such as Matching Pursuit (MP) [11], Orthogonal Matching Pursuit (OMP) [38], Stagewise Orthogonal Matching Pursuit (StOMP) [39], and their variants, all involve costly dictionary-signal computations each iteration. Other common methods such as interior-point Basis Pursuit [12] and FOCUSS [46] minimize a quadratic function each iteration, which is commonly performed using repeated application of the dictionary and its adjoint. Many additional methods rely heavily on the dictionary operators as well.

Over the years, a variety of dictionaries with fast implementations have been designed. For natural images, dictionaries such as Wavelets [7], Curvelets [16], Contourlets [18], and Shearlets [21], all provide fast transforms. However, such dictionaries are *fixed* and limited in their ability to adapt to different types of data. *Adaptability* is thus a second desirable property of a dictionary, and in practical applications, adaptive dictionaries consistently show better performance than generic ones [2, 31, 65, 74, 158]. Unfortunately, adaptive methods usually prefer explicit dictionary representations over structured ones, gaining a higher degree of freedom in the training but sacrificing regularity and efficiency of the result.[2]

[2] We should note that in *adaptive dictionaries* we are referring to dictionaries whose content can be adapted to different families of signals, typically through a learning process. *Signal-dependent* representation schemes, such as Best Wavelet Packet Bases [159] and Bandelets [148], are another type of adaptive process, but of a very different nature. These methods produce an optimized dictionary for a

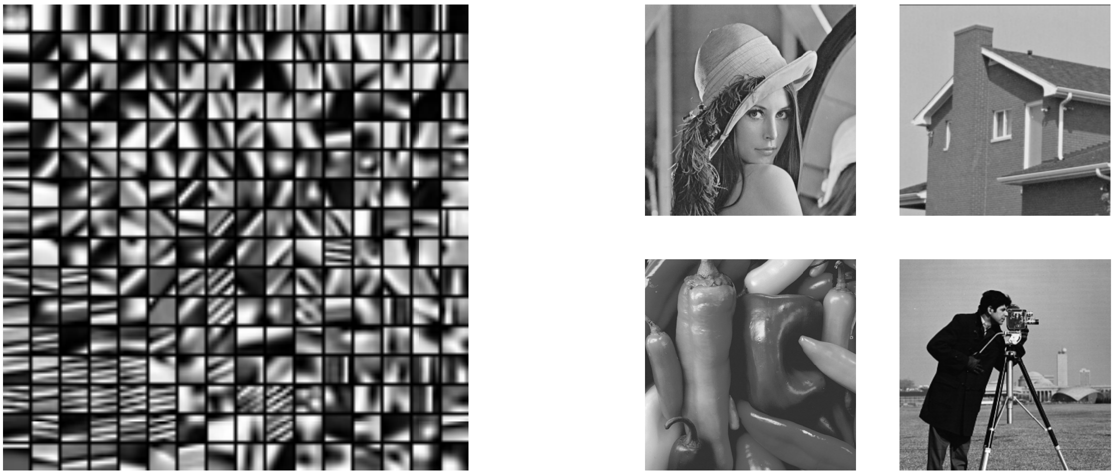


Figure 4.1: Left: dictionary for 8×8 image patches, trained using the K-SVD algorithm. Right: images used for the training. Each image contributed 25,000 randomly selected patches, for a total of 100,000 training signals.

Bridging this gap between complexity and adaptivity requires a parametric dictionary model that provides sufficient degrees of freedom. In this work, we propose the *sparse dictionary* model as a simple and effective structure for achieving this goal, based on sparsity of the atoms over a known *base dictionary*. Our approach can be motivated as follows. In Fig. 4.1 we see an example of a dictionary trained using the K-SVD algorithm [28] on a set of 8×8 natural image patches. The algorithm trains an explicit, fully un-constrained dictionary matrix, and yet, we see that the resulting dictionary is highly structured, with noticeably regular atoms. This gives rise to the hypothesis that the dictionary atoms *themselves* may have some underlying sparse structure over a more fundamental dictionary, and as we show in this paper, such a structure can indeed be recovered, and has several favorable properties.

given signal based on its specific characteristics (e.g. frequency content or geometry, respectively), and they are not considered here.

4.2.2 Dictionary Model

The sparse dictionary model suggests that each atom of the dictionary has *itself* a sparse representation over some *prespecified* base dictionary $\mathbf{\Phi}$. The dictionary is therefore expressed as

$$\mathbf{D} = \mathbf{\Phi}\mathbf{A} \ , \tag{4.4}$$

where $\mathbf{A}$ is the atom representation matrix, assumed to be sparse. For simplicity, we focus on matrices $\mathbf{A}$ having a fixed number of non-zeros per column, so $\|\mathbf{a}_i\|_0^0 \leq p$ for some p. The base dictionary $\mathbf{\Phi}$ will generally be chosen to have a quick implicit implementation, and, while $\mathbf{\Phi}$ may have any number of atoms, we assume it to span the signal space. The choice of the base dictionary obviously affects the success of the entire model, and we thus prefer one which already incorporates some prior knowledge about the data. Indeed, if more than one possible base dictionary exists, one may benefit from experimenting with a few different options in order to determine the most suitable one.

In comparison to implicit dictionaries, the dictionary model (4.4) provides *adaptability* via modification of the matrix $\mathbf{A}$, and can be efficiently trained from examples. Furthermore, as $\mathbf{\Phi}$ can be any dictionary — specifically, any existing implicit dictionary — the model can be viewed as an *extension* to existing dictionaries, adding them a new layer of adaptivity.

In comparison to explicit dictionaries, the sparse structure is significantly more efficient, depending mostly on the choice of $\mathbf{\Phi}$. It is also more compact to store and transmit. Furthermore, as we show later in this paper, the imposed structure acts as a regularizer in dictionary learning processes, and reduces overfitting and instability in the presence of noise. Training a sparse dictionary requires less examples than an explicit one, and produces useable results even when only a few examples are available.

The sparse dictionary model has another interesting interpretation. Assume the signal $\mathbf{x}$ is sparsely represented over the dictionary $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$, so $\mathbf{x} = \mathbf{\Phi}\mathbf{A}\boldsymbol{\gamma}$ for some sparse $\boldsymbol{\gamma}$. Therefore, $(\mathbf{A}\boldsymbol{\gamma})$ is the representation of $\mathbf{x}$ over $\boldsymbol{\Phi}$. Since both $\boldsymbol{\gamma}$ and the columns of $\mathbf{A}$ are sparse — having no more than, say, t and p non-zeros, respectively — this representation will have approximately tp non-zeros. However, such quadratic cardinality will generally fall beyond the success range of sparse-approximation techniques [137]. As such, it is no longer considered sparse in terms of the formulation (4.1), and sparse-coding methods will commonly fail to recover it. Furthermore, given a noisy version of $\mathbf{x}$, attempting to recover it directly over $\boldsymbol{\Phi}$ using tp atoms will likely result in capturing a significant portion of the noise along with the signal, due to the number of coefficients used.[3]

Through the sparse dictionary structure, we are able to accommodate denser signal representations over $\boldsymbol{\Phi}$ while essentially by-passing the related difficulties. The reason is that even though every t-sparse signal over $\mathbf{D}$ will generally have a denser tp-representation over $\boldsymbol{\Phi}$, not *every* tp-representation over $\boldsymbol{\Phi}$ will *necessarily* fit the model. The proposed model therefore acts as a *regularizer* for the allowed dense representations over $\boldsymbol{\Phi}$, and by learning the matrix $\mathbf{A}$, we are expressing in some form the complicated dependencies between its atoms.

4.3 Learning Sparse Dictionaries

We now turn to the question of *designing* a sparse dictionary for sparse signal representation. A straightforward approach would be to select some general (probably learned) dictionary $\mathbf{D}_0$, choose a base dictionary $\boldsymbol{\Phi}$, and sparse-code the atoms in $\mathbf{D}_0$ to obtain $\mathbf{D} = \boldsymbol{\Phi}\mathbf{A} \approx \mathbf{D}_0$. This naive approach, however, is clearly sub-optimal: specifically, the dictionary $\boldsymbol{\Phi}$ must be sufficiently compatible with $\mathbf{D}_0$, or else the representations in $\mathbf{A}$ may not be very sparse. Simulation results indicate that such dictionaries indeed perform poorly in practical signal processing applications.

[3] For white noise and a signal of length N, the expected remaining noise in a recovered signal using t atoms is approximately t/N the initial noise energy, due to the orthogonal projection.

A more desirable approach would be to learn the sparse dictionary using a process that is aware of the dictionary's specific structure. We adopt an approach which continues the line of work in [28], and develop a K-SVD-like learning scheme for training the sparse dictionary from examples. The algorithm is inspired by the Approximate K-SVD implementation presented in [163], which we briefly review.

4.3.1 K-SVD and Its Approximate Implementation

The K-SVD algorithm accepts an initial overcomplete dictionary matrix $\mathbf{D}_0 \in \mathbb{R}^{N\times L}$, a number of iterations k, and a set of examples arranged as the columns of the matrix $\mathbf{X} \in \mathbb{R}^{N\times R}$. The algorithm aims to iteratively improve the dictionary by approximating the solution to

$$\underset{\mathbf{D},\mathbf{\Gamma}}{\text{Min}}\ \|\mathbf{X}-\mathbf{D}\mathbf{\Gamma}\|_F^2 \quad \text{Subject To} \quad \begin{array}{l} \forall i \ \ \|\boldsymbol{\gamma}_i\|_0^0 \le t \\ \forall j \ \ \|\mathbf{d}_j\|_2 = 1 \end{array} . \tag{4.5}$$

Note that in this formulation, the atom normalization constraint is commonly added for convenience, though it does not have any practical significance to the result.

The K-SVD iteration consists of two basic steps: (i) sparse-coding the signals in $\mathbf{X}$ given the current dictionary estimate, and (ii) updating the dictionary atoms given the sparse representations in $\mathbf{\Gamma}$. The sparse-coding step can be implemented using any sparse-approximation method. The dictionary update is performed one atom at a time, optimizing the target function for each atom individually while keeping the remaining atoms fixed.

The atom update is carried out while preserving the sparsity constraints in (4.5). To achieve this, the update uses only those signals in $\mathbf{X}$ whose sparse representations use the current atom. Denoting by I the indices of the signals in $\mathbf{X}$ that use the j-th atom, the update of this atom is obtained by minimizing the target function

$$\|\mathbf{X}_I-\mathbf{D}\mathbf{\Gamma}_I\|_F^2 \tag{4.6}$$

for both the atom and its corresponding coefficient row in $\mathbf{\Gamma}_I$. The resulting problem is a simple rank-1 approximation, given by

$$\{\mathbf{d}, \mathbf{g}\} := \operatorname*{Argmin}_{\mathbf{d},\mathbf{g}} \|\mathbf{E} - \mathbf{d}\,\mathbf{g}^T\|_F^2 \quad \text{Subject To} \quad \|\mathbf{d}\|_2 = 1 \ , \tag{4.7}$$

where $\mathbf{E} = \mathbf{X}_I - \sum_{i \neq j} \mathbf{d}_i \mathbf{\Gamma}_{i,I}$ is the error matrix without the j-th atom, and $\mathbf{d}$ and $\mathbf{g}^T$ are the updated atom and coefficient row, respectively. The problem can be solved directly via an SVD decomposition, or more efficiently using some numerical power method.

In practice, the exact solution of (4.7) can be quite computationally demanding, especially when the number of training signals is large. As an alternative, an approximate solution may be used to reduce the complexity of this task [163]. The simplified update step is obtained by applying a single iteration of alternated-optimization [47, 164], given by

$$\begin{array}{rcl} \mathbf{d} & := & \mathbf{E}\mathbf{g}/\|\mathbf{E}\mathbf{g}\|_2 \\ \mathbf{g} & := & \mathbf{E}^T\mathbf{d} \end{array} \ . \tag{4.8}$$

The above process is known to ultimately converge to the optimum,[4] and when truncated, supplies an approximation which still reduces the penalty term. Also, this process eliminates the need to explicitly compute the matrix $\mathbf{E}$, as only its products with vectors are required.[5]

4.3.2 The Sparse K-SVD Algorithm

To train a sparse dictionary, we use the same basic framework as the original K-SVD algorithm. Specifically, we aim to (approximately) solve the optimization

[4] Applying two consecutive iterations of this process produces $\mathbf{d}^{j+1} = \mathbf{E}\mathbf{E}^T\mathbf{d}^j/\|\mathbf{E}\mathbf{E}^T\mathbf{d}^j\|_2$, which is the well-known power iteration for $\mathbf{E}\mathbf{E}^T$. The process converges, under reasonable assumptions, to the largest eigenvector of $\mathbf{E}\mathbf{E}^T$ — also the largest left singular vector of $\mathbf{E}$.

[5] Specifically, $\mathbf{E}\mathbf{g} = \mathbf{X}_I\mathbf{g} - \sum_{i \neq j} \mathbf{d}_i(\mathbf{\Gamma}_{i,I}\mathbf{g})$ can be computed via a series of vector inner products $\xi_i = \mathbf{\Gamma}_{i,I}\mathbf{g}$, followed by a vector sum $\sum_{i \neq j} \xi_i \mathbf{d}_i$ and a matrix-vector product $\mathbf{X}_I\mathbf{g}$. This is significantly faster and more memory-efficient than the explicit computation of $\mathbf{E}$, which involves matrix-matrix operations. The same applies to the computation of $\mathbf{E}^T\mathbf{d}$.

problem

$$\underset{\mathbf{A},\boldsymbol{\Gamma}}{\text{Min}} \ \|\mathbf{X} - \boldsymbol{\Phi}\mathbf{A}\boldsymbol{\Gamma}\|_F^2 \qquad\qquad \text{Subject To} \begin{cases} \forall i \ \ \|\boldsymbol{\gamma}_i\|_0^0 \le t \\ \forall j \ \ \|\mathbf{a}_j\|_0^0 \le p \ , \ \ \|\Phi\mathbf{a}_j\|_2 = 1 \end{cases} , \tag{4.9}$$

alternating sparse-coding and dictionary update steps for a fixed number of iterations. The notable change is in the atom update step: as opposed to the original K-SVD algorithm, in this case the atom is constrained to the form $\mathbf{d} = \boldsymbol{\Phi}\mathbf{a}$ with $\|\mathbf{a}\|_0^0 \le p$. The modified atom update is therefore given by

$$\{\mathbf{a}, \mathbf{g}\} := \underset{\mathbf{a},\mathbf{g}}{\text{Argmin}} \ \|\mathbf{E} - \boldsymbol{\Phi}\mathbf{a}\,\mathbf{g}^T\|_F^2 \quad \text{Subject To} \quad \begin{array}{r} \|\mathbf{a}\|_0^0 \le p \\ \|\boldsymbol{\Phi}\mathbf{a}\|_2 = 1 \end{array} , \tag{4.10}$$

with $\mathbf{E}$ defined as in (4.7).

Interestingly, our problem is closely related to a different problem known as *Sparse Matrix Approximation* (here SMA), recently raised in the context of Kernel-SVM methods [165]. The SMA problem is formulated similar to problem (4.10), but replaces the rank-1 matrix $\mathbf{a}\mathbf{g}^T$ with a general matrix $\mathbf{T}$, and the sparsity constraint on $\mathbf{a}$ with a constraint on the number of non-zero rows in $\mathbf{T}$. Our problem is therefore essentially a *rank-constrained version* of the original SMA problem. In [165], the authors suggest a greedy OMP-like algorithm for solving the problem, utilizing randomization to deal with the large amount of work involved. Unfortunately, while this approach is likely extendable to the rank-constrained case, it leads to a computationally intensive process which is impractical for large problems.

Our approach therefore takes a different path to solving the problem, employing an alternated-optimization technique over $\mathbf{a}$ and $\mathbf{g}$ parallel to (4.8). We point out that as opposed to (4.8), the process here does *not* generally converge to the optimum when repeated, due to the non-convexity of the problem. Nonetheless, the method does guarantee a reduction in the target function value, which is essentially sufficient for our purposes.

To simplify the derivation, we note that (4.10) may be solved *without* the norm constraint on $\mathbf{\Phi a}$, and adding a post-processing step which transfers energy between $\mathbf{a}$ and $\mathbf{g}$ to achieve $\|\mathbf{\Phi a}\|_2 = 1$ while keeping $\mathbf{ag}^T$ fixed. The simplified problem is given by

$$\{\mathbf{a},\mathbf{g}\} := \underset{\mathbf{a},\mathbf{g}}{\mathrm{Argmin}}\ \|\mathbf{E} - \mathbf{\Phi a}\,\mathbf{g}^T\|_F^2 \quad \text{Subject To} \quad \|\mathbf{a}\|_0^0 \le p \ . \tag{4.11}$$

We also note that the solution to this problem is guaranteed to be non-zero for all $\mathbf{E} \neq 0$, hence the described re-normalization of $\mathbf{a}$ and $\mathbf{g}$ is possible.

Optimizing over $\mathbf{g}$ in (4.11) is straightforward, and given by

$$\mathbf{g} := \mathbf{E}^T\mathbf{\Phi a}/\|\mathbf{\Phi a}\|_2^2 \ . \tag{4.12}$$

Optimizing over $\mathbf{a}$, however, requires more attention. The minimization task for $\mathbf{a}$ is given by:

$$\mathbf{a} := \underset{\mathbf{a}}{\mathrm{Argmin}}\ \|\mathbf{E} - \mathbf{\Phi a}\,\mathbf{g}^T\|_F^2 \quad \text{Subject To} \quad \|\mathbf{a}\|_0^0 \le p \ . \tag{4.13}$$

The straightforward approach to this problem is to rewrite $\mathbf{E}$ as a column vector $\mathbf{e}$, and formulate the problem as an ordinary sparse-coding task for $\mathbf{e}$ (we use $\otimes$ to denote the Kronecker matrix product [166]):

$$\mathbf{a} := \underset{\mathbf{a}}{\mathrm{Argmin}}\ \|\mathbf{e} - (\mathbf{g} \otimes \mathbf{\Phi})\mathbf{a}\|_2^2 \quad \text{Subject To} \quad \|\mathbf{a}\|_0^0 \le p \ . \tag{4.14}$$

However, this leads to an intolerably large optimization problem, as the length of the signal to sparse-code is of the same order of magnitude as the entire dataset. Instead, we show that problem (4.13) is equivalent to a much simpler sparse-coding problem, namely

$$\mathbf{a} := \underset{\mathbf{a}}{\mathrm{Argmin}}\ \|\mathbf{Eg} - \mathbf{\Phi a}\|_2^2 \quad \text{Subject To} \quad \|\mathbf{a}\|_0^0 \le p \ . \tag{4.15}$$

Here, the vector $\mathbf{Eg}$ is of the same length as a single training example, and the dictionary is the base dictionary $\mathbf{\Phi}$ which is assumed to have an efficient implementation; therefore, this problem is significantly easier to handle than the previous one. Also, as discussed above, the vector **Eg** itself is much easier to compute than the vector **e**, which is just a vectorized version of the matrix **E**.

To establish the equivalence between the problems (4.13) and (4.15), we use the following Lemma:

Lemma 4.1. *Let $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ and $\boldsymbol{Y} \in \mathbb{R}^{N\times K}$ be two matrices, and $\boldsymbol{v} \in \mathbb{R}^{M}$ and $\boldsymbol{u} \in \mathbb{R}^{K}$ be two vectors. Also assume that $\boldsymbol{v}^T\boldsymbol{v} = 1$. Then the following holds:*

$$\|\boldsymbol{X} - \boldsymbol{Y}\boldsymbol{u}\boldsymbol{v}^T\|_F^2 = \|\boldsymbol{X}\boldsymbol{v} - \boldsymbol{Y}\boldsymbol{u}\|_2^2 + f(\boldsymbol{X}, \boldsymbol{v}) \ .$$

Proof. The equality follows from elementary properties of the trace function:

$$\begin{aligned}
&\|\mathbf{X} - \mathbf{Y}\mathbf{u}\mathbf{v}^T\|_F^2 = \\
&= Tr((\mathbf{X} - \mathbf{Y}\mathbf{u}\mathbf{v}^T)^T(\mathbf{X} - \mathbf{Y}\mathbf{u}\mathbf{v}^T)) \\
&= Tr(\mathbf{X}^T\mathbf{X}) - 2Tr(\mathbf{X}^T\mathbf{Y}\mathbf{u}\mathbf{v}^T) + Tr(\mathbf{v}\mathbf{u}^T\mathbf{Y}^T\mathbf{Y}\mathbf{u}\mathbf{v}^T) \\
&= Tr(\mathbf{X}^T\mathbf{X}) - 2Tr(\mathbf{v}^T\mathbf{X}^T\mathbf{Y}\mathbf{u}) + Tr(\mathbf{v}^T\mathbf{v}\mathbf{u}^T\mathbf{Y}^T\mathbf{Y}\mathbf{u}) \\
&= Tr(\mathbf{X}^T\mathbf{X}) - 2\mathbf{v}^T\mathbf{X}^T\mathbf{Y}\mathbf{u} + \mathbf{u}^T\mathbf{Y}^T\mathbf{Y}\mathbf{u} \\
&= Tr(\mathbf{X}^T\mathbf{X}) - 2\mathbf{v}^T\mathbf{X}^T\mathbf{Y}\mathbf{u} + \mathbf{u}^T\mathbf{Y}^T\mathbf{Y}\mathbf{u} + \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} - \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} \\
&= \|\mathbf{X}\mathbf{v} - \mathbf{Y}\mathbf{u}\|_2^2 + Tr(\mathbf{X}^T\mathbf{X}) - \mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v} \\
&= \|\mathbf{X}\mathbf{v} - \mathbf{Y}\mathbf{u}\|_2^2 + f(\mathbf{X}, \mathbf{v}) \ .
\end{aligned}$$

□

The Lemma implies that, assuming $\mathbf{g}^T\mathbf{g} = 1$, then for every representation vector **a**,

$$\|\mathbf{E} - \mathbf{\Phi}\mathbf{a}\mathbf{g}^T\|_F^2 = \|\mathbf{E}\mathbf{g} - \mathbf{\Phi}\mathbf{a}\|_2^2 + f(\mathbf{E}, \mathbf{g}) \ .$$

Clearly the important point in this equality is that the two sides differ by a constant independent of **a**. Thus, the target function in (4.13) can be safely replaced with the right hand side of the equality (sans the constant), establishing the equivalence to (4.15).

When using the Lemma to solve (4.13), we note that the energy assumption on $\mathbf{g}$ can be easily overcome, as dividing $\mathbf{g}$ by a non-zero constant simply results in a solution $\mathbf{a}$ scaled by that same constant. Thus (4.13) can be solved for any $\mathbf{g}$ by normalizing it to unit length, applying the Lemma, and re-scaling the solution $\mathbf{a}$ by the appropriate factor. Conveniently, since $\mathbf{a}$ is independently re-normalized at the end of the process, this re-scaling can be skipped completely, scaling $\mathbf{a}$ instead to $\|\mathbf{\Phi a}\|_2 = 1$ and continuing with the update of $\mathbf{g}$.

Combining the pieces, the final atom update process consists of the following steps: (*i*) normalizing $\mathbf{g}$ to unit length; (*ii*) solving (4.15) for $\mathbf{a}$; (*iii*) normalizing $\mathbf{a}$ to $\|\mathbf{\Phi a}\|_2 = 1$; and (*iv*) updating $\mathbf{g} := \mathbf{E}^T\mathbf{\Phi a}$. This process may generally be repeated, though we have found little practical advantage in doing so. The complete Sparse K-SVD algorithm is detailed in Alg. 4.1. Figs. 4.2, 4.3 show an example result, obtained by applying this algorithm to the same training set as that used to train the dictionary in Fig. 4.1.

4.4 Complexity of Sparse Dictionaries

Sparse dictionaries are generally much more efficient than explicit ones, and provide significant gains especially for larger dictionaries and higher-dimensional signals. In this section we discuss the complexity of sparse dictionaries and describe the cases where they are most advantageous. To focus the discussion, we concentrate on the case of Orthogonal Matching Pursuit (OMP) sparse-coding, which is a widely used method which is relatively simple to analyze.

4.4.1 Sparse Dictionary Operator Complexity

The dictionary structure (4.4) is implemented by multiplying the sparse representation $\boldsymbol{\gamma}$ by $\mathbf{A}$ and applying $\mathbf{\Phi}$. In the following, we assume that $\mathbf{A}$ has a total of pL non-zeros, and that $\mathbf{\Phi}$ has an efficient implementation with complexity T_{Φ}.

Operations with sparse matrices are not immediate to analyze, with many fac-

Algorithm 4.1 Sparse K-SVD

```
Input: Signal set $\mathbf{X}$, base dictionary $\mathbf{\Phi}$, initial dictionary representation $\mathbf{A}_0$, target atom sparsity $p$, target signal sparsity $t$, number of iterations $k$.
Output: Sparse dictionary representation $\mathbf{A}$ and sparse signal representations $\mathbf{\Gamma}$ such that $\mathbf{X} \approx \mathbf{\Phi A \Gamma}$
Init: Set $\mathbf{A} := \mathbf{A}_0$
for $n = 1 \dots k$ do
    $\forall i :\ \mathbf{\Gamma}_i := \operatorname*{Argmin}_{\boldsymbol{\gamma}} \|\mathbf{x}_i - \mathbf{\Phi A}\boldsymbol{\gamma}\|_2^2 \quad \text{Subject To} \quad \|\boldsymbol{\gamma}\|_0^0 \le t$
    for $j = 1 \dots L$ do
        $\mathbf{A}_j := \mathbf{0}$
        $I :=$ {indices of the signals in $\boldsymbol{X}$ whose reps. use $\boldsymbol{a}_j$}
        $\mathbf{g} := \mathbf{\Gamma}_{j,I}^T$
        $\mathbf{g} := \mathbf{g}/\|\mathbf{g}\|_2$
        $\mathbf{z} := \mathbf{X}_I\mathbf{g} - \mathbf{\Phi A \Gamma}_I\mathbf{g}$
        $\mathbf{a} := \operatorname*{Argmin}_{\mathbf{a}} \|\mathbf{z} - \mathbf{\Phi a}\|_2^2 \quad \text{Subject To} \quad \|\mathbf{a}\|_0^0 \le p$
        $\mathbf{a} := \mathbf{a}/\|\mathbf{\Phi a}\|_2$
        $\mathbf{A}_j := \mathbf{a}$
        $\mathbf{\Gamma}_{j,I} := (\mathbf{X}_I^T\mathbf{\Phi a} - (\mathbf{\Phi A \Gamma}_I)^T\mathbf{\Phi a})^T$
    end for
end for
```

tors affecting actual performance (see [167] for some insights on the topic). In this paper we make the simplifying assumption that the complexity of such operations is proportional to the number of non-zeros in the sparse matrix, so multiplying a vector by a sparse matrix with Z non-zeros is equivalent to multiplying it by a full matrix with αZ ($\alpha \ge 1$) coefficients (a total of $2\alpha Z$ multiplications and additions). For a concrete figure, we use $\alpha = 7$, which is roughly what our machine (an Intel Core 2 running *Matlab 2007a*) produced. With this assumption, the complexity of the sparse dictionary $\mathbf{D} = \mathbf{\Phi}\mathbf{A}$ is given by

$$T_D\left\{\textit{sparse-dict}\right\} = 2\alpha pL + T_\Phi \ . \tag{4.16}$$

The base dictionary $\mathbf{\Phi}$ will usually be chosen to have a compact representation and sub-N^2 implementation. Indeed, most implicit dictionaries provide these properties, with complexities ranging from linear to low-degree (< 2) polynomial. In the following analysis we focus on two very common types of base dictionaries, which roughly represent this range:

Separable dictionaries: Dictionaries which are the Kronecker product of several 1-dimensional dictionaries. Assuming $\mathbf{\Phi}_0 \in \mathbb{R}^{n\times m}$ is a dictionary for 1-D signals of length n, the dictionary $\mathbf{\Phi} = \mathbf{\Phi}_0 \otimes \mathbf{\Phi}_0 \in \mathbb{R}^{n^2\times m^2}$ can be constructed for representing $n \times n$ signals arranged in column-major order as vectors of length n^2. The dictionary adjoint is separable as well and given by $\mathbf{\Phi}^T = \mathbf{\Phi}_0^T \otimes \mathbf{\Phi}_0^T$. The dictionary and its adjoint are efficiently implemented by applying $\mathbf{\Phi}_0$ or $\mathbf{\Phi}_0^T$ (respectively) along each of the signal dimensions, in any order. Denoting $a = m/n$, and assuming $\mathbf{\Phi}_0$ is applied via explicit matrix multiplication, the complexity of this dictionary in the 2-D case is

$$T_\Phi = 2N\sqrt{M}(1+a) \tag{4.17}$$

where $N = n^2$ and $M = m^2$ are the dictionary dimensions. Examples of separable dictionaries include the DCT (Fourier), overcomplete DCT (Fourier), and Wavelet dictionaries, among others. Generalizations to higher dimensions are straightforward to derive.

Linear-time dictionaries: Dictionaries which are implemented with a constant number of operations per sample, so

$$T_\Phi = \beta N \tag{4.18}$$

for some constant value β. Examples include the Wavelet, Contourlet, and Complex Wavelet dictionaries, among others.

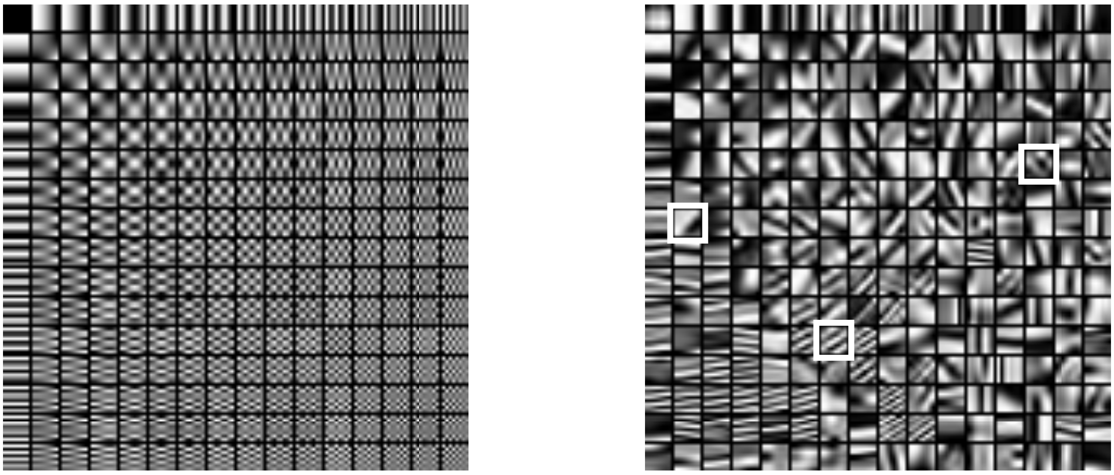


Figure 4.2: Left: overcomplete DCT dictionary for 8 × 8 image patches. Right: sparse dictionary trained over the overcomplete DCT using Sparse K-SVD. Dictionary atoms are represented using 6 coefficients each. Marked atoms are magnified in Fig. 4.3.

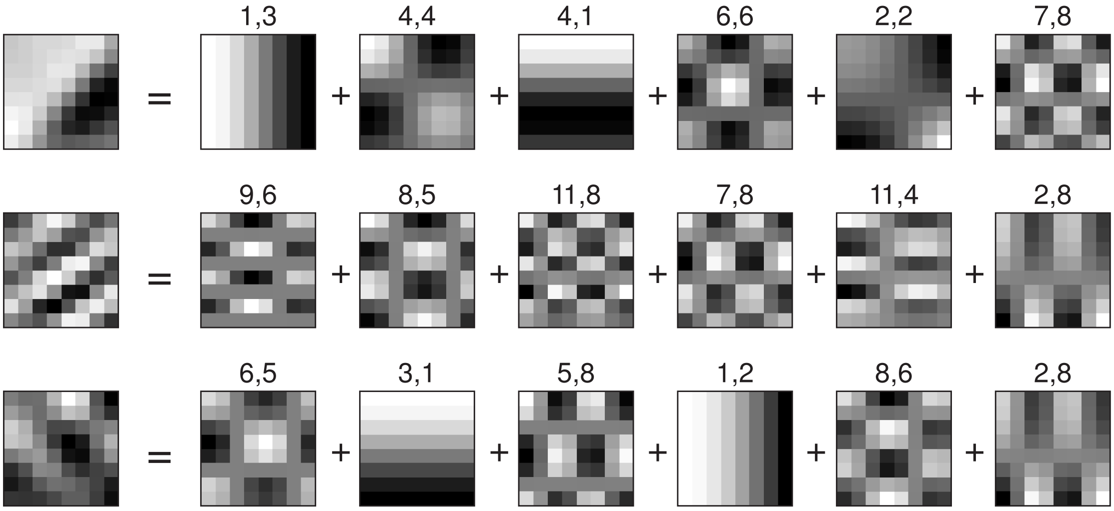


Figure 4.3: Some atoms from the trained dictionary in Fig. 4.2, and their overcomplete DCT components. The index pair above each overcomplete DCT atom denotes the wave number of the atom, with (1,1) corresponding to the upper-left atom, (16,1) corresponding to the lower-left atom, etc. In each row, the components are ordered by decreasing magnitude of the coefficients, the most significant component on the left. The coefficients themselves are not shown due to space limitations, but are all of the same order of magnitude.

4.4.2 Complexity of OMP

OMP is a greedy sparse-coding algorithm which has several efficient implementations. One of the most common ones is *OMP-Cholesky* [40, 163, 168] which employs a progressive Cholesky decomposition to perform efficient matrix inversions.

When the dictionary is represented explicitly, the number of operations performed by OMP-Cholesky can be shown to be [163]

$$T_{omp}\{\textit{explicit-dict}\} = 2tNL + 2t^2N + 2t(L+N) + t^3 \ , \tag{4.19}$$

where t is the number of OMP iterations (also the number of selected atoms), and N and L are the dictionary dimensions. Note that since $N \sim L \gg t$, the dominant term in this expression is the first one, which is associated with the explicit dictionary operator.

With a sparse dictionary, one can show that the complexity of OMP-Cholesky becomes

$$T_{omp}\{\textit{sparse-dict}\} = 4tT_\Phi + 2\alpha tpL + 2t(L+N) + t^3 \ , \tag{4.20}$$

where p is the sparsity of the dictionary atoms over the base dictionary, and α is the sparse operation overhead factor discussed above (for a derivation of this result we refer the reader to [169]). We observe that the term proportional to tNL in (4.19) has been replaced by terms proportional to tT_Φ and tpL in this expression. Therefore, when the base dictionary Φ has an efficient implementation, and assuming $p \ll N$, the sparse dictionary indeed provides an order-of-magnitude complexity advantage over an explicit one.

The complexity gain of OMP-Cholesky with a sparse dictionary is depicted in Fig. 4.4. The Figure shows the speedup factor of OMP-Cholesky with a sparse dictionary compared to an explicit one, for 2-D and 3-D signals, and using either a separable or linear base dictionary. The x-axis corresponds to the signal length N, where $N = n^d$ for $d = 2, 3$.

As can be seen, sparse dictionaries provide a pronounced performance increase compared to explicit ones, especially in the 3-D case where the speedup is around $\times 5 - \times 10$ for the separable case and $\times 10 - \times 30$ for the linear case. We also see that the speedup continues to increase as the signal becomes larger. In a practical signal processing application, where large numbers of signals are involved, this difference may make sparse dictionaries the only feasible option.

4.4.3 Dictionary Training

Seeing the complexity gain in sparse-coding, it is unsurprising that Sparse K-SVD is similarly much faster than the standard and approximate K-SVD methods. Indeed, the gain mostly stems from the acceleration in the sparse-coding step (line 5 of the algorithm). In the asymptotic case where $t \sim p \ll M \sim L \sim N \ll R$, with R the number of training signals, the complexity of the approximate K-SVD becomes proportional to the complexity of its sparse-coding method [163]. Indeed, this result is easily extended to Sparse K-SVD as well; consequently, Sparse K-SVD is faster than the approximate K-SVD by *approximately the sparse-coding speedup*.

As we will see in the experimental section, a more significant (though less obvious) advantage of Sparse K-SVD is the reduction in overfitting. This results in a substantially smaller number of examples required for the training process, and leads to a further reduction in training complexity.

4.5 Applications and Simulation Results

The sparse dictionary structure has several advantages. It enables larger dictionaries to be trained, for instance to fill-in bigger holes in an image inpainting task [31]. Specifically of interest are dictionaries for *high-dimensional* data. Indeed, employing sparsity-based techniques to high-dimensional signal data is challenging, as the complicated nature of these signals limits the availability of analytic transforms for

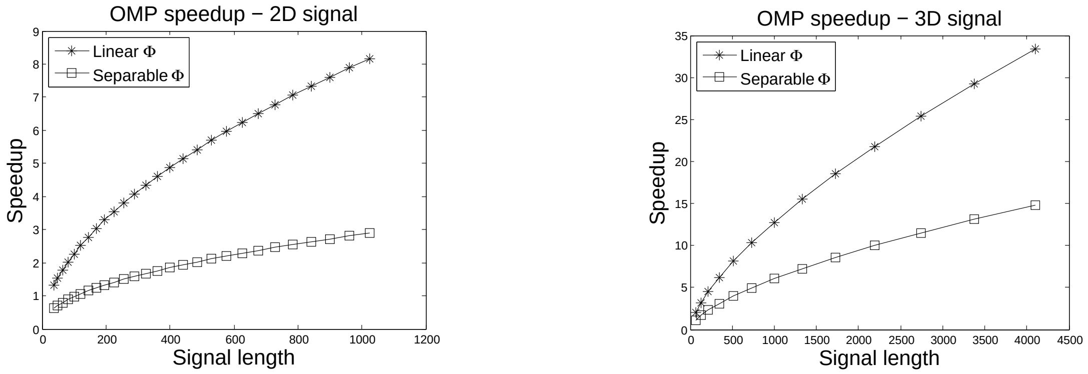


Figure 4.4: Speedup of OMP-Cholesky using a sparse dictionary compared to an explicit dictionary. Left: speedup for 2-D signals. Right: speedup for 3-D signals. Signal length is $N = n^d$ where n is the block size and $d = 2, 3$ is the number of dimensions. Dictionary size is chosen to be $n^d \times (n+3)^d$ (base dictionary is of the same size, and the matrix $\mathbf{A}$ is square). Atom sparsity is set to $p = n/2$ in the 2-D case and to $p = n$ in the 3-D case. Complexity of linear dictionary is $T_\Phi = 8N$.

them, while the complexity of the training problem constrains the use of existing adaptive techniques as well. The sparse dictionary structure — coupled with the Sparse K-SVD algorithm — makes it possible to process such signals and design rich dictionaries for representing them.

Another application for sparse dictionaries is signal compression. Using an adaptive dictionary to code signal blocks leads to sparser representations than generic dictionaries, and therefore to higher compression rates. Such dictionaries, however, must be stored alongside the compressed data, and this becomes a limiting factor when used with explicit dictionary representations. Sparse dictionaries significantly reduce this overhead. In essence, wherever a prespecified dictionary is used for compression, one may introduce adaptivity by training a sparse dictionary over this predesigned one. The facial compression algorithm in [158] makes a good candidate for such a technique, and research in this direction is currently undergoing.

In the following experiments we focus on a specific type of signal, namely 3-D computed tomography (CT) imagery. We compare the sparse and explicit dictionary structures in their ability to adapt to specific data and generalize from it. We also provide concrete CT denoising results for the two dictionary structures, and show that the sparse dictionary consistently outperforms the explicit one, while operating substantially faster. Our simulations make use of the CT data provided by the NIH *Visible Human Project* [170].

4.5.1 Training and Generalization

Training a large dictionary generally requires increasing the number of training signals accordingly. Heuristically, we expect the training set to grow *at least* linearly with the number of atoms, to guarantee sufficient information for the training process. Uniqueness is in fact only known to exist for an *exponential* number of training signals in the general case [171]. Unfortunately, large numbers of training signals quickly become impractical when the dictionary size increases, and it is therefore highly desirable to develop methods for reducing the number of required examples.

In the following experiments we compare the generalization performance of K-SVD versus Sparse K-SVD with small to moderate training sets. We use both methods to train a 512×1000 dictionary for $8 \times 8 \times 8$ signal patches. The data is taken from the *Visible Male - Head* CT volume. We extract the training blocks from a noisy version of the CT volume (PSNR=17dB), while the validation blocks are extracted directly from the original volume. Training is performed using 10,000, 30,000, and 80,000 training blocks, randomly selected from the noisy volume, and with each set including all the signals in the previous sets. The validation set consists of 20,000 blocks, randomly selected from the locations not used for training. The initial dictionary for both methods is the overcomplete DCT dictionary[6]. For Sparse K-SVD, we use the overcomplete DCT as the base dic-

[6] The 1-D $N \times L$ overcomplete DCT dictionary is essentially a cropped version of the orthogonal $L \times L$ DCT dictionary matrix. The k-D overcomplete DCT dictionary is simply the Kronecker product of k 1-D overcomplete DCT dictionaries. Note that the number of atoms in such a dictionary is L^k, and

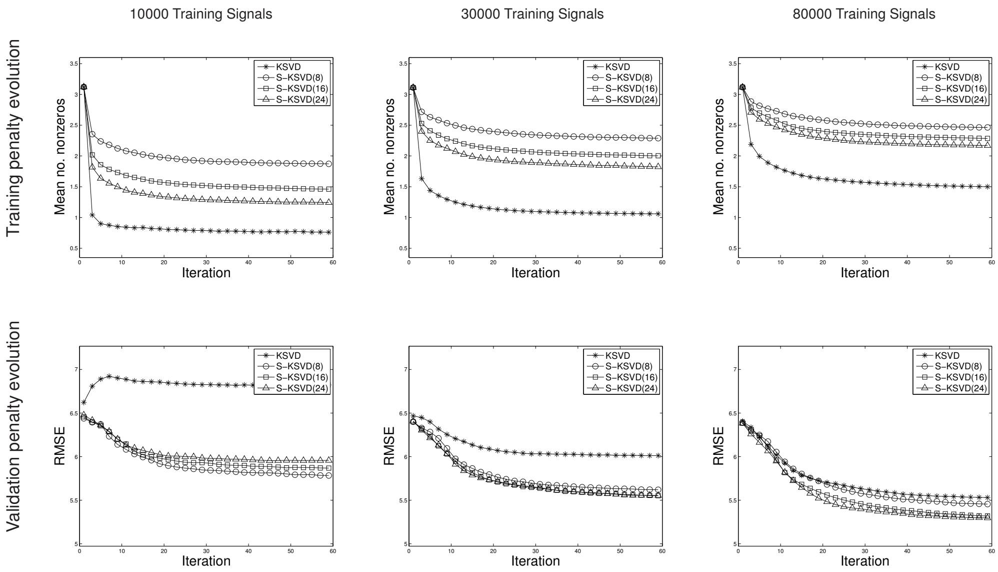


Figure 4.5: Training and validation results for patches from *Visible Male - Head*. Training signals are taken from the noisy volume (PSNR=17dB), and validation signals are taken from the original volume. Block size is $8 \times 8 \times 8$, and dictionary size is 512×1000. Training signals (noisy) are sparse-coded using an error stopping criterion proportional to the noise; validation signals (noiseless) are sparse-coded using a fixed number of atoms. Shown penalty functions are respectively the average number of non-zeros in the sparse representations and the coding RMSE. Sparse K-SVD with atom-sparsity p is designated in the legend as S-KSVD(p).

tionary, and set the initial $\mathbf{A}$ matrix to identity. The sparse dictionary is trained using either 8, 16, or 24 coefficients per atom.

Fig. 4.5 shows our results. The top and bottom rows show the performance of the K-SVD and Sparse K-SVD dictionaries on the training and validation sets (respectively) during the algorithm iterations. Following [2], we code the noisy training signals using an error target proportional to the noise, and have the ℓ^0 sparsity of the representations as the training target function. We evaluate performance on the validation signals (which are noiseless) by sparse-coding with a fixed number of atoms, and measuring the resulting representation RMSE.

We can see that the average number of non-zeros for the training signals decreases rapidly in the K-SVD case, especially for smaller training sets. However,

must have a whole k-th root (in our case, $10^3 = 1000$ atoms).

this phenomena is mostly an indication of overfitting, as the drop is greatly attenuated when adding training data. The overfitting consequently leads to degraded performance on the validation set, as can be seen in the bottom row.

In contrast, the sparse dictionary shows much more stable performance. Even with only 10,000 training signals, the learned dictionary performs reasonably well on the validation signals. As the training set increases, we find that the performance of the sparsest ($p = 8$) dictionary begins to weaken, indicating the limits of the constrained structure. However, for $p = 16$ and $p = 24$ the sparse dictionary continues to gradually improve, and consistently outperforms the standard K-SVD. It should be noted that while the K-SVD dictionary is also expected to improve as the training set is increased — possibly surpassing the Sparse K-SVD at some point — such large training sets are extremely difficult to process, to the point of being impractical.

4.5.2 CT Volume Denoising

We used the adaptive K-SVD denoising algorithm [2] to evaluate CT volume denoising performance. The algorithm trains an overcomplete dictionary using blocks from the noisy signal, and then denoises the signal using this dictionary, averaging the denoised blocks when they overlap in the result. We should mention that newer, state-of-the-art variants of the K-SVD denoising scheme, such as multiscale K-SVD denoising [31] and non-local simultaneous sparse-coding [65], could also be used here to further improve the results, however in this work we focus on the original denoising formulation for simplicity.

We performed our experiments on the *Visible Male - Head* and *Visible Female - Ankle* volumes. The intensity values of each volume were first fitted to the range [0,255] for compatibility with image denoising results, and then subjected to additive white Gaussian noise with varying standard deviations of $5 \leq \sigma \leq 100$. We tested both 2-D denoising, in which each CT slice is processed separately, and

Test	σ / PSNR	2-D Denoising			3-D Denoising		
		ODCT	KSVD	S-KSVD	ODCT	KSVD	S-KSVD
Vis. F. Ankle	5 / 34.15	43.07	43.23	43.15	44.42	**44.64**	**44.64**
	10 / 28.13	39.25	39.70	39.45	40.91	**41.24**	**41.22**
	20 / 22.11	35.34	36.12	35.87	37.57	**37.98**	**38.03**
	30 / 18.59	33.01	33.76	33.67	35.62	36.02	**36.21**
	50 / 14.15	30.15	30.43	30.48	33.07	33.48	**33.85**
	75 / 10.63	27.88	27.84	27.92	31.18	31.63	**31.98**
	100 / 8.13	26.42	26.31	26.39	29.89	30.08	**30.46**
Vis. M. Head	5 / 34.15	43.61	43.94	43.72	**45.11**	**45.12**	**45.17**
	10 / 28.13	39.34	40.13	39.70	41.46	**41.56**	**41.57**
	20 / 22.11	34.97	36.08	35.81	37.77	**38.02**	**38.10**
	30 / 18.59	32.48	33.13	33.08	35.54	35.91	**36.18**
	50 / 14.15	29.62	29.67	29.74	32.79	33.08	**33.56**
	75 / 10.63	27.84	27.75	27.82	30.73	30.69	**31.09**
	100 / 8.13	26.51	26.40	26.48	29.60	29.47	**29.72**

Table 4.1: CT denoising results using K-SVD, Sparse K-SVD, and overcomplete DCT dictionaries. Values represent Peak SNR (dB), and are averaged over 4 executions. Bold numerals denote the best result in each test up to a 0.1dB difference.

3-D denoising, in which the volume is processed as a whole. The atom sparsity for these experiments was heuristically set to $p = 6$ for the 2-D case and $p = 16$ for the 3-D case, motivated by results such as those in Fig. 4.5. Our denoising results are actually expected to improve as these values are increased, up to a point where overfitting becomes a factor. However, we preferred to limit the atom sparsity in these experiments to maintain the complexity advantage of the sparse dictionary. Further work may establish a more systematic way of selecting these values.

Our denoising results are summarized in Table 4.1. Table 4.2 shows the running times obtained by our Intel Core 2 machine for the different algorithms in the 3-D case. For completeness, Table 4.3 lists the full set of parameters used in these experiments. Some actual denoising results are shown in Fig. 4.6.

The most evident result in Table 4.1 is that 3-D denoising is indeed substan-

	Vis. F. Ankle						
Dictionary / σ	5	10	20	30	50	75	100
K-SVD	22:06:27	10:11:06	4:07:33	2:27:47	1:24:23	57:48	45:36
Sparse K-SVD	1:08:49	33:44	13:05	8:07	5:15	4:26	3:54
O-DCT	24:51	13:27	4:51	2:59	1:45	1:17	1:03

	Vis. M. Head						
Dictionary / σ	5	10	20	30	50	75	100
K-SVD	25:32:37	11:58:59	4:54:04	3:00:27	1:39:32	1:04:29	46:32
Sparse K-SVD	1:14:37	34:26	14:11	9:44	5:56	4:47	4:04
O-DCT	31:45	14:15	6:10	4:01	2:25	1:30	1:12

Table 4.2: Running times of K-SVD, Sparse K-SVD, and overcomplete DCT denoising for the results in Table 4.2 (3-D case). Timings include dictionary training. Simulations were performed on an Intel Core 2 processor, utilizing a single core. Note: running times listed here can be significantly improved, and the reader is referred to section 4.5.3 for a discussion.

	2-D Denoising	3-D Denoising
Block size	8×8	$8 \times 8 \times 8$
Dictionary size	64×100	512×1000
Atom sparsity (Sparse K-SVD)	6	16
Initial dictionary	Overcomplete DCT	Overcomplete DCT
Training signals	30,000	80,000
K-SVD iterations	15	15
Noise gain	1.15	1.04
Lagrange multiplier	0	0
Step size	1	2

Table 4.3: Parameters of the K-SVD denoising algorithm (see [2] for more details). Note that a Lagrange multiplier of 0 means that the noisy image is not weighted when computing the final denoised result.

tially more effective than 2-D denoising for this task, with significant gains of 1.5dB-4dB in all cases. These results provide further motivation for the move towards larger dictionaries and higher-dimensional signals, where sparse dictionaries are truly advantageous.

Turning to the 3-D denoising results, we find that the Sparse K-SVD matches or outperforms the standard K-SVD in all test cases. Indeed, in the low noise range ($\sigma \leq 10$), both methods perform essentially the same, and provide only marginal improvement over the fixed overcomplete DCT dictionary. However in the medium and high noise ranges ($\sigma \geq 20$), the training process becomes beneficial, and leads to improved recovery compared to the fixed dictionary. In this noise range, the increased stability of the Sparse K-SVD in the presence of noise and limited training data becomes advantageous, and it performs consistently better than standard K-SVD. We note that in some cases of very high noise, the standard K-SVD actually performs *worse* than its initial overcomplete DCT dictionary, due to overfitting and its weakness in the presence of noise.

Reviewing the results in Table 4.1, we note that the raw PSNR gain of Sparse K-SVD over standard K-SVD, while consistent, is typically small. Indeed, the main appeal of the Sparse K-SVD here is its substantially better complexity, as depicted in Table 4.2. As can be seen, the complexity advantage of Sparse K-SVD translates to a $\times 10 - \times 20$ reduction in denoising time compared to the standard K-SVD, and in fact, the long running time of standard K-SVD makes it practically useless for this task. In contrast, the Sparse K-SVD is much faster, performing especially reasonably in the interesting noise range of $\sigma \geq 20$ (in the next section we discuss methods to further reduce running time in practical applications). Thus, we conclude that the Sparse K-SVD is indeed able to introduce adaptivity where the standard K-SVD is impractical, making sparse dictionaries an appealing alternative to both fixed dictionaries and explicit learned dictionaries alike.

4.5.3 Further Acceleration and Practical Considerations

The running times in Table 4.2 may be significantly improved to allow incorporation of the Sparse K-SVD in practical applications. First, analysis of the Sparse K-SVD denoising run-time shows that it is mostly dedicated to training, while the actual denoising requires similar time to the overcomplete DCT option. In many cases, training time may be decreased (and denoising results improved) by pre-training an initial sparse dictionary on a large set of generic data of the same type as handled by the application. This method, employed e.g. in [65], reduces the number of training iterations required, and can substantially accelerate the process.

Another source of acceleration is replacing the OMP-Cholesky implementation with a more efficient OMP implementation such as Batch-OMP [163]. This option, which is not discussed here due to its relative technicality, is analyzed in detail in [169]. Experiments done with Batch-OMP show that it achieves a $\times 2 - \times 3$ speedup in Sparse K-SVD and overcomplete DCT denoising over the running times shown in Table 4.2, reducing the Sparse K-SVD denoising time to less than 5 minutes for the $\sigma \geq 20$ noise range. The software package published with this paper (see below) implements both OMP-Cholesky and Batch-OMP options.

Finally, we should mention that all algorithms discussed here are highly parallelizeable, with an expected near-linear speedup with the number of processors. Thus we expect an 8-core processor, combined with the Batch-OMP implementation, to carry out the entire 3-D Sparse K-SVD denoising process in less than a minute for any $\sigma \geq 20$.

4.5.4 Reproducible Research

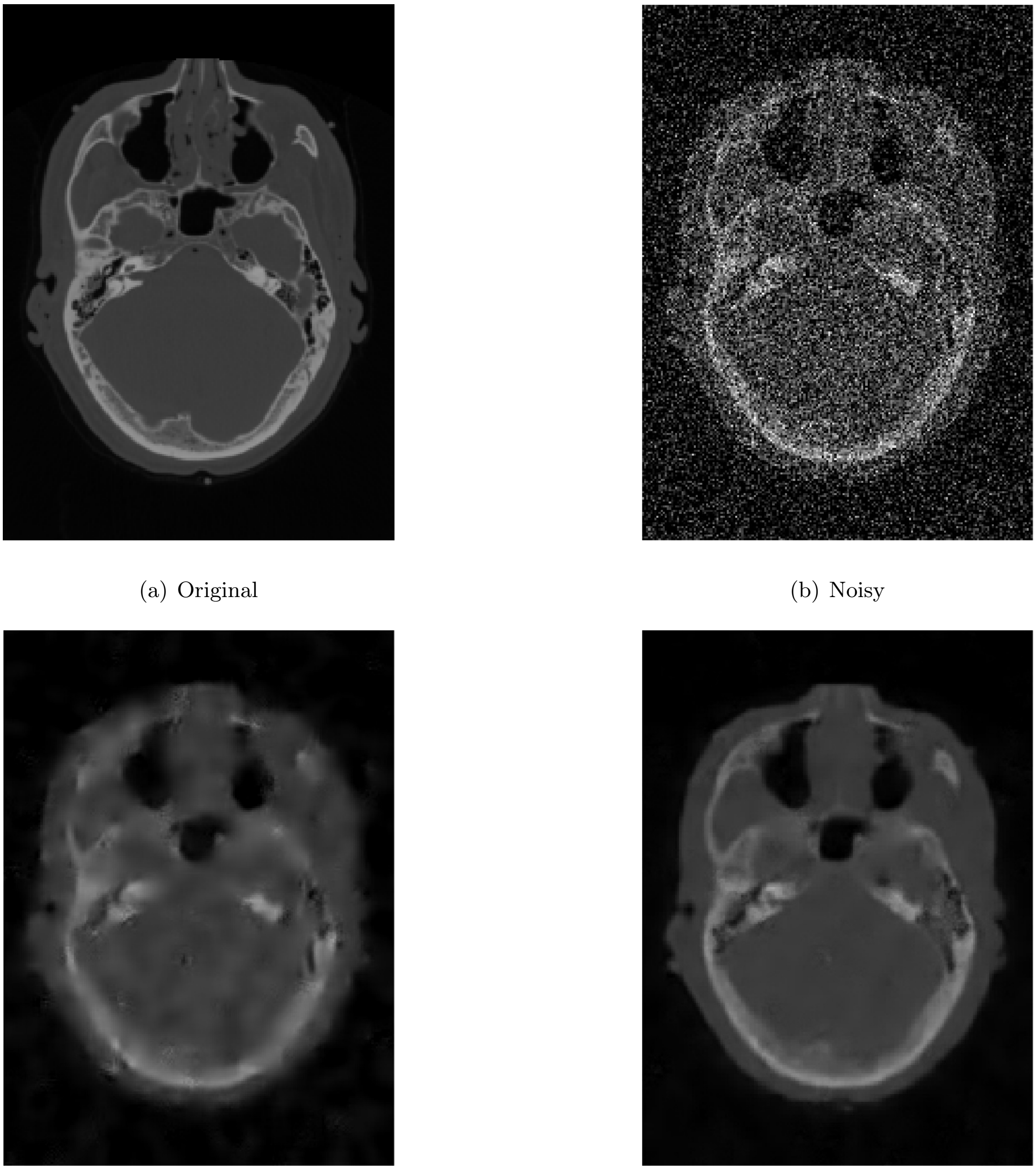
The complete K-SVD and Sparse K-SVD code reproducing the results in this paper, along with the original CT volumes used, are made available for download [172]. The code is provided as a set of Matlab packages that combine Matlab code and compilable C MEX functions. The packages implement both the OMP-Cholesky and the Batch-OMP options. See the `README` files and the accompanying documentation in each of the packages for more information.

4.6 Summary and Future Work

We have presented a novel dictionary structure which is both adaptive and efficient. The sparse structure is simple and can be easily integrated into existing sparsity-based methods. It provides fast forward and adjoint operators, enabling its use with larger dictionaries and higher-dimensional data. Its compact form is beneficial for tasks such as compression, communication, and real-time systems. It may be combined with any implicit dictionary to enhance its adaptability, with very little overhead.

We developed an efficient K-SVD-like algorithm for training the sparse dictionary, and showed that the structure provides better generalization abilities than the non-constrained one. The algorithm was applied to noisy CT data, where the sparse structure was found to outperform and operate significantly faster than the explicit representation under moderate and high noise. The proposed dictionary structure is thus a compelling alternative to existing explicit and implicit dictionaries alike, offering the benefits of both.

The full potential of the new dictionary structure is yet to be realized. We have provided preliminary results for CT denoising, however other signal processing tasks are expected to benefit from the new structure as well, and additional work is required to establish these gains. As noted in the introduction, the generality of the sparse dictionary structure allows it to be easily combined with other dictionary forms. As dictionary design receives increasing attention, the proposed structure can become a valuable tool for accelerating, regularizing, and enhancing adaptability in future dictionary structures.



(a) Original

(b) Noisy

(c) 2-D Sparse KSVD

(d) 3-D Sparse KSVD

Figure 4.6: Denoising results for *Visible Male - Head*, slice #137 ($\sigma = 50$). Images are mainly provided for qualitative evaluation, and are best viewed by zooming-in using a computer display.

Chapter 5

Adaptive Image Compression Using Sparse Dictionaries

Technical Report, Computer Science Dept., Technion, Sept. 2011.

Coauthored with Inbal Horev and Ori Bryt.

Abstract

Transform-based coding is a widely used image compression technique, where entropy reduction is achieved by decomposing the image over a *dictionary* of atoms, known to provide compaction. Existing algorithms assume the dictionary to be fixed and pre-shared by the encoder and decoder. Algorithms such as JPEG and JPEG2000 utilize *generic* dictionaries (e.g., the DCT and Wavelet dictionaries, respectively), and support compression of arbitrary signals. More recently, *content-specific* dictionaries have been used to improve compression rates by optimizing the dictionary to a specific image class. Such approaches lose generality, though, as they require sharing the specialized dictionary in advance between the encoder and decoder.

Utilizing *image-adaptive* dictionaries has the potential of both restoring generality and improving compression rates by encoding any given input image over

a dictionary specifically adapted to it. However, this approach has so far been avoided as it requires transmitting the dictionary along with the compressed data.

In this work we explore the use of the *sparse dictionary* structure to implement image-adaptive compression, aimed at generic images. This dictionary structure has a compact representation, and thus can be transmitted with relatively low overhead. We employ this structure in a compression scheme which adaptively trains the dictionary for the input image. Our results show that although this method involves transmitting the dictionary, it remains competitive with fixed-dictionary schemes such as JPEG and JPEG2000.

5.1 Introduction

Compression of natural images relies on the ability to capture and exploit redundancies found in these images. The most common compression approach, known as *transform coding*, utilizes a *dictionary* of atomic signals, such as the DCT or wavelet dictionaries, over which the image is known to be compressible. The dictionary is typically arranged as a matrix $\mathbf{D} = [\mathbf{d}_1\mathbf{d}_2 \dots \mathbf{d}_L] \in \mathbb{R}^{N \times L}$, with the columns $\mathbf{d}_i$ constituting the atoms, and $L \geq N$. Given a signal $\mathbf{x} \in \mathbb{R}^N$, compression is achieved by approximating it as a linear combination of the atoms,

$$\mathbf{x} \approx \mathbf{D}\boldsymbol{\gamma} \ , \tag{5.1}$$

where the representation vector $\boldsymbol{\gamma}$ is expected to have lower entropy than the entries of $\mathbf{x}$.

When $\mathbf{D}$ is invertible, the representation $\boldsymbol{\gamma}$ can be computed by inverting $\mathbf{D}$ and quantizing the coefficients: $\boldsymbol{\gamma} = \mathcal{Q}(\mathbf{D}^{-1}\mathbf{x})$. This is the case in the JPEG [107] and JPEG2000 [173] compression standards, where $\mathbf{D}$ is the DCT or wavelet dictionary, respectively.

When $\mathbf{D}$ is overcomplete ($L \geq N$), the null space of $\mathbf{D}$ introduces additional degrees of freedom in the choice of $\boldsymbol{\gamma}$, which can be exploited to improve its compressibility. The representation is typically selected by minimizing some penalty function $C(\boldsymbol{\gamma})$ which estimates its compressibility, such as the ℓ^0 penalty $C(\boldsymbol{\gamma}) = \|\boldsymbol{\gamma}\|_0$ which measures the number of non-zeros in the representation:

$$\hat{\boldsymbol{\gamma}} = \underset{\boldsymbol{\gamma}}{\text{Argmin}} \ \ \|\boldsymbol{\gamma}\|_0 \quad \text{Subject To} \quad \|\mathbf{x} - \mathbf{D}\boldsymbol{\gamma}\|_2^2 \leq \epsilon^2 \ . \tag{5.2}$$

Here, ϵ is the approximation error target, controlling the distortion of the compressed signal. This problem is known as the *sparse approximation* problem [36], and though NP-hard in general, it can be approximated by a wide range of techniques [11, 38, 136]. Other choices for $C(\boldsymbol{\gamma})$ include the variety of robust penalty functions such as the ℓ^p cost functions with $0 \leq p \leq 1$. All these functions promote the *sparsity* of the representation $\boldsymbol{\gamma}$ (i.e., the fast decay of its coefficients) by strongly penalizing small non-zero values. Indeed, we should remark that in practice, the compressibility of a representation is affected by additional factors other than sparsity (e.g. quantization, entropy coding etc.). Nonetheless, sparsity provides a simple and relatively reliable approximation of compressibility.

Transform-based coding schemes generally assume the dictionary $\mathbf{D}$ to be fixed, and built into both the encoder and decoder. This is the case for the JPEG family of algorithms, which are based on predetermined fixed dictionaries and are targeted at general-purpose image compression. Recently, compression schemes aimed at more specific classes of images have been developed, and show substantial gains by employing a *content adapted* dictionary which is optimized for a specific class of images [158, 174, 175]. Unfortunately, though these approaches show substantial potential for improving compression rates, a significant drawback of these methods is their loss of generality, due to the need to pre-share a specialized dictionary for every class of images.

In this work, we take a different approach and target compression of *generic* images using adaptive dictionaries. Our goal is to increase sparsity by encoding the input image over a *specifically-trained* dictionary adapted to it. This goal is ambitious, as it requires transmitting the dictionary along with the compressed data, which introduces substantial overhead. To address this, we propose using a *parametric dictionary*, which can be represented by a relatively small number of values. Several such dictionaries have been recently proposed (see [13]).

In this work we focus on the use of the *sparse dictionary* structure [33] for image compression, which we choose due to its simplicity and ability to represent relatively rich dictionaries. Our compression scheme thus trains the sparse dictionary specifically for the input image, and encodes it as part of the compressed stream. In this way, the compression method can accommodate a wide range of images, since it imposes few assumptions on their behavior. Our simulations show that even though our method must transmit the dictionary, it consistently outperforms JPEG compression, and comes close to JPEG2000 in several cases. We view these as significant and encouraging results, which demonstrate the feasibility of the image-adaptive approach, and open the door to further research.

5.1.1 Related Work

Several works on image compression using content-adaptive dictionaries have been recently published. In all these works, the trained dictionary is assumed to be known to both the encoder and decoder. One of the first works to successfully employ this approach is [158], where the authors propose an algorithm for facial image compression. The algorithm employs a pre-processing geometric alignment step, followed by a sparse approximation of the image patches over a set of pre-trained dictionaries. The method is shown to achieve dramatically higher compression rates than JPEG and JPEG2000 for facial imagery due to the optimized dictionaries, and clearly demonstrates the potential of content-aware compression. Unfortunately, this approach is not readily extendible to more complex classes of images.

A different method, applicable to a wider range of image classes, is proposed in [174]. In this work, a set of orthogonal dictionaries is pre-trained for a given class of images, and the compression is implemented by allowing each patch in the input to select its optimal dictionary from the set. The authors show that for facial images, this method typically outperforms JPEG and comes close to JPEG2000. Experiments with natural images show varying performance, matching or surpassing JPEG.

Finally, a method based on iteration-tuned dictionaries (ITDs) has been recently proposed [175]. In this work, a single hierarchical ITD is pre-trained for a specific image class, and used to encode the input image patches. The authors test their method with facial images, and show that it can convincingly outperform JPEG and JPEG2000 for this class of images. Other classes of images remain to be investigated.

5.1.2 Report Organization

This report is organized as follows: In section 5.2 we review the sparse dictionary structure, which forms the core of our algorithm. The compression scheme is described in section 5.3, followed by results in section 5.4. We conclude and discuss future research directions in section 5.5.

5.2 Sparse Dictionaries

The *sparse dictionary* structure is a parametric dictionary model recently proposed as a means of bridging the gap between analytic and trained dictionaries [33]. It is a simple and effective structure based on sparsity of the atoms over a known base dictionary. The motivation for this structure comes from the observation that dictionaries trained from natural image data are typically highly structured, and show notable regularity. For example, Fig. 5.1 shows a dictionary trained using the K-SVD algorithm [28] on a set of 8×8 natural image patches. The regularity of the trained atoms suggests that these atoms *themselves* may have some underlying sparse structure over a more fundamental *base dictionary*. Thus,

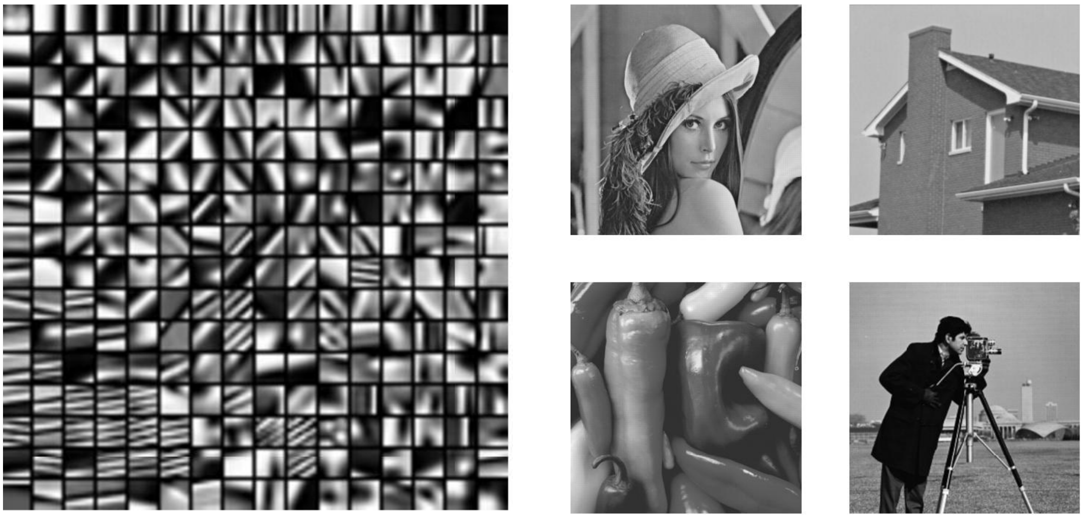


Figure 5.1: Left: overcomplete dictionary for 8 × 8 image patches, trained using the K-SVD algorithm. Right: images used for the training.

according to this view, the dictionaries describing different images may not be completely independent, and instead have a *common underlying explanation* in the form of the base dictionary. This base dictionary in turn consists of a set of signals — which can be thought of as ”sub-atomic” signals — from which all observable dictionary atoms are formed.

Formally, the sparse dictionary structure represents each atom of the dictionary as a sparse combination of atoms from a prespecified base dictionary $\mathbf{\Phi}$. The dictionary is therefore expressed as

$$\mathbf{D} = \mathbf{\Phi}\mathbf{A} \ , \tag{5.3}$$

where $\mathbf{A}$ is the atom representation matrix, and is assumed to be sparse. For simplicity, we assume $\mathbf{A}$ has a fixed number of non-zeros per column, so $\|\mathbf{a}_i\|_0 \leq p$ for some p. The base dictionary $\mathbf{\Phi}$ is a *fixed* non-adaptive dictionary which is part of the model.

Benefits of this model include adaptability (via modification of $\mathbf{A}$), efficiency (assuming $\mathbf{\Phi}$ has an efficient implementation), and compact representation (as only $\mathbf{A}$ requires specification). Training the sparse dictionary is done using the *Sparse K-SVD* algorithm [33], which efficiently adapts the matrix $\mathbf{A}$ given a set of examples. The algorithm alternates between sparse coding and dictionary update steps, similar to the original K-SVD algorithm [28]. We refer the reader to [33] for a complete description of the algorithm.

5.3 Adaptive Image Compression

The adaptive encoding process is summarized in Fig. 5.2. The process begins by partitioning the image to non-overlapping patches and subtracting the mean (DC) value from each. The DC values are subsequently quantized, and their running differences are entropy coded. The DC-free patches, which contain the bulk of the image information, are used to train a sparse dictionary using Sparse K-SVD. As the base dictionary, we use the overcomplete DCT[1], which is known to be an efficient generic dictionary for representing small image patches.

The outcome of this training is a matrix $\mathbf{A}$ describing an *image-specific* dictionary for representing the image patches. This matrix undergoes quantization and is then used to encode the DC-free patches. We perform sparse coding over the quantized dictionary $\mathbf{D}_q = \mathbf{\Phi}\mathbf{A}_q$ to allow inversion of the process at the decoder. For the sparse coding, we use a variant of Orthogonal Matching Pursuit (OMP) [38] which we name *Global OMP*. The sparse coding step produces a sparse matrix $\mathbf{\Gamma}$ with the sparse representations of the patches as its columns, and $\mathbf{\Gamma}$ is subsequently quantized to form $\mathbf{\Gamma}_q$. Finally, both $\mathbf{A}_q$ and $\mathbf{\Gamma}_q$ are fed to a sparse matrix encoder which generates the compressed representation of the DC-free content. The full compressed stream consists of the encoded DC values and the two compressed sparse matrices.

We note that the sparse dictionary is learned from zero-mean patches, however the sparse coding step is performed with patches from which *quantized* DC values were subtracted, and therefore may have non-zero means. We resolve this

[1] The overcomplete DCT dictionary is an extension of the standard DCT dictionary which allows non-integer wave numbers.

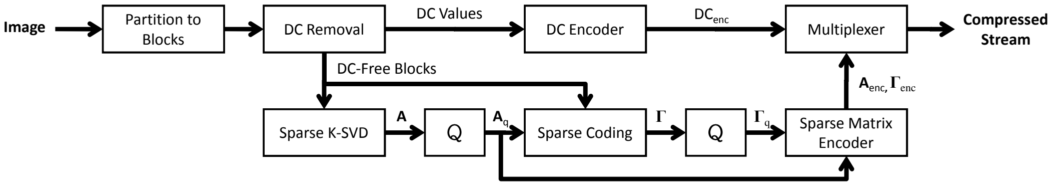


Figure 5.2: The proposed encoding scheme.

discrepancy by adding a fixed constant-valued DC atom to the trained dictionary, and implicitly assume its existence as the first atom of the dictionary in both the encoder and decoder. Decoding the stream is straightforward and efficient, and includes reversing the sparse matrix encoding, computing the DC-free patches $\mathbf{X} = \mathbf{\Phi A \Gamma}$, and restoring the encoded DC values.

In the next sections we describe in more detail the key components of the scheme.

5.3.1 Global OMP

Our implementation of the compression scheme accepts a target PSNR as the control of the output rate. In the sparse coding stage, this target can be enforced individually for each image patch by setting ϵ in (5.2) to $\epsilon^2 = I_{max}^2 b / 10^{\frac{PSNR}{10}}$, where b denotes the number of pixels in a patch. Alternatively, we can lift this constraint and allow the error to be distributed *arbitrarily* among the patches. This results in a more flexible sparse coding scheme which potentially achieves higher sparsity. Thus, we solve a *global* sparse coding problem for all image patches simultaneously:

$$\underset{\mathbf{\Gamma}}{\text{Min}} \; \|\mathbf{\Gamma}\|_0 \quad \text{Subject To} \quad \|\mathbf{Y} - \mathbf{D}_q \mathbf{\Gamma}\|_F^2 \leq \epsilon_g^2 \; . \tag{5.4}$$

Here, $\mathbf{Y}$ is a matrix with the image patches as its columns, and ϵ_g is the global error target for the image.

Problem (5.4) can be formulated as a sparse-coding problem for the column-stack representation $\mathbf{y}$ of $\mathbf{Y}$ over the dictionary $\mathbf{I} \otimes \mathbf{D}_q$ (the block matrix containing instances of $\mathbf{D}_q$ as its main diagonal):

$$\underset{\boldsymbol{\gamma}}{\text{Min}} \; \|\boldsymbol{\gamma}\|_0 \quad \text{Subject To} \quad \|\mathbf{y} - (\mathbf{I} \otimes \mathbf{D}_q)\boldsymbol{\gamma}\|_2^2 \leq \epsilon_g^2 \; . \tag{5.5}$$

We perform the sparse coding using OMP, which can be implemented efficiently since the computations can be localized to a single patch each iteration. Specifically, we store for each patch its current residual vector, sparse coefficients, and inner-products with the dictionary atoms. At the selection stage, we choose the largest inner product among all image patches, which determines both the patch to process and the atom to add to its representation. The addition of the atom involves only local updates to the patch information, and thus introduces no overhead compared to encoding the patch independently.

We name the resulting process *Global OMP*, as it globally processes all image patches towards a single collective error target. We use it in the dictionary training as well as the sparse coding steps, in order to better tune the learned result to the global process.

5.3.2 Quantization

We quantize the non-zero values in $\mathbf{A}$ and $\mathbf{\Gamma}$ using a uniform quantizer. While the distribution of these values is highly non-uniform, it is known that using uniform quantization followed by entropy coding generally outperforms non-uniform quantization. Of course, a side effect of the quantization of $\mathbf{\Gamma}$ is that the PSNR target achieved by the sparse coding step is lost. To restore the desired PSNR target, we employ a simple iterative refinement process, in which coefficients are added to $\mathbf{\Gamma}$ to compensate for the quality loss.

We begin with the original PSNR target and its associated error value ϵ_g. We denote this original (user-specified) PSNR target by p_0 and the achieved PSNR after quantization by q_0 ($< p_0$), and let $r_0 = p_0 - q_0$. Assuming r_0 is relatively small, we estimate that the quantization-induced PSNR loss will be approximately the same for any target PSNR close to p_0. This implies that the target PSNR $p_1 = p_0 + r_0$ should roughly achieve the user-specified PSNR target *after* quantization. We therefore add coefficients to $\mathbf{\Gamma}$ until reaching the updated target p_1, by *continuing* the greedy Global OMP from the point it terminated at p_0. After quantization, if the result is still below the desired PSNR target, the Global OMP target can be raised again using a similar process, based on the updated PSNR loss $r_1 = p_1 - q_1$. The process repeats as necessary until reaching the user-specified PSNR target $p_0 \pm \delta$ for some δ. The overall process is efficient and requires relatively few repetitions (typically 2-5). It should be noted that since the Global OMP is continued rather than restarted, the overhead of these repetitions (compared to having known the ”correct” sparse coding target to begin with) is small.

5.3.3 Sparse Matrix Encoding

Our sparse matrix encoder represents the matrices $\mathbf{A}_q$ and $\mathbf{\Gamma}_q$ in *column compressed* (CCS) form. It encodes the quantized values of the coefficients via entropy coding, and the locations of these coefficients via difference coding of the row indices, followed by entropy coding.

A useful observation is that the order of the columns in $\mathbf{A}$ is arbitrary, and is essentially a degree of freedom of the representation. Indeed, we can apply any permutation to the columns of $\mathbf{A}$, along with the same permutation to the rows of $\mathbf{\Gamma}$, without altering the product $\mathbf{A\Gamma}$. This freedom can be used to improve the compressibility of the row indices in $\mathbf{\Gamma}$. In this work we reorder the columns of $\mathbf{A}$ such that they become ordered in *decreasing order of popularity.* In other words, the rows of $\mathbf{\Gamma}$ are sorted in descending order of non-zero count. This sorting results in concentration of the non-zero values in $\mathbf{\Gamma}$ near the top of the matrix, and thus the overall entropy of the index differences is reduced.

To facilitate the difference-coding of the row indices in $\mathbf{\Gamma}$, we transmit for each column the index of the first non-zero value, followed by a sequence of differences for the remaining indices. Owing to the above sorting, the order of the rows in $\mathbf{\Gamma}$ is such that the index of the first non-zero value in each column is typically small, and hence has a particulary low entropy. We thus use a dedicated entropy coder for the indices of the first value in each column, and a separate entropy coder for the index differences. We indicate an empty column in $\mathbf{\Gamma}$ by sending the special symbol 0 as the index of the first non-zero value in that column (we index valid matrix rows from 1). For a non-empty column — which can have a variable number of non-zero values — we indicate the end of its index sequence by sending the special symbol 0 as the index difference.

For the coefficient values, we entropy-code their absolute value as a consecutive stream (a different stream for $\mathbf{A}$ and $\mathbf{\Gamma}$). The signs are sent unprocessed.

5.3.4 Entropy Coding

The entropy coding in this work is implemented using an arithmetic coder. We note that given a set of symbols, the arithmetic coder and decoder require the symbol probabilities $\{p_i\}$ as side information. These probabilities are determined by the encoder, and must be transmitted to the decoder. To avoid sending floating-point numbers, we quantize and transmit the log-probabilities $\log_2(1/p_i)$. These values represent the optimal codeword lengths of the symbols, and thus have a relatively small range which can be uniformly quantized. We have found that using very few bits $(5-6)$ for the quantized values results in practically no increase to the code length, while providing an effective way of transmitting the side information.

5.3.5 Parameter Tuning

One can imagine that compression schemes, such as the one described in [158], rely on many parameters that need to be set before actual coding. Some of these parameters can be predetermined, while others depend on the image content and the requested output quality. The proposed scheme involves several such parameters as well. In this section we discuss the main parameters in the scheme — the patch size, the dictionary size, the atom sparsity and the quantization step sizes — and their selection process in our implementation.

Patch Size	**Dictionary Size**	**Atom Sparsity**
3×3	300	8
4×4	300	10
5×5	200	12
6×6	200	12
7×7	200	14
8×8	200	16
9×9	150	16
10×10	150	16
11×11	150	18
12×12	120	18
13×13	100	20
14×14	100	20

Table 5.1: Dictionary size and atom sparsity for each patch size.

Our experiments have shown that of the mentioned parameters, only a few have a significant effect on compression performance. Based on these experiments, our system implements a *semi-automatic* parameter tuning process which requires no manual intervention. We should remark that although this heuristic process has been found relatively effective, manual experimentation has verified that it is still sub-optimal, and improved results can be achieved by further refining it.

For the dictionary size and atom sparsity, our system employs hard-coded values which depend only on the patch size, as detailed in Table 5.1. Based on our experiments, we have found that the optimal values for these parameters are quite consistent among images assigned with the same patch sizes, with compression results remaining stable when deviating from these values. Thus, fixed values for these two parameters suffice for our encoding system. In the same way, our base dictionary size is fixed as well, and depends only on the patch size. For a patch size of $N \times N$, our overcomplete DCT base dictionary is of size $N^2 \times (N+2)^2$, i.e., the Kronecker product of two 1-D $N \times (N+2)$ overcomplete DCT dictionaries.

Selecting the quantization steps is a more elaborate process. Our scheme involves three quantization step sizes which must be chosen: one for the DC values, and two for the non-zero values in $\mathbf{A}$ and $\mathbf{\Gamma}$. Beginning with the DC quantization step, we recall that our scheme adds a fixed DC atom to the trained dictionary to overcome DC quantization effects. Thus, coarser DC quantization results in more non-zero values appearing in $\mathbf{\Gamma}$, associated with this DC atom. Our selection rule for the DC quantization step heuristically chooses this value such that the increase in non-zero count in $\mathbf{\Gamma}$ due to the DC quantization is around 6%.

Regarding the quantization steps for the non-zeros in $\mathbf{A}$ and $\mathbf{\Gamma}$, we notice that for both these values there is a direct trade-off between the harshness of the quantization and the number of coefficients that will be required in $\mathbf{\Gamma}$ to achieve a given PSNR target. Specifically, by coarsening the quantization, it remains possible (up to some point) to satisfy the PSNR target, but at the expense of more non-zero values added to $\mathbf{\Gamma}$. Our system employs a heuristic process which simultaneously selects both step sizes, with the goal of achieving a $\sim$ 0.85dB PSNR loss due to the quantization. This loss is then compensated for by adding coefficients to $\mathbf{\Gamma}$. To achieve this goal, we extend the iterative process in 5.3.2 to repeatedly refine both step sizes based on the current PSNR loss, modifying the step sizes as necessary according to a set of empirically designed rules.

Of all the compression parameters, we have found the most influential one to be the patch size. As mentioned above, given the optimal patch size, many other parameters of the process can be immediately set. Unfortunately, we have not yet found a sufficiently effective heuristic for selecting this size. Thus, our system determines the optimal size for each image by applying several predetermined patch sizes in a highly reduced compression scheme, and choosing the size that achieves the optimal rate for the target PSNR. We consider patch sizes from 3×3 pixels to 14×14, as listed in Table 5.1. As expected, we have found that lower PSNR targets generally prefer larger patch sizes, due to the reduced accuracy

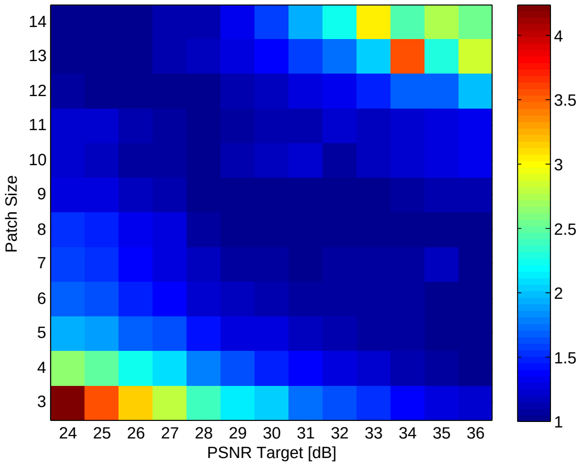


Figure 5.3: The effect of the patch size choice on the compressed file size, for the image *Barbara*. For each PSNR target, the corresponding column shows the ratios between the obtained file sizes for different patch size choices, and the optimal file size for that PSNR.

required by the compression. Fig. 5.3 illustrates the relation between the patch size and the resulting file size for the image *Barbara*. For each PSNR target, the figure shows the relative increase in file size incurred by different patch size choices, compared to the optimal size. As can be seen, compression performance is quite stable under minor deviations from the optimal patch size, though selecting an unsuitable size can result in a substantially large file.

For further details on the parameter tuning process, we refer the reader to [176].

5.4 Results

We have tested the proposed scheme on a variety of images, and the results for seven standard test images are presented below. The Sparse K-SVD results were produced using the parameter selection process described in the previous section. The JPEG and JPEG2000 images were produced using MATLAB R2010a.

Representative compression results are listed in Table 5.2. Figs. 5.4-5.6 show the corresponding compressed images, and Fig. 5.7 presents comparative rate-distortion graphs. The three parts in the table correspond to the three figures

5.4,5.5,5.6. In each part, all test images are compressed to the same target PSNR by the Sparse K-SVD compression scheme, and the JPEG and JPEG2000 algorithms are tuned to match the resulting file size. The three cases represent low, medium and high bit-rates, corresponding to PSNR targets of 25dB, 29dB and 34dB for the Sparse K-SVD. The rate-distortion graphs summarize our complete results for the seven test images.

As can be seen, our scheme consistently outperforms JPEG, and comes close to JPEG2000 in several cases. Our method typically performs better on images containing more texture, owing to the ability of the dictionary to capture and efficiently represent repetitive behavior. Similar to the JPEG algorithm, our method suffers from blockiness due to partitioning of the image. This artifact can likely be reduced by employing a post-processing deblocking scheme.

Finally, Fig. 5.8 demonstrates a typical decomposition of the compressed stream resulting from our encoder. As can be seen, the indices of the representation coefficients in $\mathbf{\Gamma}$ occupy the majority of the compressed file. This behavior is due to the random structure of the coefficients in $\mathbf{\Gamma}$, which exhibits little organization or repetitiveness, and is thus difficult to compress efficiently.

		File Size	**PSNR (dB)**		
Figure	**Image**	**(KB)**	**JPEG^2**	**Sparse K-SVD**	**JPEG2000**
5.4	barbara	5.9 ± 0.1	22.3	25	26.96
	lena	3.4 ± 0.1	—	25	30.11
	peppers	3.9 ± 0.1	—	25	30.57
	pirate	5.1 ± 0.1	23.28	25	26.63
	zentime	5.4 ± 0.1	23.12	25	25.83
	table	14.5 ± 0.1	23.17	25	25.6
	dollar	23.4 ± 0.1	22.92	25	25.58
5.5	barbara	11.1 ± 0.1	25.8	29	29.79
	lena	4.77 ± 0.1	25.56	29	31.69
	peppers	6.33 ± 0.1	28.23	29	32.09
	pirate	13.3 ± 0.1	28.36	29	30.31
	zentime	16.3 ± 0.1	27.86	29	29.97
	table	30.2 ± 0.1	26.45	29	29.9
	dollar	41.9 ± 0.1	26.23	29	30.2
5.6	barbara	26.7 ± 0.1	31.65	34	35.65
	lena	12.6 ± 0.1	33.41	34	35.83
	peppers	14.8 ± 0.1	33.61	34	35.49
	pirate	36.6 ± 0.1	32.91	34	35.92
	zentime	39.3 ± 0.1	32.77	34	36.6
	table	59 ± 0.1	30.74	34	35.11
	dollar	72 ± 0.1	31.32	34	36.6

Table 5.2: Quantitative comparison of JPEG[2], JPEG2000 and Sparse K-SVD compression. Each part of the table corresponds to a single target PSNR for the Sparse K-SVD (25dB, 29dB and 34dB), and the JPEG and JPEG2000 are tuned to match the resulting file size.

[2] The empty entries in the table correspond to cases where the Sparse K-SVD file sizes were below the possible minimum of the JPEG algorithm. The smallest file sizes achieved by JPEG for *Lena* and *Peppers* were 4.27KB (24.25dB) and 4.42KB (24.3dB), respectively. The corresponding images are shown in Fig. 5.4.

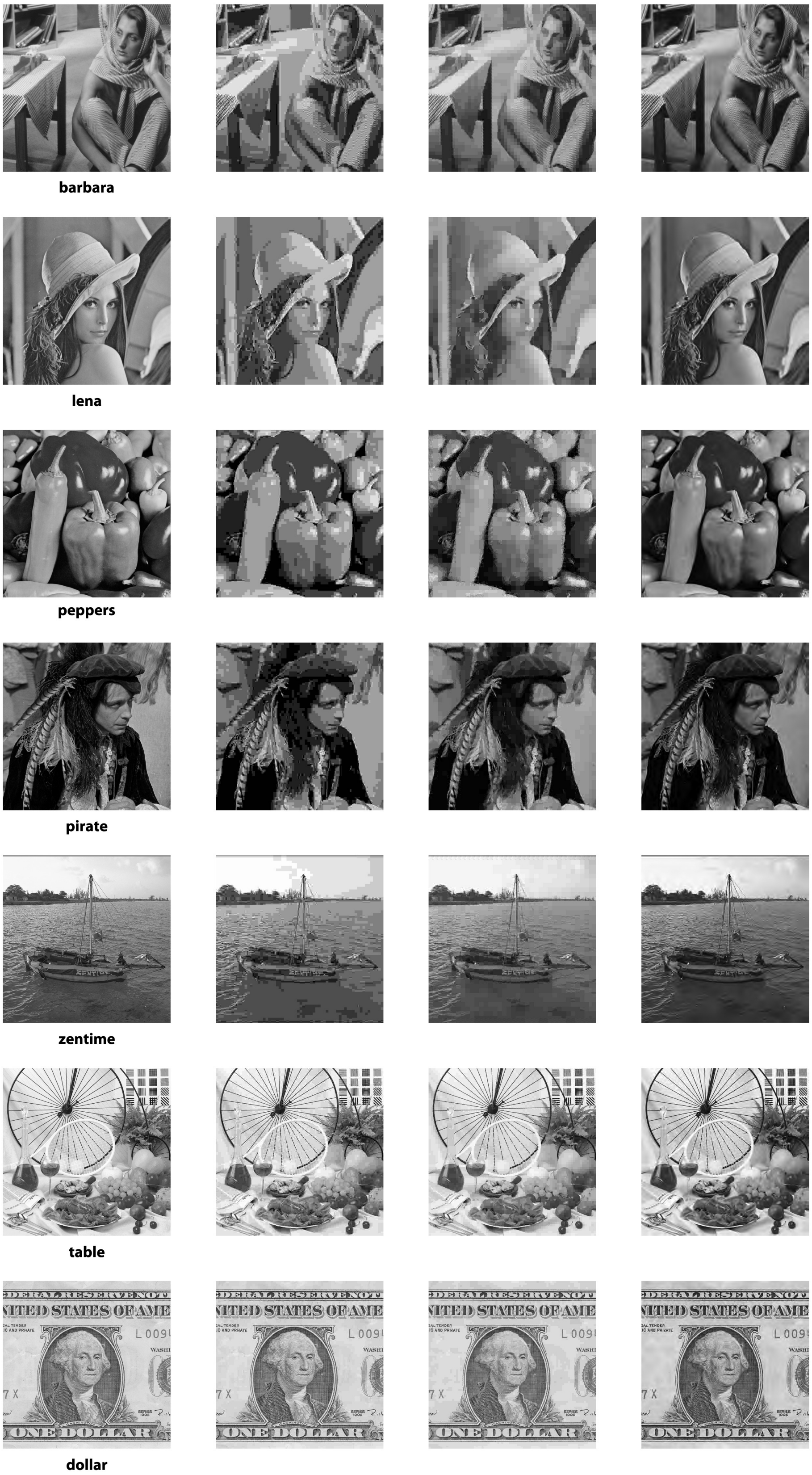


Figure 5.4: Visual comparison of the schemes (low bit-rate). Left to right: Original, JPEG, Sparse K-SVD and JPEG2000. For the images *Lena* and *Peppers*, where JPEG could not achieve the target file size, the shown images use the lowest possible quality settings.

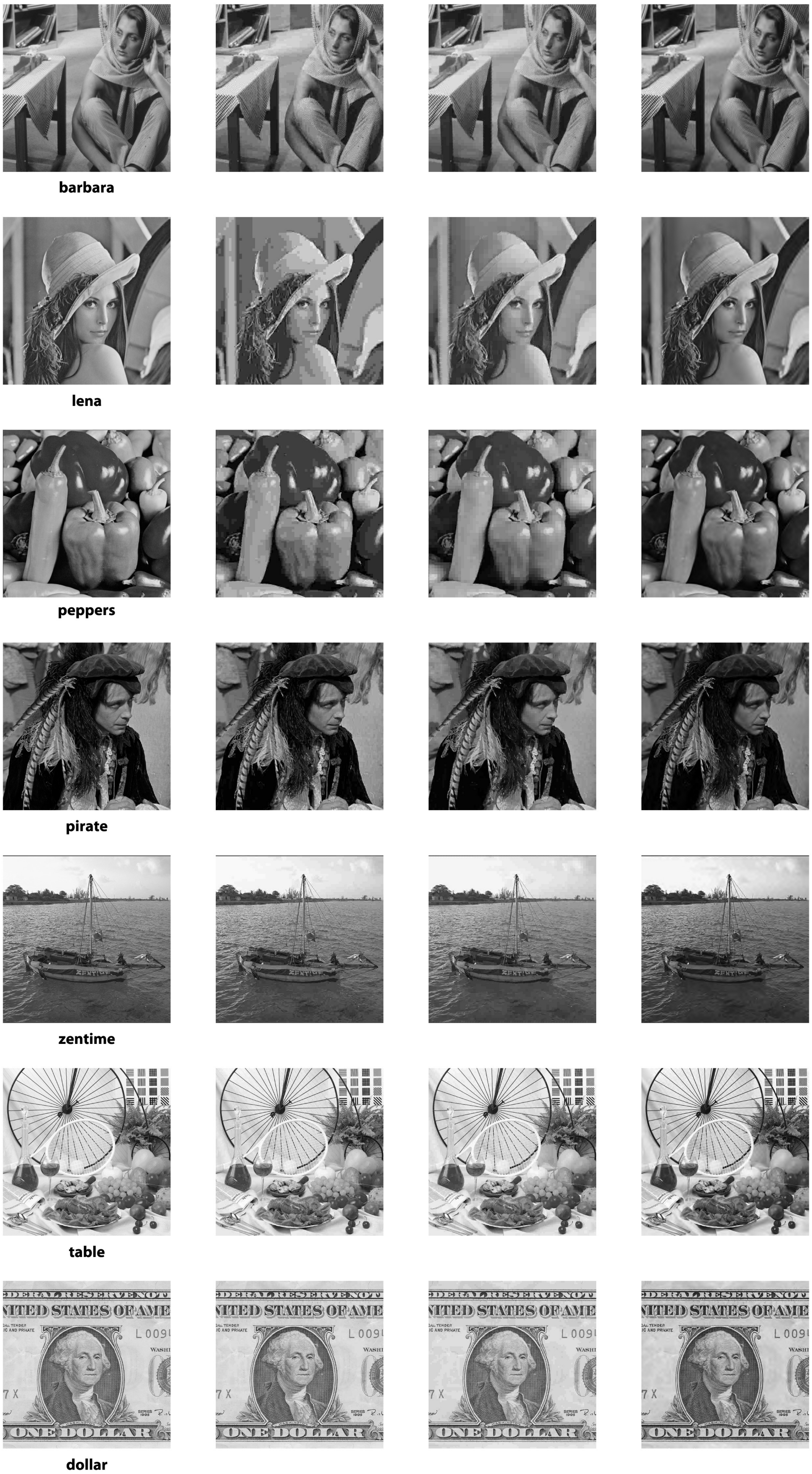


Figure 5.5: Visual comparison of the schemes (medium bit-rate). Left to right: Original, JPEG, Sparse K-SVD and JPEG2000.

Figure 5.6: Visual comparison of the schemes (high bit-rate). Left to right: Original, JPEG, Sparse K-SVD and JPEG2000.

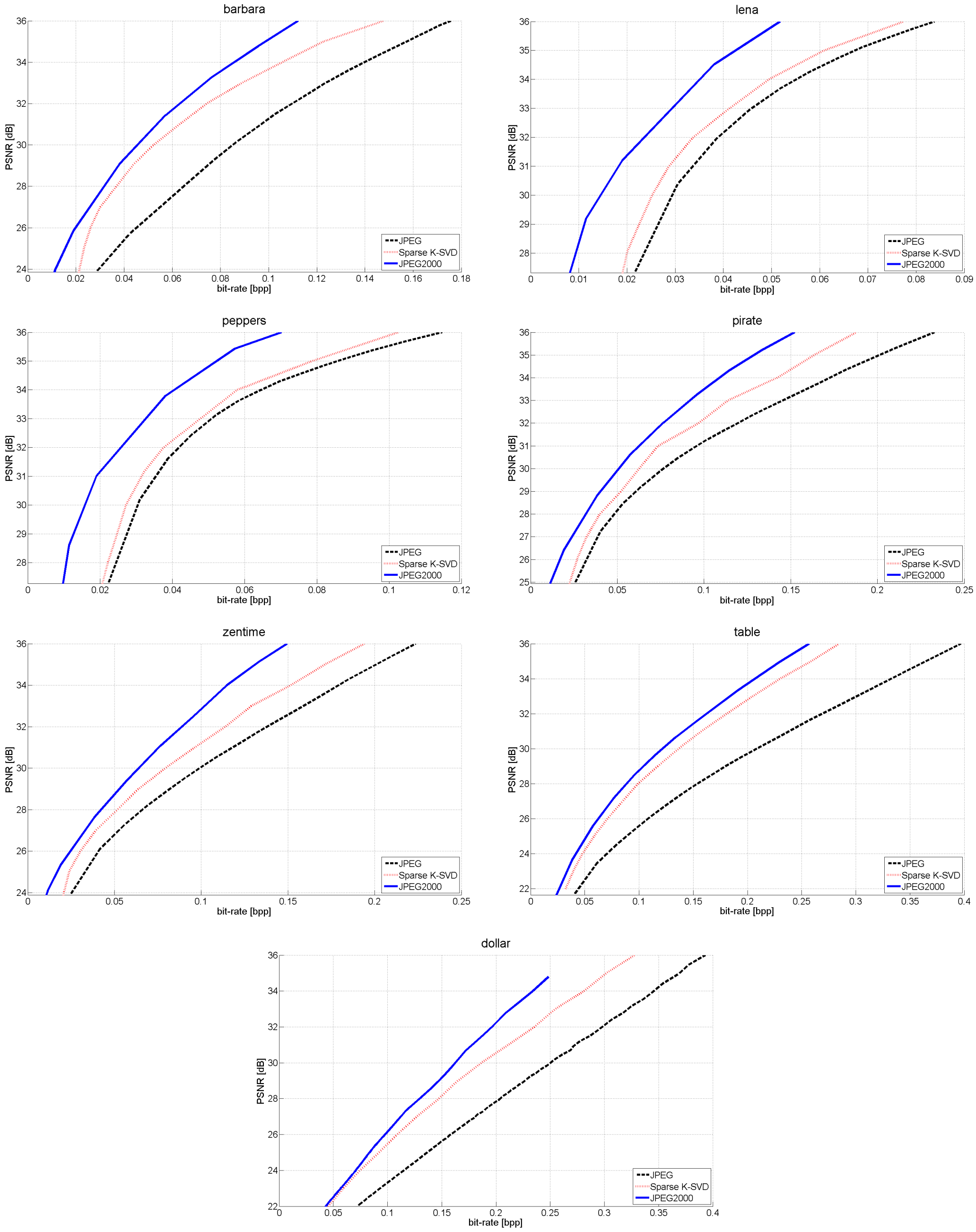


Figure 5.7: Rate-distortion curves for the seven test images. Comparison of JPEG, Sparse K-SVD, and JPEG2000 compression.

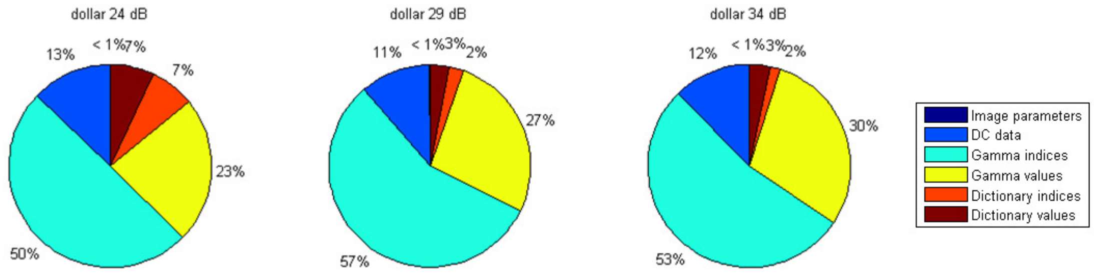


Figure 5.8: Decomposition of the compressed stream for the image *Dollar*.

5.5 Conclusion and Future Directions

This work has presented a new image compression scheme based on image-adaptive dictionaries. The system is unique in that it encodes the image over a dictionary *specifically trained* for the input. This approach, which requires transmission of the dictionary as part of the compressed stream, is made possible owing to the compact representation of the sparse dictionary structure.

We have shown that despite the overhead in sending the dictionary, our system consistently outperforms the JPEG algorithm, which is a similar patch-based scheme, but utilizes a pre-shared fixed dictionary. Indeed, while our current implementation does not reach JPEG2000 performance, our results remain significant in that they demonstrate the feasibility and potential of the adaptive approach. Such an approach, as far as the authors are aware of, has so far been considered impractical.

Many enhancements to the scheme could be introduced. Most notably, working with several image scales could more efficiently represent differently-sized features, as well as eliminate the need to select a patch size for each input individually. Alternatively, enabling variable-sized patches based on local image complexity could also accomplish this. Another interesting way to achieve multi-scale behavior is to apply the scheme on the wavelet (or other multi-scale) transform of the image, which could at the same time reduce blockiness effects.

In another direction, an important observation is that the encoded indices occupy a significant part of the resulting compressed stream. Thus, discovering hidden patterns in $\mathbf{\Gamma}$, or alternatively, modifying the sparse coding process to create more regular patterns, could dramatically improve compression efficiency. Finally, the scheme could be extended to allow fixed pre-shared parts in the dictionary alongside adaptive ones, thus reducing the overall cost of sending the dictionary.

5.6 Acknowledgements

The authors would like to thank Prof. Michael Elad, Prof. David Malah and Nimrod Peleg for their guidance and helpful advice throughout this work.

Chapter 6

Learning ℓ^0 Analysis Dictionaries

Joint work with Michael Elad.

Abstract

The synthesis-based sparse representation signal model has drawn considerable attention over the past decade. The synthesis approach models signals as coming from linear combinations of a few columns, or *atoms*, from a given dictionary. In this work we concentrate on an alternative *analysis* model, where signal representations come from the *inner products* of the signals and the dictionary atoms, producing a sparse outcome. According to this approach, the atoms are arranged as the *rows* of the analysis dictionary, and the signals of interest are described as *orthogonal* to sets of rows from this dictionary. In this chapter we present this new modeling approach, and propose an algorithm for learning the analysis operator from sparse examples. The algorithm we develop is closely related to the K-SVD training algorithm for synthesis dictionaries, and we thus name it *Analysis K-SVD*. Our experiments demonstrate the effectiveness of the algorithm in recovering an underlying analysis dictionary from examples, as well as its ability to discover meaningful structures in natural image data.

6.1 Introduction

The ℓ^0 analysis model is a new signal model in which signals are described in terms of *orthogonality* to the dictionary atoms. Very little is currently known about this model, with only a handful of works published on the topic [86, 87, 177]. The model describes the signals of interest $\mathbf{x} \in \Omega$ as coming from subspaces orthogonal to sets of rows in the analysis dictionary. Thus, $\|\mathbf{\Omega x}\|_0 = L - P$, where P is the number of atoms $\mathbf{x}$ is orthogonal to. For natural signals, it is well-known that localized derivative operators exhibit highly sparse behavior, i.e., many inner-products are near-zero. Dictionaries such as short-time Fourier [106], wavelets [7], curvelets [16], contourlets [18], and high-order derivatives, constitute good examples of this behavior.

Particular motivation for the ℓ^0 analysis model comes from the observation in [1] that the local modes (the high probability signals) of the ℓ^1 analysis model are orthogonal to large sets of rows in the analysis dictionary. This is parallel to the synthesis model, where it is known that the high probability signals of the ℓ^1 model constitute sparse combinations of columns from the synthesis dictionary. Indeed, the outstanding success of the resulting ℓ^0 synthesis formulation naturally raises interest in the yet unexplored ℓ^0 variant of the analysis model.

The new analysis signal model raises several interesting questions. The first concerns recovery: given a possibly noisy measurement of the signal $\mathbf{x}$, can we recover its analysis representation $\mathbf{\Omega x}$? Clearly when we have an exact measurement of $\mathbf{x}$ this becomes trivial. However, if we add noise to the measurements, we arrive at an estimation process which we name *analysis sparse approximation.* We describe two algorithms for this in the next section.

The second interesting question concerns dictionary learning: given a set of noisy training examples coming from an analysis sparse model, can we estimate the underlying dictionary? We consider this question in Section 6.3, where we propose the *Analysis K-SVD* algorithm for analysis dictionary training. Initial experiment results presented in Section 6.4 demonstrate the ability of our algorithm to recover underlying analysis structures in both synthetic and natural signal data.

6.2 Analysis Sparse Approximation

In the analysis framework, computing signal representations $\boldsymbol{\gamma}(\mathbf{x}) = \boldsymbol{\Omega}\mathbf{x}$ is a remarkably simple process. However, if we assume some contamination in the signal, recovering its analysis representation from the noisy measurements $\mathbf{y} = \mathbf{x} + \mathbf{n}$ becomes a non-trivial optimization task, which takes the form:

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\text{Argmin}} \;\; \|\mathbf{y} - \mathbf{z}\|_2 \quad \text{Subject To} \quad \|\boldsymbol{\Omega}\mathbf{z}\|_0 \leq L - P \;. \tag{6.1}$$

Here we assume that the sparsity of the original signal is known, and thus the optimization process searches for P-sparse signals in the vicinity of $\mathbf{y}$ which minimize the representation error. We refer to this problem, as well as to its error-constrained equivalent, as the *analysis sparse approximation* problem.

Similar to the synthesis sparse approximation problem, problem (6.1) is combinatorial in nature and can thus only be approximated. One approach to approximating the solution is to relax the ℓ^0 norm and replace it with some ℓ^p penalty function with $p > 0$, producing

$$\hat{\mathbf{z}} = \underset{\mathbf{z}}{\text{Argmin}} \;\; \|\mathbf{y} - \mathbf{z}\|_2 \quad \text{Subject To} \quad \|\boldsymbol{\Omega}\mathbf{z}\|_p^p \leq L - P \;. \tag{6.2}$$

This approach is parallel to the basis pursuit approach for synthesis approximation [12], and the resulting problem may be solved e.g., via an iterated re-weighted least squares (IRLS) method. Going to $p = 1$ results in the ℓ^1 analysis approximation problem, which is solvable using a variety of algorithms (Section 1.2.2).

A second approach, parallel to the synthesis greedy pursuit approaches [11, 38], suggests selecting rows from $\boldsymbol{\Omega}$ one-by-one in a greedy fashion. The process begins by setting $\mathbf{z} = \mathbf{y}$ and initializing an empty set of rows. Each iteration, the inner products $\boldsymbol{\Omega}\mathbf{z}$ are computed, and the row with the *smallest* non-zero inner product

Algorithm 6.1 Analysis-OMP

```
Input: Dictionary $\mathbf{\Omega} \in \mathbb{R}^{L \times N}$, signal $\mathbf{y} \in \mathbb{R}^N$, target sparsity $P$
Output: Signal $\mathbf{z} \in \mathbb{R}^N$ satisfying $\|\mathbf{\Omega z}\|_0 \leq L - P$ and minimizing $\|\mathbf{y} - \mathbf{z}\|_2$
Init: Set $\Phi := \emptyset$, $\Psi := \{1, 2, \ldots L\}$, $\mathbf{z} := \mathbf{y}$
for $i = 1 \ldots P$ do
    $\hat{k} := \operatorname*{Argmin}_{k \in \Psi} |\, \mathbf{w}_k^T \mathbf{z} \,|$
    $\Phi := \Phi \cup \{\, \hat{k} \,\}$
    $\Psi := \Psi \setminus \{\, \hat{k} \,\}$
    $\mathbf{z} := \mathbf{y} - (\mathbf{\Omega}_\Phi)^+ \mathbf{\Omega}_\Phi \mathbf{y}$
end for
return $\mathbf{z}$
```

is selected and added to the set. The solution $\mathbf{z}$ is then updated by projecting $\mathbf{y}$ on the orthogonal space of the selected rows. This process is repeated until the target sparsity (or error) is achieved. We refer to this method as *Analysis-OMP*, and detail it in Algorithm 6.1.

We compare the two options in Fig. 6.1. The plot shows the results of a synthetic experiment comparing the fraction of correctly recovered vanishing coefficients for a few noise and sparsity levels. As can be seen, in all cases the recovery performance improves with the sparsity of the signal, as could be expected. Among the two options, we see that their performance is mostly comparable, with a small advantage to the relaxation approach with $p \leq 0.5$. However, we note that the Analysis OMP is a much simpler and faster option, and thus we generally prefer it over the relaxation alternative.

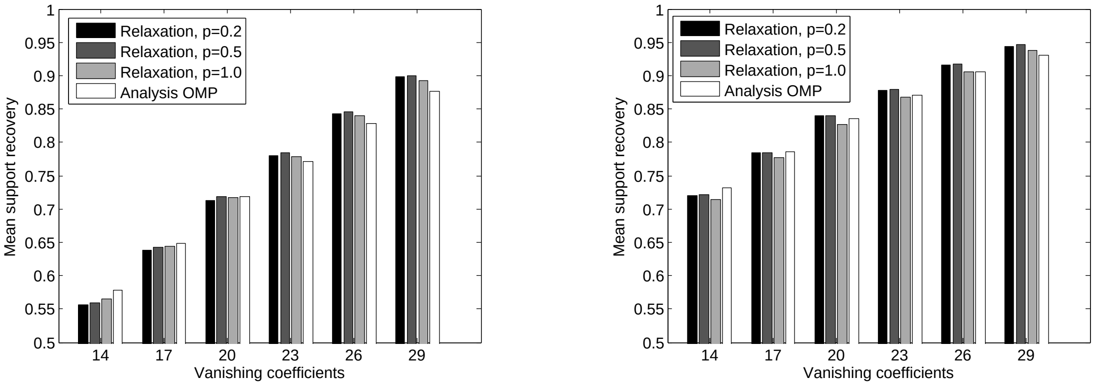


Figure 6.1: Analysis sparse approximation performance. The bars show the fraction of correctly recovered vanishing coefficients for different sparsity levels, using the relaxation ($p = 0.2, 0.5, 1$) and Analysis OMP algorithms. In this experiment, $\mathbf{\Omega}$ is of size 40×30 and contains random Gaussian entries. For each sparsity level P, we generate a test set of $1,000$ random signals, each orthogonal to P different rows in $\mathbf{\Omega}$, and contaminated with white Gaussian noise. The bars show the mean recovery rate achieved for each sparsity level, for noise levels of SNR=8dB (left) and SNR=15dB (right).

6.3 Dictionary Training

We now turn to the question of dictionary learning. Our goal is to assess the possibility of recovering an underlying analysis dictionary $\mathbf{\Omega}$ given a set of analysis-sparse realizations. We therefore consider the following setting: given a set of examples $\mathbf{Y} = [\mathbf{y}_1 \; \mathbf{y}_2 \; \ldots \; \mathbf{y}_R]$, we assume each example is a noisy version of a signal orthogonal to P rows from the unknown dictionary $\mathbf{\Omega}$. Thus, $\mathbf{y}_i = \mathbf{x}_i + \mathbf{n}_i$, where $\mathbf{n}_i$ is additive Gaussian noise, and $\mathbf{x}_i$ satisfies $\|\mathbf{\Omega}\mathbf{x}_i\|_0 = L - P$. Our goal is to find the dictionary $\mathbf{\Omega}$ giving rise to these signals, which can be translated to the following optimization task:

$$\begin{aligned} \underset{\mathbf{\Omega},\mathbf{Z}}{\text{Argmin}} \;\; \|\mathbf{Y}-\mathbf{Z}\|_F^2 \;\; \text{Subject To} \;\; & \forall i \;\; \|\mathbf{\Omega}\mathbf{z}_i\|_0 \le L - P \\ & \forall j \;\; \|\mathbf{w}_j\|_2 = 1 \; . \end{aligned} \tag{6.3}$$

Here, $\mathbf{z}_i$ are our estimates of the noiseless signals, arranged as the columns of the matrix $\mathbf{Z}$. The vectors $\mathbf{w}_j$ denote the rows of $\mathbf{\Omega}$ (as column vectors). The normalization constraint on the rows of $\mathbf{\Omega}$ is introduced to avoid degeneracy, but has no other influence on the result. We note that our formulation closely follows the structure of the ℓ^0 synthesis training problem, given by [28]:

$$\underset{\mathbf{D},\mathbf{\Gamma}}{\text{Argmin}} \ \|\mathbf{Y}-\mathbf{D}\mathbf{\Gamma}\|_F^2 \quad \text{Subject To} \quad \forall i \ \ \|\boldsymbol{\gamma}_i\|_0 \le T \qquad (6.4)$$
$$\forall j \ \ \|\mathbf{d}_j\|_2 = 1 \ .$$

Problem (6.3) is highly non-convex, and thus we cannot hope for a global solution. The optimization scheme we adopt here assumes an initial estimate $\mathbf{\Omega}_0$ of the analysis operator, and is based on a two-phase block-coordinate-relaxation approach, similar to the MOD [23] and K-SVD [28]. In the first phase we optimize for $\mathbf{Z}$ while keeping $\mathbf{\Omega}$ fixed, and in the second phase we update $\mathbf{\Omega}$ using the computed signals $\mathbf{Z}$. The process repeats until some stopping criterion (typically a fixed number of iterations) is achieved.

Optimizing for $\mathbf{Z}$ is done independently for each of its columns $\mathbf{z}_i$, defining a set of ℓ^0 analysis denoising problems which may be solved using any one of the sparse approximation methods:

$$\hat{\mathbf{z}}_i \ = \ \underset{\mathbf{z}_i}{\text{Argmin}} \ \|\mathbf{y}_i - \mathbf{z}_i\|_2 \quad \text{Subject To} \quad \|\mathbf{\Omega}\mathbf{z}_i\|_0 \le L-P \ . \qquad (6.5)$$

Once this step is complete, $\mathbf{\Omega}$ and $\mathbf{Z}$ are updated simultaneously in the second step. The optimization is carried out sequentially for each of the rows $\mathbf{w}_j$ in $\mathbf{\Omega}$. We note that the update of $\mathbf{w}_j$ only affects those columns of $\mathbf{Z}$ which are orthogonal to it, while the remaining columns are indifferent to the update (they may only gain from it). Thus, letting $\mathbf{Z}_J$ denote the submatrix of $\mathbf{Z}$ containing the columns orthogonal to $\mathbf{w}_j$, and denoting by $\mathbf{Y}_J$ the corresponding submatrix of $\mathbf{Y}$, the update step for $\mathbf{w}_j$ can be written as:

$$\underset{\mathbf{w}_j,\mathbf{Z}_J}{\text{Arg}\min} \ \|\mathbf{Y}_J - \mathbf{Z}_J\|_F^2 \quad \text{Subject To} \quad \forall i \in J \, , \ \ \|\mathbf{\Omega}\mathbf{z}_i\|_0 \le L-P \qquad (6.6)$$
$$\|\mathbf{w}_j\|_2 = 1 \ .$$

The straightforward approach to maintaining the sparsity constraints on the analysis representations is to force each $\mathbf{z}_i$ to remain orthogonal to the rows in $\mathbf{\Omega}$

it is already orthogonal to. This is parallel to the K-SVD atom update process where the representation supports are kept fixed. To formalize this, we use the notation $\mathbf{\Omega}^i$ to denote the submatrix of $\mathbf{\Omega}$ containing the rows which $\mathbf{z}_i$ is currently orthogonal to, *excluding* $\mathbf{w}_j$. This leads to the optimization task:

$$\begin{aligned} \underset{\mathbf{w}_j,\mathbf{Z}_J}{\text{Arg}\min}\ \|\mathbf{Y}_J - \mathbf{Z}_J\|_F^2 \quad \text{Subject To} \quad & \forall i \in J\,, \quad \mathbf{\Omega}^i \mathbf{z}_i = 0 \\ & \mathbf{w}_j^T \mathbf{Z}_J = 0 \\ & \|\mathbf{w}_j\|_2 = 1 \ . \end{aligned} \tag{6.7}$$

However, solving this problem directly turns out to be a difficult task. As we show in the Appendix, the atom update process resulting from this expression is given by

$$\underset{\mathbf{w}_j}{\text{Argmin}} \sum_{i \in J} \frac{\mathbf{w}_j^T \mathbf{y}_i^\perp (\mathbf{y}_i^\perp)^T \mathbf{w}_j}{\mathbf{w}_j^T \mathbf{P}_i \mathbf{w}_j} \quad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ ,$$

where $\mathbf{P}_i$ is the orthogonal projection operator on the null-space of $\mathbf{\Omega}^i$. As can be seen, in contrast to the K-SVD, this problem is difficult to optimize, and has no closed-form solution.

We therefore take a different route here. Rather than fix *all* current orthogonality relations, as suggested by the K-SVD-like path, we return to the original problem statement (6.6), and require only orthogonality to $\mathbf{w}_j$ to be maintained. The modified update process is thus given by:

$$\begin{aligned} \underset{\mathbf{w}_j,\mathbf{Z}_J}{\text{Arg}\min}\ \|\mathbf{Y}_J - \mathbf{Z}_J\|_F^2 \quad \text{Subject To} \quad & \forall i \in J\,, \quad \|\mathbf{\Omega}_{\widehat{j}} \mathbf{z}_i\|_0 \le L - P + 1 \\ & \mathbf{w}_j^T \mathbf{Z}_J = 0 \\ & \|\mathbf{w}_j\|_2 = 1 \ , \end{aligned} \tag{6.8}$$

where $\mathbf{\Omega}_{\widehat{j}}$ is the analysis operator without the j-th row.

To solve this problem, we adopt a *projected-optimization* approach. In the first step, we *relax* (6.8) by optimizing for $\mathbf{w}_j$ and $\mathbf{Z}_J$ *without the first constraint.* In the second step, we project $\mathbf{Z}_J$ back to the feasible domain, reinstating the constraints. For the first step we thus obtain a simple task:

$$\underset{\mathbf{w}_j,\,\mathbf{Z}_J}{\text{Arg}\min}\ \ \|\mathbf{Y}_J - \mathbf{Z}_J\|_F^2 \ \ \text{Subject To} \ \ \begin{aligned} &\mathbf{w}_j^T\mathbf{Z}_J = 0 \\ &\|\mathbf{w}_j\|_2 = 1 \ . \end{aligned} \tag{6.9}$$

This problem is a standard rank-reduction problem for $\mathbf{Y}_J$, and its solution is given by the rank-$(N-1)$ matrix $\mathbf{Z}_J$ closest to $\mathbf{Y}_J$, and its null-space $\mathbf{w}_j$. The updated $\mathbf{w}_j$ is thus the left singular vector corresponding to the *smallest* singular value of $\mathbf{Y}_J$, which can be computed from the SVD of $\mathbf{Y}_J$, or using a more efficient inverse power method. The second (projection) step is given by:

$$\underset{\mathbf{Z}_J}{\text{Argmin}}\ \ \|\mathbf{Y}_J - \mathbf{Z}_J\|_F^2 \ \ \text{Subject To} \ \ \begin{aligned} &\forall i \in J\,, \ \ \|\mathbf{\Omega}_{\widehat{j}}\mathbf{z}_i\|_0 \le L - P + 1 \\ &\mathbf{w}_j^T\mathbf{Z}_J = 0 \ , \end{aligned} \tag{6.10}$$

which can be solved by an analysis sparse approximation method.

We now note that the update step for $\mathbf{w}_j$, as suggested by (6.9), depends only on the input signals $\mathbf{Y}_J$, and not on the denoised signals $\mathbf{X}_J$. This suggests a *parallel update step* for the dictionary atoms, in which each row is independently set to the singular vector defined by its associated set of examples. Following the row updates, the projection steps are *bypassed* by continuing directly to the sparse-coding stage, which restores the sparsity constraints. Adopting this approach, we can thus rewrite the atom update (6.9) as

$$\hat{\mathbf{w}}_j = \underset{\mathbf{w}_j}{\text{Argmin}}\ \ \|\mathbf{w}_j^T\mathbf{Y}_J\|_2^2 \ \ \text{Subject To} \ \ \|\mathbf{w}_j\|_2 = 1 \ , \tag{6.11}$$

whose solution coincides with that of (6.9) for $\mathbf{w}_j$, but avoids computing $\mathbf{Z}_J$.

The complete training process thus alternates analysis sparse-approximation steps for the columns of $\mathbf{Z}$ (Eq. (6.5)), and SVD-based atom updates for the rows of $\mathbf{\Omega}$ (Eq. (6.11)). We name the resulting process *Analysis K-SVD*, due to its resemblance to the original K-SVD algorithm, and specifically, the similar use of K SVD processes per iteration (K representing the number of dictionary atoms). The full algorithm is detailed in Algorithm 6.2.

Algorithm 6.2 Analysis K-SVD

```
Input: Training signals $\mathbf{Y} \in \mathbb{R}^{N \times R}$, initial dictionary $\mathbf{\Omega}_0 \in \mathbb{R}^{L \times N}$, target sparsity $P$, number of iterations $k$
Output: Dictionary $\mathbf{\Omega}$ and signal set $\mathbf{Z}$ minimizing (6.3)
Init: Set $\mathbf{\Omega} := \mathbf{\Omega}_0$
for $n = 1 \ldots k$ do
    $\forall i: \quad \mathbf{z}_i := \underset{\mathbf{z}}{\mathrm{Argmin}} \, \|\mathbf{y}_i - \mathbf{z}|_2^2 \quad \text{Subject To} \quad \|\mathbf{\Omega z}\|_0 \le L - P$
    for all $j \in \{1 \ldots L\}$ do
        $J := \{$indices of the columns of $\mathbf{Z}$ orthogonal to $\mathbf{w}_j\}$
        $\mathbf{w}_j := \underset{\mathbf{w}}{\mathrm{Argmin}} \, \|\mathbf{w}^T \mathbf{Y}_J\|_2 \quad \text{Subject To} \quad \|\mathbf{w}\|_2 = 1$
        $\mathbf{\Omega}\{j\text{-th row}\} := \mathbf{w}_j^T$
    end for
end for
```

We mention that as an alternative to the parallel approach, a *serial* update of the dictionary atoms could also be considered, in which case the projection steps (6.10) are applied between the atom updates. These projections result in a re-assignment of the examples to the dictionary atoms, which affects subsequent atom updates by modifying the subsets of examples used. We have not explored the serial option in this work, however, due to its substantial computational cost.

6.4 Simulation Results

In the remainder of this chapter we present experiment results with the proposed training algorithm. In the first part we provide results for synthetic signals, demonstrating the ability of the method to recover a true underlying operator $\mathbf{\Omega}$ given a sparse training set. In the second part we show qualitative training results for natural image data, and observe the emergence of meaningful structures in the trained dictionary, indicating the ability of the algorithm to capture fundamental behaviors in natural signals.

6.4.1 Synthetic Experiments

To demonstrate the performance of the proposed algorithm in recovering an underlying dictionary $\mathbf{\Omega}$, we performed a set of synthetic experiments with a known ground-truth. We designed the experiments to follow those carried out in [86], which similarly targets the ℓ^0 analysis model, in order to allow a direct comparison. In these experiments, a known dictionary $\mathbf{\Omega} \in \mathbb{R}^{L \times N}$ is generated with random Gaussian entries. R sparse examples are then generated from this dictionary, and the training algorithm is employed to produce an estimate of the original dictionary. Each example is generated as a sparse analysis signal by randomly selecting P rows from $\mathbf{\Omega}$, computing their null-space, and sampling a random Gaussian vector within this null-space. The sparse signals are normalized to unit energy, and optionally subjected to additive white Gaussian noise, to produce the final training set.

In the following experiments, the analysis dictionary is of size 20×10, the sparsity level (number of vanishing coefficients) is $P = 8$, and the training set size is $R = 10,000$, in line with [86]. In the first set of simulations the training signals are noiseless, and in the second set we add noise with standard deviation $\sigma = 0.1/\sqrt{N}$ (SNR≈20dB). For each of the two cases, we run the algorithm with five different sets of inputs, to verify consistency of the results. The initial dictionary $\mathbf{\Omega}_0$ for the training is constructed in all cases by randomly selecting L sets of $N-1$ examples and computing their 1-dimensional null-spaces.

Fig. 6.2 shows representative training results for the noiseless and noisy cases, using Analysis-OMP for the sparse coding. As can be seen, in the noiseless case the algorithm fully recovers the original dictionary, while in the noisy case the algorithm recovers 19 out of 20 atoms (95%). Over the five noiseless experiments, the algorithm recovered $\mathbf{\Omega}$ with complete accuracy (machine level accuracy, MSE$< 10^{-30}$) in all executions. In the noisy case, the algorithm recovered 19 out of 20 atoms in 4 of the 5 executions, producing an MSE< 0.005, and in the remaining execution it recovered 17 of the 20 atoms, producing an MSE$= 0.012$. In comparison, [86] reports an accuracy of 10^{-8} for the noiseless case and an accuracy of 10^{-4} for the noisy case.

For the noisy case, we performed an additional set of experiments using the relaxation sparse approximation approach (6.2) with $p = 0.5$. Since the noise level is known, we optimize the error-constrained version of the sparse-coding problem in step 5 of the training algorithm:

$$\hat{\mathbf{z}}_i = \operatorname*{Argmin}_{\mathbf{z}_i} \ \|\mathbf{\Omega}\mathbf{z}_i\|_p^p \quad \text{Subject To} \quad \|\mathbf{z}_i - \mathbf{y}_i\|_2 \leq \epsilon \ . \tag{6.12}$$

In this case, however, the denoised signal $\hat{\mathbf{z}}_i$ is not expected to be precisely orthogonal to any of the rows in $\mathbf{\Omega}$. Thus, to achieve exact P-sparsity, we subsequently detect for each estimate $\hat{\mathbf{z}}_i$ the P rows in $\mathbf{\Omega}$ it is most orthogonal to (i.e., has minimal correlation with), and use this set $\mathbf{\Phi}$ to define the final denoised result as $\hat{\mathbf{z}}_i = \mathbf{y}_i - \mathbf{\Omega}_\Phi^+ \mathbf{\Omega}_\Phi \mathbf{y}_i$.

Using the relaxation approach improves recovery results in the noisy case, producing an accurate estimate of $\mathbf{\Omega}$ in 2 out of 5 executions (100% of the atoms recovered, MSE$<2 \cdot 10^{-5}$), and recovering 19 of 20 atoms in the remaining three execution (MSE$<$0.005). Fig. 6.3 shows an example result. As can be seen, the method smoothes the penalty function behavior and the convergence to the true dictionary. Optimizing the relaxed target function, however, is significantly slower than the greedy alternative.

6.4.2 Experiments with Natural Images

We now present experiment results with natural image patches, aiming to qualitatively evaluate the behavior of the training algorithm on real image data. For these experiments, we randomly extracted $5,000$ 8×8 image patches from each of

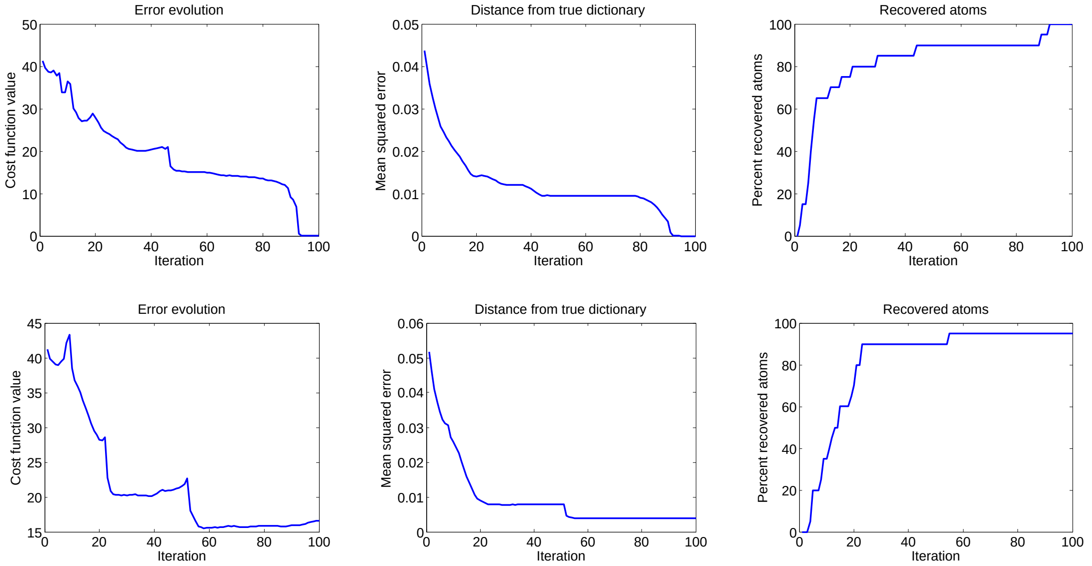


Figure 6.2: Example training results for a noiseless case (top row) and noisy case (bottom row). Left to right: penalty function value ($\|\mathbf{Y} - \mathbf{Z}\|_F$), distance to true dictionary (MSE), percent of recovered atoms. All plots show evolution over algorithm iterations.

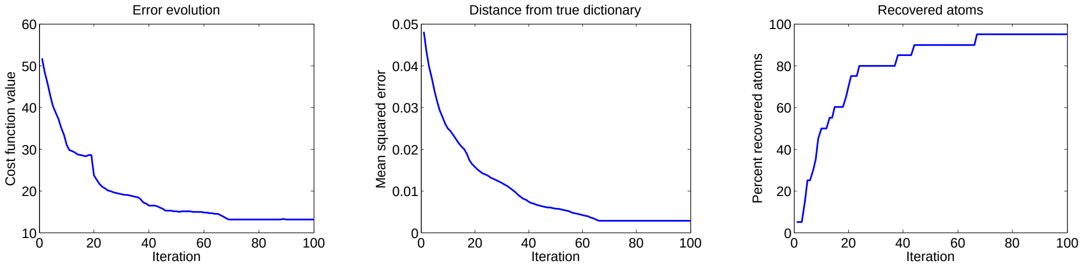


Figure 6.3: Example training result for a noisy case, using the relaxation sparse-coding technique with $p = 0.5$. Left to right: penalty function value, distance to true dictionary (MSE), percent of recovered atoms.

five standard test images (Fig. 6.4), for a total of $25,000$ training signals. We then applied the Analysis K-SVD on these signals, using Analysis-OMP sparse-coding, to train dictionaries of size 100×64 using varying levels of sparsity P. We note that as opposed to many synthesis training methods, in the analysis case it is not necessary to remove the mean (DC) from the training signals in a preprocessing step, as the training target tends towards zero-mean atoms by construction.

Training results for the sparsity levels of $P = 16, 32$ are presented in Fig. 6.5. In both cases the algorithm was executed for 100 training iterations. As can be seen, the Analysis K-SVD algorithm efficiently reduces the penalty function in both cases, achieving much of the reduction in the first few iterations. The resulting trained atoms capture high-frequency signal characteristics, which are known to be sparse in natural images (see e.g., [9]). For the higher sparsity level, we see the formation of more localized and oriented structures in the analysis atoms, reminiscent of Gabor and wavelet-type filters. Such properties are fundamental in sparsifying transforms, as discussed in [13]. Interestingly, the atoms of the *pseudo-inverse* dictionary $\mathbf{D} = \mathbf{\Omega}^+$ bear some visual resemblance to the K-SVD synthesis atoms, as can be found in [28]. We can reasonably assume that this phenomenon is related to the similarity between the two algorithms, though the details of the relationship remain to be studied.

6.5 Conclusions

The ℓ^0 analysis model is an intriguing new signal model motivated by ideas from ℓ^0 synthesis models, natural image statistics, and insights from the ℓ^1 analysis model. In this work we presented two methods for approximating the analysis sparse-coding problem, and developed an efficient algorithm for analysis dictionary training. Our training algorithm shares much of the structure of the K-SVD synthesis training algorithm, with the replacement of a maximum eigenvalue problem with a minimum eigenvalue one. We have shown that our training method

Figure 6.4: Test images used to generate the training set for Analysis K-SVD.

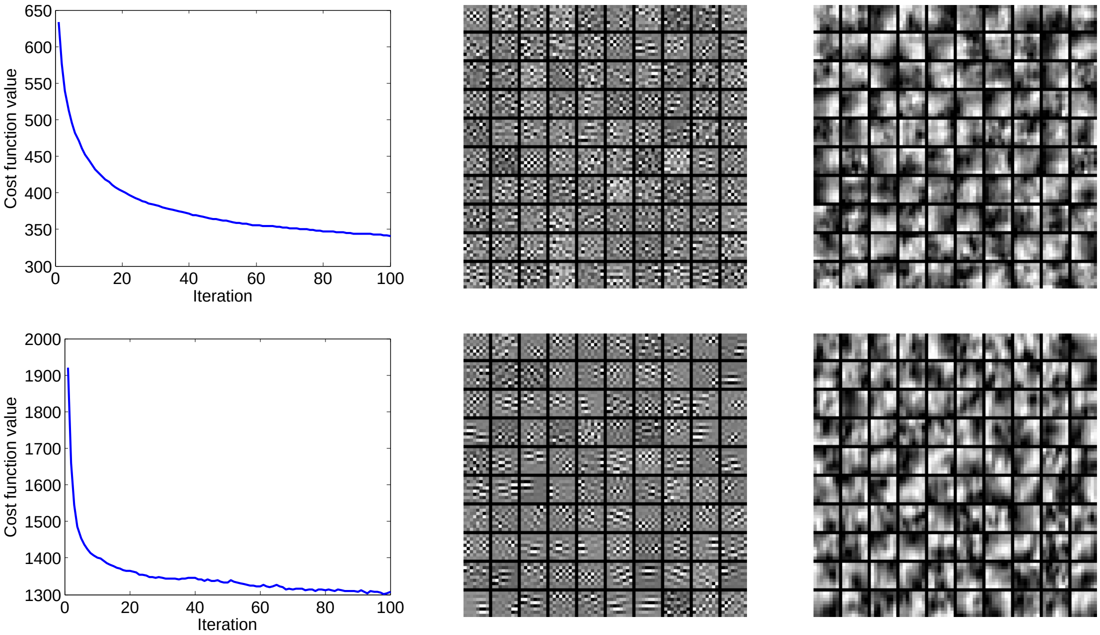


Figure 6.5: Training results for natural image patches using sparsity levels of $P = 16$ (top row) and $P = 32$ (bottom row). Left to right: convergence of the target function, atoms of the trained dictionary, and atoms of the dictionary pseudo-inverse.

is able to effectively minimize the ℓ^0 analysis target, and successfully recover an underlying model given data examples. We have also shown training results for natural image data, where the learned dictionary exhibited localized and oriented behavior, known to characterize natural images.

Our work is an initial effort which opens the door to many future research directions. Clearly, uniqueness theorems as well as formal success bounds for the ℓ^0 pursuit algorithms are highly desirable, similar to the vast literature on ℓ^0 synthesis models. Some work in this direction has already commenced, with uniqueness results obtained for a related ℓ^0 analysis formulation [177]. Practical applications in the fields of image recovery, understanding, compression and analysis remain to be explored. Finally, recent trends in dictionary design, which tend towards more structured and robust dictionary forms [13], could be implemented in the analysis framework as well. Many of these new structures have natural extensions to the analysis framework, and exploring the benefits and properties of such structures in the context of analysis models is an interesting future research goal.

6.A Formulation of Explicit Atom Update

In this appendix we derive the explicit form of the atom update problem:

$$\begin{aligned} \operatorname*{Arg\,min}_{\mathbf{w}_j,\,\mathbf{Z}_J} \ \|\mathbf{Y}_J - \mathbf{Z}_J\|_F^2 \ \text{ Subject To } \ & \forall i \in J\,, \ \ \mathbf{\Omega}^i \mathbf{z}_i = 0 \\ & \mathbf{w}_j^T \mathbf{Z}_J = 0 \\ & \|\mathbf{w}_j\|_2 = 1 \ , \end{aligned} \tag{6.13}$$

by eliminate $\mathbf{Z}_J$ from the optimization process and expressing the task as an optimization problem for $\mathbf{w}_j$ alone.

To achieve this, we begin by computing a closed-form expression for the dependence $\mathbf{Z}_J(\mathbf{w}_j)$ for any $\mathbf{w}_j$. We note that given $\mathbf{w}_j$, the optimization can be carried out separately for each column $\mathbf{z}_i \in \mathbf{Z}_J$:

$$\begin{aligned} \operatorname*{Argmin}_{\mathbf{z}_i} \ \|\mathbf{y}_i - \mathbf{z}_i\|_2^2 \ \text{ Subject To } \ & \mathbf{\Omega}^i \mathbf{z}_i = 0 \\ & \mathbf{w}_j^T \mathbf{z}_i = 0 \ . \end{aligned} \tag{6.14}$$

The solution to this problem is the projection of $\mathbf{y}_i$ on the space orthogonal to the rows of $\mathbf{\Omega}^i \in \mathbb{R}^{L_i \times N}$ and the atom $\mathbf{w}_j$. To express this analytically, we let $\mathbf{W}_i \in \mathbb{R}^{N \times \tilde{L}_i}$ be a matrix whose columns orthonormally span the row-space of $\mathbf{\Omega}^i$ (note that $\tilde{L}_i \leq L_i$ with an inequality if the rows of $\mathbf{\Omega}^i$ are linearly dependent), and similarly let the columns of $\mathbf{V}_i = \mathbf{W}_i^{\perp} \in \mathbb{R}^{N \times (N - \tilde{L}_i)}$ span the orthogonal space. Note that $\operatorname{span}\{\mathbf{W}_i\} \oplus \operatorname{span}\{\mathbf{V}_i\} = \mathbb{R}^N$.

The training signal $\mathbf{y}_i$ has a unique decomposition as $\mathbf{y}_i = \mathbf{W}_i\boldsymbol{\alpha}_i + \mathbf{V}_i\boldsymbol{\beta}_i = \mathbf{y}_i^{\|} + \mathbf{y}_i^{\perp}$, with $\mathbf{y}_i^{\|}$ denoting the component of $\mathbf{y}_i$ spanned by the rows of $\boldsymbol{\Omega}^i$, and $\mathbf{y}_i^{\perp}$ denoting the orthogonal component. Now, if we initially ignore the constraint $\mathbf{w}_j^T\mathbf{z}_i = 0$, the solution to (6.14) is clearly $\mathbf{z}_i = \mathbf{y}_i^{\perp}$, the component of $\mathbf{y}_i$ orthogonal to the row-span of $\boldsymbol{\Omega}^i$. Reintroducing the constraint $\mathbf{w}_j^T\mathbf{z}_i = 0$, it is easy to see that the solution can be computed within the subspace $\mathbf{V}_i$, by projecting $\mathbf{w}_j$ onto this subspace and orthogonalizing $\mathbf{y}_i^{\perp}$ in respect to the projected atom. Specifically, since the solution must be spanned by $\mathbf{V}_i$, it can be written as $\mathbf{z}_i = \mathbf{V}_i\boldsymbol{\gamma}_i$, leading to the minimization:

$$\underset{\boldsymbol{\gamma}_i}{\text{Argmin}} \;\; \|\mathbf{W}_i\boldsymbol{\alpha}_i + \mathbf{V}_i\boldsymbol{\beta}_i - \mathbf{V}_i\boldsymbol{\gamma}_i\|_2^2 \quad \text{Subject To} \quad \mathbf{w}_j^T\mathbf{V}_i\boldsymbol{\gamma}_i = 0 \ .$$

From the orthogonality of $\mathbf{W}_i$ and $\mathbf{V}_i$, and utilizing the fact that $\mathbf{W}_i\alpha_i$ does not affect the minimization, the above reduces to

$$\underset{\boldsymbol{\gamma}_i}{\text{Argmin}} \;\; \|\boldsymbol{\beta}_i - \boldsymbol{\gamma}_i\|_2^2 \quad \text{Subject To} \quad (\mathbf{V}_i^T\mathbf{w}_j)^T\boldsymbol{\gamma}_i = 0 \ .$$

The solution to this problem is obtained by orthogonalizing $\boldsymbol{\beta}_i$ in respect to $\mathbf{V}_i^T\mathbf{w}_j$, i.e.,

$$\boldsymbol{\gamma}_i = \boldsymbol{\beta}_i - \frac{(\mathbf{V}_i^T\mathbf{w}_j)^T\boldsymbol{\beta}_i}{\|\mathbf{V}_i^T\mathbf{w}_j\|_2^2}\mathbf{V}_i^T\mathbf{w}_j \ .$$

Finally, recalling that $\mathbf{z}_i = \mathbf{V}_i\boldsymbol{\gamma}_i$, the solution of (6.14) is thus given by

$$\begin{aligned} \mathbf{z}_i &= \mathbf{V}_i\boldsymbol{\gamma}_i = \mathbf{V}_i\boldsymbol{\beta}_i - \frac{(\mathbf{V}_i^T\mathbf{w}_j)^T\boldsymbol{\beta}_i}{\|\mathbf{V}_i^T\mathbf{w}_j\|_2^2}\mathbf{V}_i\mathbf{V}_i^T\mathbf{w}_j \\ &= \mathbf{y}_i^{\perp} - \frac{\mathbf{w}_j^T\mathbf{y}_i^{\perp}}{\mathbf{w}_j^T\mathbf{V}_i\mathbf{V}_i^T\mathbf{w}_j}\mathbf{V}_i\mathbf{V}_i^T\mathbf{w}_j \\ &= \mathbf{y}_i^{\perp} - \frac{\mathbf{w}_j^T\mathbf{y}_i^{\perp}}{\mathbf{w}_j^T\mathbf{P}_i\mathbf{w}_j}\mathbf{P}_i\mathbf{w}_j \ . \end{aligned} \tag{6.15}$$

Here, we denoted by $\mathbf{P}_i = \mathbf{V}_i\mathbf{V}_i^T$ the projection operation on the span of $\mathbf{V}_i$.

Eq. (6.15) defines the analytical solution for the $\mathbf{z}_i$'s given $\mathbf{w}_j$. We can now substitute this in (6.13) to obtain the following optimization problem for the atom $\mathbf{w}_j$:

$$\underset{\mathbf{w}_j}{\text{Argmin}} \;\; \sum_{i\in J}\left\|\mathbf{y}_i - \mathbf{y}_i^{\perp} + \left(\frac{\mathbf{w}_j^T\mathbf{y}_i^{\perp}}{\mathbf{w}_j^T\mathbf{P}_i\mathbf{w}_j}\right)\mathbf{P}_i\mathbf{w}_j\right\|_2^2 \quad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ .$$

Since $\mathbf{y}_i - \mathbf{y}_i^\perp = \mathbf{W}_i\boldsymbol{\alpha}_i$, it is constant in the optimization and orthogonal to the third term (which is spanned by $\mathbf{V}_i$ from the definition of $\mathbf{P}_i$). Thus we can reduce the minimization to:

$$\underset{\mathbf{w}_j}{\text{Argmin}} \quad \sum_{i\in J} \left\| \left(\frac{\mathbf{w}_j^T \mathbf{y}_i^\perp}{\mathbf{w}_j^T \mathbf{P}_i \mathbf{w}_j} \right) \mathbf{P}_i \mathbf{w}_j \right\|_2^2 \quad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ . \tag{6.16}$$

The target function in this minimization simplifies to:

$$\left\| \left(\frac{\mathbf{w}_j^T \mathbf{y}_i^\perp}{\mathbf{w}_j^T \mathbf{P}_i \mathbf{w}_j} \right) \mathbf{P}_i \mathbf{w}_j \right\|_F^2 = \frac{(\mathbf{w}_j^T \mathbf{y}_i^\perp)^2}{(\mathbf{w}_j^T \mathbf{P}_i \mathbf{w}_j)^2} \mathbf{w}_j^T \mathbf{P}_i^T \mathbf{P}_i \mathbf{w}_j = \frac{\mathbf{w}_j^T \mathbf{y}_i^\perp (\mathbf{y}_i^\perp)^T \mathbf{w}_j}{\mathbf{w}_j^T \mathbf{P}_i \mathbf{w}_j} \ , \tag{6.17}$$

where we used $\mathbf{P}_i^T\mathbf{P}_i = \mathbf{P}_i$ as $\mathbf{P}_i$ is a symmetric projection matrix.

Combining (6.17) with (6.16), the atom update (6.13) finally becomes:

$$\underset{\mathbf{w}_j}{\text{Argmin}} \quad \sum_{i\in J} \frac{\mathbf{w}_j^T \mathbf{y}_i^\perp (\mathbf{y}_i^\perp)^T \mathbf{w}_j}{\mathbf{w}_j^T \mathbf{P}_i \mathbf{w}_j} \quad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ . \tag{6.18}$$

It is interesting to note that in this process, two forces are simultaneously acting on $\mathbf{w}_j$: The first force comes from the numerator, which pushes $\mathbf{w}_j$ to be orthogonal to $\mathbf{y}_i^\perp$ — the current solution without $\mathbf{w}_j$; this force is expected. At the same time, the denominator tries to make $\mathbf{w}_j$ *orthogonal* to the rows in $\boldsymbol{\Omega}^i$, as a large denominator means $\mathbf{w}_j$ is close to the span of $\mathbf{V}_i$, which spans the complement space to the rows of $\boldsymbol{\Omega}^i$. Thus, we see that the ℓ^0 analysis problem naturally incorporates a regularizing force which aims to ”spread out” the rows in $\boldsymbol{\Omega}$.

Chapter 7

Learning Thresholding Dictionaries

Joint work with Michael Elad.

Abstract

Wavelet thresholding is a classical and widely used algorithm for signal denoising. This process decomposes a noisy signal over an orthogonal dictionary, eliminates the smallest coefficients, and applies the dictionary inverse to produce an estimate of the noiseless signal. More recently, the process has been extended to non-orthogonal *overcomplete* dictionaries, in which case the dictionary inverse is replaced by a pseudo-inverse. The use of overcomplete dictionaries improves estimation results for images and high-dimensional signal data, due to the ability of such dictionaries to better capture complex multi-dimensional signal behavior. Nonetheless, using fixed dictionaries in these processes remains a limiting factor on the recovery performance, due to the non-adaptive nature of generic transforms.

The incorporation of adaptive, trained dictionaries in thresholding methods has the potential of improving recovery performance by tailoring the dictionary to the specific signal data and estimation task. In this work we propose a framework for training dictionaries for thresholding-based recovery processes. We present a generalization of the basic thresholding framework which utilizes a *pair* of over-

complete dictionaries, and can be applied to a wider range of tasks. The two dictionaries are associated with the analysis and synthesis stages of the algorithm, and we thus name the process *analysis-synthesis thresholding.* The proposed training algorithm simultaneously trains both dictionaries given examples of origin and degraded signals, and requires no prior knowledge of the degradation model. Experiments with small-kernel image deblurring demonstrate the ability of our method to favorably compete with dedicated deconvolution processes, using a simple, stable, and fast recovery process.

7.1 Introduction

The shrinkage-based denoising process is based on a non-linear operation applied to each of the analysis coefficients of a noisy signal. Given the measured signal $\mathbf{y} = \mathbf{x} + \mathbf{n}$, this process is given by

$$\hat{\mathbf{x}} = \mathbf{\Omega}^{+} S_{\lambda}(\mathbf{\Omega}\mathbf{y}) \ , \tag{7.1}$$

where $S_{\lambda}(\cdot)$ is a scalar *shrink operator* governed by the parameter λ. As discussed in section 1.2.2, in the overcomplete case this estimator does not generally emerge as a solution to an analysis task of the form (1.13). Instead, it constitutes the formal solution to a *representation-domain* sparsification process, of the form

$$\hat{\mathbf{z}} = \mathbf{\Omega}^{+} \cdot \underset{\boldsymbol{\gamma}}{\operatorname{Argmin}} \ \|\boldsymbol{\gamma} - \mathbf{\Omega}\mathbf{y}\|_2^2 + \lambda C(\boldsymbol{\gamma}) \ , \tag{7.2}$$

with $C(\boldsymbol{\gamma})$ a suitably chosen separable penalty function.

A specific widely-used choice for S_{λ} is the *hard thresholding* operator, which applies a fixed threshold to each of the representation coefficients:

$$S_{\lambda}(\alpha) = \begin{cases} \alpha & |\alpha| \geq \lambda \\ 0 & |\alpha| < \lambda \end{cases} \ . \tag{7.3}$$

This operator is associated with the choice $C(\boldsymbol{\gamma}) = \|\boldsymbol{\gamma}\|_0$ in (7.2), and nullifies the smallest coefficients in $\mathbf{\Omega}\mathbf{y}$, essentially performing an ℓ^0 sparsification of the analysis coefficients. However, we note that whereas the ℓ^0 analysis formulation (6.1) seeks a signal $\hat{\mathbf{z}}$ whose analysis coefficients are truly ℓ^0-sparse, the shrinkage process above simply performs an unconstrained sparsification of the representation coefficients, followed by a projection of the result back to the feasible domain $\{\mathbf{\Omega x} \mid \mathbf{x} \in \mathbb{R}^N\} \subset \mathbb{R}^L$ through the dictionary pseudo-inverse.

The simplicity and efficiency of the thresholding operator make it an attractive technique for denoising. The dictionary $\mathbf{\Omega}$ is typically chosen to be an analytic dictionary such as wavelets [135], curvelets [77] or contourlets [18]. However, a desirable goal would be to *learn* the dictionary from actual data instances. In this work we focus on the specific case of ℓ^0 (hard) thresholding, where we exploit the resemblance to the analysis and synthesis ℓ^0 frameworks to develop a simple and efficient dictionary training technique. We apply our training method to an image deblurring application to demonstrate the effectiveness and usefulness of the proposed method.

We should mention that while dictionary training has not yet been addressed in the context of thresholding, a different aspect of this process — the scalar shrinking operator — was recently considered in [178]. Given a fixed dictionary $\mathbf{\Omega}$ and a set of training examples, an individual shrink operator S_i is learned for each of the dictionary atoms using a piecewise-linear approximation. An interesting outcome of this process, relevant to the current work, is the notable resemblance of the resulting shrinkage operators to the hard thresholding operator used here (though with a small and intriguing non-monotonicity around the center in some cases, see Fig. 8 there). Indeed, this is an encouraging result that demonstrates the potential usefulness of the hard thresholding operator in practical applications.

7.2 Analysis-Synthesis Thresholding

Reviewing the denoising process (7.1), we notice that it can be easily extended to handle more general recovery tasks by simply decoupling the analysis and synthesis dictionaries. Such a modification leads to a recovery process of the form:

$$\hat{\mathbf{x}} = \mathbf{D}S_\lambda(\mathbf{\Omega}\mathbf{y}) \ , \tag{7.4}$$

where $\mathbf{D} \in \mathbb{R}^{M\times L}$, $\mathbf{\Omega} \in \mathbb{R}^{L\times N}$, and $M \neq N$ in general. An added advantage of this decoupling is that it results in a simpler dictionary training task, due to the elimination of the pseudo-inverse constraint between the dictionaries. This decoupling of the dictionaries makes the process a true analysis-synthesis hybrid, and we thus name it *analysis-synthesis thresholding.*

An important point which must be addressed in any process of the type (7.4) is the choice of the threshold λ. Common threshold-selection processes include the SureShrink [179], VisuShrink [180], BayesShrink [181], K-Sigma shrink [182], and FDR-based shrink [183]. In this work , however, we adopt a *learning* approach in which the threshold is trained as part of the dictionary learning process. Indeed, the threshold value will generally depend on the noise level. In the following, we take a simplistic approach and train an individual triplet ($\mathbf{\Omega}$, $\mathbf{D}$, λ) for each noise level. In practice, it is likely that a single dictionary pair could be trained for all noise levels, adapting only the threshold value to each noise level individually using the proposed threshold training process.

7.3 Dictionary Training

7.3.1 Training Target

The process definition (7.4) naturally gives rise to a training formulation for the recovery parameters [184]. Given a set of training *pairs* $\{(\mathbf{x}_i, \mathbf{y}_i)\}$, representing origin signals $\mathbf{x}_i$ and their degraded versions $\mathbf{y}_i$, we wish to find a triplet $(\mathbf{D}, \mathbf{\Omega}, \lambda)$ which best recovers the $\mathbf{x}_i$'s from the $\mathbf{y}_i$'s. Letting $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_R]$ and $\mathbf{Y} = [\mathbf{y}_1\mathbf{y}_2 \ldots \mathbf{y}_R]$, the training process takes the form:

$$\{\,\hat{\mathbf{\Omega}}, \hat{\mathbf{D}}, \hat{\lambda}\,\} = \operatorname{Arg}\min_{\mathbf{\Omega},\mathbf{D},\lambda} \|\mathbf{X} - \mathbf{D}S_\lambda(\mathbf{\Omega}\mathbf{Y})\|_F^2 \ . \tag{7.5}$$

Note that this problem is actually defined up to a factor, as we are free to rescale $\mathbf{\Omega} \to (\alpha\mathbf{\Omega})$, $\mathbf{D} \to (1/\alpha\,\mathbf{D})$, $\lambda \to (\alpha\lambda)$ for any $\alpha > 0$. Thus, we could normalize this problem by selecting, e.g., $\lambda = 1$, and allowing the optimization process to set the norms of the rows in $\mathbf{\Omega}$ to fit this threshold. Alternatively, the normalization we choose here (mainly for simplicity of presentation) is to fix the norm of *each* row in $\mathbf{\Omega}$ to unit length, and allow the threshold to vary. Clearly, to accommodate such a normalization we must allow the threshold to differ for each row. Thus, we introduce individual thresholds λ_i for each of the rows in $\mathbf{\Omega}$, providing L degrees of freedom of the form $\mathbf{w}_i \to (\alpha_i\mathbf{w}_i)$, $\mathbf{d}_i \to (1/\alpha_i\,\mathbf{d}_i)$, $\lambda_i \to (\alpha_i\lambda_i)$ for $i = 1 \ldots L$. This allows us to set each α_i such that $\|\mathbf{w}_i\|_2 = 1$. Adopting this normalization, our training target becomes:

$$\begin{aligned}\{\,\hat{\mathbf{\Omega}}, \hat{\mathbf{D}}, \hat{\boldsymbol{\lambda}}\,\} \quad &= \quad \operatorname{Arg}\min_{\mathbf{\Omega},\mathbf{D},\boldsymbol{\lambda}} \|\mathbf{X} - \mathbf{D}S_{\boldsymbol{\lambda}}(\mathbf{\Omega}\mathbf{Y})\|_F^2 \\ & \quad\quad \text{Subject To} \quad \forall i \ \|\mathbf{w}_i\|_2 = 1 \ ,\end{aligned} \tag{7.6}$$

with $\boldsymbol{\lambda} = (\lambda_1, \ldots \lambda_L)$ constituting a *vector* of thresholds for the L atoms.

7.3.2 Optimization Scheme

We optimize (7.6) by adopting a sequential approach similar to the K-SVD and Analysis K-SVD algorithms. At the j-th step, we keep all but the j-th pair of atoms fixed, and optimize:

$$\begin{aligned}\{\,\hat{\mathbf{w}}_j, \hat{\mathbf{d}}_j, \hat{\lambda}_j\,\} \quad &= \quad \operatorname*{Argmin}_{\mathbf{w}_j,\mathbf{d}_j,\lambda_j} \|\mathbf{X} - \mathbf{D}S_{\boldsymbol{\lambda}}(\mathbf{\Omega}\mathbf{Y})\|_F^2 \\ & \quad\quad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ .\end{aligned} \tag{7.7}$$

Simplifying the cost function, we obtain:

$$\begin{aligned}\|\mathbf{X} - \mathbf{D}S_{\boldsymbol{\lambda}}(\mathbf{\Omega}\mathbf{Y})\|_F^2 \quad &= \quad \|\mathbf{X} - \sum_k \mathbf{d}_k S_{\lambda_k}(\mathbf{w}_k^T\mathbf{Y})\|_F^2 \\ &= \quad \|\mathbf{X} - \sum_{k\neq j} \mathbf{d}_k S_{\lambda_k}(\mathbf{w}_k^T\mathbf{Y}) - \mathbf{d}_j S_{\lambda_j}(\mathbf{w}_j^T\mathbf{Y})\|_F^2 \\ &= \quad \|\mathbf{E}_j - \mathbf{d}_j S_{\lambda_j}(\mathbf{w}_j^T\mathbf{Y})\|_F^2 \ ,\end{aligned}$$

with $\mathbf{E}_j = \mathbf{X} - \sum_{k \neq j} \mathbf{d}_k S_{\lambda_k}(\mathbf{w}_k^T \mathbf{Y})$. Thus, our optimization goal for the j-the atom pair becomes:

$$\begin{aligned} \{\, \hat{\mathbf{w}}_j, \hat{\mathbf{d}}_j, \hat{\lambda}_j \,\} \quad &= \quad \underset{\mathbf{w}_j, \mathbf{d}_j, \lambda_j}{\text{Argmin}} \; \|\mathbf{E}_j - \mathbf{d}_j S_{\lambda_j}(\mathbf{w}_j^T \mathbf{Y})\|_F^2 \\ & \quad\;\; \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \; . \end{aligned} \tag{7.8}$$

We note that the hard thresholding operator, controlled by $\mathbf{w}_j$ and λ_j, partitions the signals in $\mathbf{Y}$ to two sets, depending on their relation with the threshold. We denote the indices of the examples that survive the threshold ($|\mathbf{w}_j^T \mathbf{y}_i| \geq \lambda_j$) by $J = J(\mathbf{w}_j, \lambda_j)$, and split the matrix $\mathbf{Y}$ to the signals $\mathbf{Y}^J$ that survive the threshold, and the remaining signals $\mathbf{Y}^{\overline{J}}$. We similarly split $\mathbf{E}_j$ to the corresponding submatrices $\mathbf{E}_j^J$ and $\mathbf{E}_j^{\overline{J}}$. With these notations, the above can be rearranged as:

$$\begin{aligned} \{\, \hat{\mathbf{w}}_j, \hat{\mathbf{d}}_j, \hat{\lambda}_j \,\} \quad &= \quad \underset{\mathbf{w}_j, \mathbf{d}_j, \lambda_j}{\text{Argmin}} \; \|\mathbf{E}_j^J - \mathbf{d}_j \mathbf{w}_j^T \mathbf{Y}^J\|_F^2 + \|\mathbf{E}_j^{\overline{J}}\|_F^2 \\ & \quad\;\; \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \; . \end{aligned} \tag{7.9}$$

Obviously, minimizing this expression is an ambitious task, as the target function is non-convex and highly discontinuous. The main difficulty in the optimization is due to the fact that updating $\mathbf{w}_j$ and λ_j may modify the signal partitioning J, causing a non-smooth change to the cost function. One straightforward approach is thus to perform the update while constraining the partitioning of the signals to remain fixed. Under such a constraint, the atom update task can be formulated as a convex Quadratic Programming (QP) problem, and can be globally solved. Unfortunately, this approach can clearly accommodate only a small deviation of the solution from the initial estimate, and thus we take a different approach here. For completeness, we detail the derivation of the QP formulation in Appendix 7.A.

Optimization via Rank-One Approximation

A simple and surprisingly effective alternative to the constrained partitioning approach involves making the *approximation* that the update process does not change much the partitioning of the signals about the threshold. This approach assumes that the set J remains roughly constant during the update process, and thus, the target function in (7.9) can be approximated by the function

$$\operatorname{Arg}\min_{\mathbf{w}_j,\mathbf{d}_j,\lambda_j} \|\mathbf{E}_j^{J_0} - \mathbf{d}_j\mathbf{w}_j^T\mathbf{Y}^{J_0}\|_F^2 + \|\mathbf{E}_j^{\overline{J}_0}\|_F^2 \quad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ , \qquad (7.10)$$

with J_0 denoting the current partitioning of the signals. Formally, this approach is equivalent to optimizing (7.8) under a first-order expansion of S_{λ_j}, which is relatively accurate for coefficients far from the threshold.

Deriving a formal bound on the error of the proposed approximation is difficult: in fact, when the set J is small, the approximation becomes useless as the signal partitioning may substantially change by the update process. However, when the set J covers a significant enough portion of the examples, we expect the majority of the examples to follow this assumption due to the nature of the update which favors signals already using the atom. Our simulations support this assumption, indicating that the typical fraction of signals moving between J and $\overline{J}$ in practice is quite small. A representative case is provided in Fig. 7.1: as can be seen, the fraction of signals moving between J and $\overline{J}$ in this case is $< 12\%$ for the first iteration, and goes down to just $2-6\%$ for the remaining iterations. We see that in this case the proposed approximation is quite reliable, while at the same time leading to a substantially easier optimization goal.

By refraining from explicit constraints on the partitioning, we not only simplify the optimization problem, but also gain flexibility by allowing some outlier signals to "switch sides" relative to the threshold. Returning to the approximate optimization target (7.10), $E_j^{\overline{J}_0}$ in this formulation is now a constant in the optimization, and thus the update task reduces to:

$$\begin{aligned} \{\,\hat{\mathbf{w}}_j, \hat{\mathbf{d}}_j\,\} \quad &= \quad \operatorname{Arg}\min_{\mathbf{w}_j,\mathbf{d}_j} \|\mathbf{E}_j^{J_0} - \mathbf{d}_j\mathbf{w}_j^T\mathbf{Y}^{J_0}\|_F^2 \qquad (7.11) \\ &\qquad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ . \end{aligned}$$

Note that in this formulation λ_j is omitted, as J_0 is fixed and thus the value of λ_j

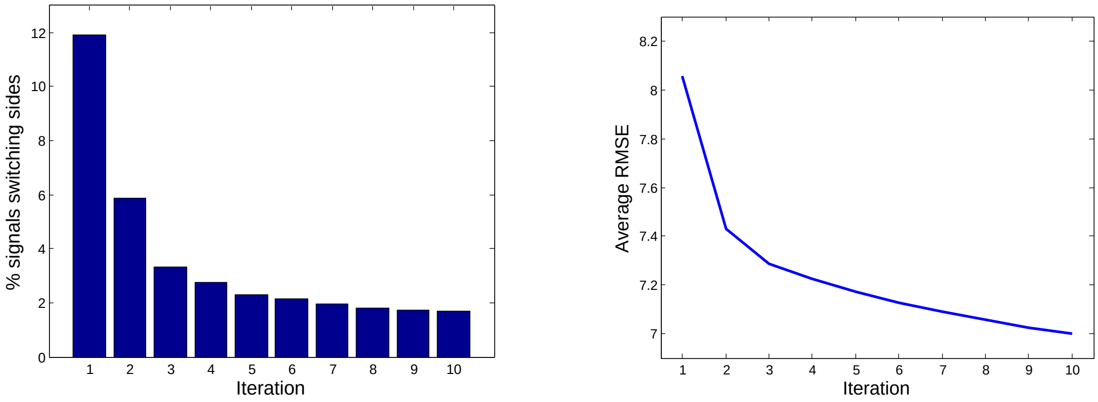


Figure 7.1: Fraction of signals changing sides relative to the threshold at different training iterations. The figure shows results for training a pair of thresholding dictionaries with 256 atoms each for denoising 8×8 image patches with a noise level of $\sigma = 10$. The training patches were extracted from eight arbitrary images in the CVG Granada [185] data set, $40,000$ patches from each image, for a total of $320,000$ training patches. During the training, the fraction of training patches moving between J and $\overline{J}$ was recorded for each atom pair, and the bars (on the left) show the median of these values for each training iteration. Note that the typical fraction of signals changing sides is $< 12\%$ for the first iteration, and around $2 - 6\%$ for the remaining iterations. The corresponding error evolution for this training process in depicted on the right.

has no effect. The threshold value will indeed require individual optimization following the update of $\mathbf{w}_j$ and $\mathbf{d}_j$, as it should be tuned to the values of the updated atoms.

Problem (7.11) is a simple rank-one approximation task whose solution can be obtained via the SVD. Due to the presence of the matrix $\mathbf{Y}^{J_0}$ to the right of the atom pair, the solution process entails a few technical details which we leave for the appendix (see Appendix 7.B). The resulting rank-one approximation procedure is listed in Algorithm 7.1.

Updating the Threshold

Once $\mathbf{w}_j, \mathbf{d}_j$ have been updated according to (7.11), we must recompute the threshold λ_j to match the new atom values. Based on our previous assumption, we expect most of the analysis coefficients $|\mathbf{w}_j^T \mathbf{y}_i|$ to be well separated in respect to J_0 and

Algorithm 7.1 Thresholding — Rank-One Approximation

```
Input: Matrices $\mathbf{E}, \mathbf{Y} \in \mathbb{R}^{N \times R}$
Output: Solution to $\underset{\mathbf{d},\mathbf{w}}{\text{Argmin}} \; \|\mathbf{E} - \mathbf{d}\mathbf{w}^T\mathbf{Y}\|_F^2$  Subject To  $\|\mathbf{w}\|_2 = 1$
procedure:
    Compute the SVD: $\mathbf{Y} = \mathbf{U}\mathbf{S}\mathbf{V}^T$
    $\boldsymbol{\Delta} := \text{diag}(s_1^{-1}, \ldots, s_N^{-1})$
    $\widetilde{\mathbf{Y}} := \boldsymbol{\Delta}\mathbf{U}^T\mathbf{Y}$
    $\{\mathbf{d}, \tilde{\mathbf{w}}\} := \underset{\mathbf{d},\tilde{\mathbf{w}}}{\text{Argmin}} \; \|\mathbf{E}\widetilde{\mathbf{Y}}^T - \mathbf{d}\tilde{\mathbf{w}}^T\|_F^2$
    $\mathbf{w}^T := \tilde{\mathbf{w}}^T\boldsymbol{\Delta}\mathbf{U}^T$
    $\mathbf{d} := \mathbf{d} \cdot \|\mathbf{w}\|_2$
    $\mathbf{w}^T := \mathbf{w}^T / \|\mathbf{w}\|_2$
end
```

$\overline{J}_0$ around some threshold point. However, rather than seek this separation point, a more straightforward and effective way to select λ_j is to simply minimize the original error target:

$$\hat{\lambda}_j = \underset{\lambda_j}{\text{Argmin}} \; \|\mathbf{E}_j - \mathbf{d}_j S_{\lambda_j}(\mathbf{w}_j^T\mathbf{Y})\|_F^2 \; . \tag{7.12}$$

In practice, this process typically produces a partitioning close to the original one, due to the relatively good separation of the signals in J_0 and $\overline{J}_0$. This is illustrated in Fig. 7.1, which employs this technique for choosing λ_j.

Owing to the discrete nature of the hard threshold operator, problem (7.12) can be globally and efficiently solved via a simple process. Without loss of generality, we assume the signals are ordered such that $|\mathbf{w}_j^T\mathbf{y}_1| \le |\mathbf{w}_j^T\mathbf{y}_2| \le \cdots \le |\mathbf{w}_j^T\mathbf{y}_R|$. Thus, for any value of $\lambda_j \in (|\mathbf{w}_j^T\mathbf{y}_1|, |\mathbf{w}_j^T\mathbf{y}_R|)$, there exists a unique index $k = k(\lambda_j)$ such that $|\mathbf{w}_j^T\mathbf{y}_{k-1}| < \lambda_j \le |\mathbf{w}_j^T\mathbf{y}_k|$. The examples which survive the threshold are therefore given by $y_k, y_{k+1}, \ldots, y_R$, and we can rewrite (7.12) as:

$$\hat{\lambda}_j = \underset{\lambda_j}{\text{Argmin}} \sum_{i=1}^{k(\lambda_j)-1} \|\mathbf{e}_i\|_2^2 + \sum_{i=k(\lambda_j)}^{R} \|\mathbf{e}_i - \mathbf{d}_j \mathbf{w}_j^T \mathbf{y}_i\|_2^2 \ .$$

In this formulation, k encloses all the necessary information about λ_j. The optimization can therefore be carried out over the discrete breakpoint k, which is a simple task. Introducing the notations $\alpha_i = \|\mathbf{e}_i\|_2^2$ and $\beta_i = \|\mathbf{e}_i - \mathbf{d}_j \mathbf{w}_j^T \mathbf{y}_i\|_2^2$, the optimization for k is given by:

$$\hat{k} = \underset{k}{\text{Argmin}} \sum_{i=1}^{k-1} \alpha_i + \sum_{i=k}^{R} \beta_i \ . \tag{7.13}$$

This expression is minimized directly by computing the values $s_k = \sum_{i=1}^{k-1} \alpha_i + \sum_{i=k}^{R} \beta_i$ for all k and taking the global minimum. The values s_k are computed via the recursion $s_1 = \sum_{i=1}^{R} \beta_i$ and $s_{k+1} = s_k + \alpha_k - \beta_k$. Once the value $\hat{k}$ is known, any suitable value for λ_j can be selected, e.g., $\lambda_j = (|\mathbf{w}_j^T \mathbf{y}_{\hat{k}-1}| + |\mathbf{w}_j^T \mathbf{y}_{\hat{k}}|)/2$. The threshold update process is summarized in Algorithm 7.2.

Full Training Process and Implementation Details

Putting the pieces together, the atom update process for the j-th atom pair consists of the following three steps: (a) detecting the set J_0 of signals using the current atom pair; (b) updating $\mathbf{w}_j$ and $\mathbf{d}_j$ using (7.11); and (c) recomputing the threshold by solving (7.13). The algorithm processes the dictionary atoms in sequence, and thus benefits from having the updated atoms and error matrix available for the subsequent updates. The full training process is detailed in Algorithm 7.3. Note that the algorithm assumes some initial choice for $\mathbf{\Omega}_0$, $\mathbf{D}_0$ and $\boldsymbol{\lambda}_0$. In practice, our implementation only requires an initial $\mathbf{\Omega}_0$; for $\mathbf{D}_0$ we initialize $\mathbf{D}_0 = \mathbf{X}(\mathbf{\Omega}_0\mathbf{Y})^+$, and for $\boldsymbol{\lambda}_0$ we begin with an arbitrary choice $\boldsymbol{\lambda}_0 = (\hat{\lambda}, \ldots \hat{\lambda})$ where $\hat{\lambda}$ is the median of the coefficients in $|\mathbf{\Omega}\mathbf{Y}|$. We then run one sweep of Algorithm 7.2 over all threshold values to adapt them to the initial dictionaries.

As previously mentioned, the proposed atom update process is subject to the condition that J_0 have some minimal size. In practice, we set this minimum size

Algorithm 7.2 Thresholding — Threshold Update

```
Input: Matrices $\mathbf{E}, \mathbf{Y} \in \mathbb{R}^{N \times R}$, atoms $\mathbf{d}, \mathbf{w} \in \mathbb{R}^N$
Output: Solution to $\underset{\lambda}{\text{Argmin}}\ \|\mathbf{E} - \mathbf{d} S_\lambda(\mathbf{w}^T \mathbf{Y})\|_F^2$
Preprocess: Sort the columns of $\mathbf{E}$ and $\mathbf{Y}$ in increasing order of $|\mathbf{w}^T \mathbf{Y}|$
procedure:
    $\forall i: \quad \alpha_i := \|\mathbf{e}_i\|_2^2$
    $\forall i: \quad \beta_i := \|\mathbf{e}_i - \mathbf{d}\mathbf{w}^T \mathbf{y}_i\|_2^2$
    $s_1 := \sum_{i=1}^R \beta_i$
    for $k = 1 \dots R$ do
        $s_{k+1} := s_k + \alpha_k - \beta_k$
    end for
    $\hat{k} := \underset{k}{\text{Argmin}}\ s_k$
    $\lambda := \begin{cases} 0 & \hat{k} = 1 \\ |\mathbf{w}^T \mathbf{y}_R| + 1 & \hat{k} = R + 1 \\ (|\mathbf{w}^T \mathbf{y}_{\hat{k}-1}| + |\mathbf{w}^T \mathbf{y}_{\hat{k}}|)/2 & \text{otherwise} \end{cases}$
end
```

to a liberal 5% of the examples, which is satisfied in most cases. When this is not satisfied, however, we use a default procedure which discards the current atom pair, and applies steps (b) and (c) above with J_0 being the entire set of signals. This heuristic process replaces the atom pair with a new pair, which is typically used by more examples. A complementing approach, which we do not currently employ but is also potentially useful, is to allow a few atoms with a smaller number of associated examples to prevail, and optimize these using the constrained QP process described in the Appendix.

Algorithm 7.3 Thresholding Dictionary Training

```
Input: Training signals $\mathbf{X} \in \mathbb{R}^{M \times R}$, degraded signals $\mathbf{Y} \in \mathbb{R}^{N \times R}$, initial dictionaries $\mathbf{\Omega}_0 \in \mathbb{R}^{L \times N}, \mathbf{D}_0 \in \mathbb{R}^{M \times L}$, initial thresholds $\boldsymbol{\lambda}_0$, number of iterations $k$
Output: Dictionary pair $\mathbf{\Omega}, \mathbf{D}$ and threshold vector $\boldsymbol{\lambda}$ minimizing (7.6)
Init: Set $\mathbf{\Omega} := \mathbf{\Omega}_0, \mathbf{D} := \mathbf{D}_0$, $\boldsymbol{\lambda} = \boldsymbol{\lambda}_0$
for $n = 1 \ldots k$ do
  for $j = 1 \ldots L$ do
    $J := \{\ i \in \{1 \ldots R\} \mid |\mathbf{w}_j^T \mathbf{y}_i| \geq \lambda\}$
    $E_j = \mathbf{X} - \sum_{k \neq j} \mathbf{d}_k S_{\lambda_k}(\mathbf{w}_k^T \mathbf{Y})$
    $\{\mathbf{d}_j, \mathbf{w}_j\} := \underset{\mathbf{d},\mathbf{w}}{\mathrm{Argmin}} \|\mathbf{E}_j^J - \mathbf{d}\mathbf{w}^T \mathbf{Y}^J\|_F^2 \quad \text{s.t.} \quad \|\mathbf{w}\|_2 = 1$   (Algorithm 7.1)
    $\lambda_j := \underset{\lambda}{\mathrm{Argmin}} \|\mathbf{E}_j - \mathbf{d}_j S_\lambda(\mathbf{w}_j^T \mathbf{Y})\|_F^2$   (Algorithm 7.2)
    $\mathbf{\Omega}\{j\text{-}th\ row\} := \mathbf{w}_j^T$
    $\mathbf{D}\{j\text{-}th\ col\} := \mathbf{d}_j$
    $\boldsymbol{\lambda}\{j\text{-}th\ elem\} := \lambda_j$
  end for
end for
```

7.4 Empirical Evaluation and Discussion

7.4.1 Experiment Setup

To evaluate the performance of the proposed formulation, we employed the described training process for image deblurring. Our training set consists of eight natural images taken from the CVG-Granada [185] data set. Four of these images are shown in Fig. 7.2. Each of the training images was blurred and subjected to additive white Gaussian noise, to produce eight pairs of origin and degraded input images. We then extracted from each pair 40,000 random training blocks along with their degraded versions, for a total of 320,000 example pairs. We subtract the mean from each example to obtain the final training set.

The initial dictionary $\mathbf{\Omega}_0$ for the training is the overcomplete DCT dictionary, and training is performed for 20 iterations. An example result of the training process is shown in Fig. 7.3. The top row shows the trained $\mathbf{\Omega}$ (left) and $\mathbf{D}$ (right). The bottom-left figure shows the absolute values of the coefficients in $\mathbf{D\Omega}$, and as can be seen, the matrix $\mathbf{D\Omega}$ exhibits a diagonal structure as expected from an operator for recovery of a localized convolution process. Finally, the bottom-right figure depicts the error evolution during the algorithm iterations.

For the deblurring process, we begin by extracting all overlapping blocks from the degraded image, and subtracting their mean. We then apply the learned thresholding process to the mean-subtracted blocks. Finally, the block means are restored, and we compute the deblurred result by averaging the overlapping recovered blocks. We evaluate our method on seven standard test images, all of which are not included in the training set: *Barbara, Cameraman, Chemical Plant, House, Lena, Peppers* and *Man*.

7.4.2 Results

Results of our deblurring process for two different blurring kernels are shown in Figs. 7.4 and 7.5. The two cases are taken from the work [186], whose inputs are made available online [187]. The figures compare our results with those of ForWaRD [188], LPA-ICI [189] and AKTV [186], the latter considered the current state-of-the-art in deblurring. The first case (Fig. 7.4) represents strong noise and small blur, and the second case (Fig. 7.5) represents moderate noise and moderate blur. In the current work we limit ourselves to handling small to moderate blur kernels, as large kernels would require much larger block sizes which are impractical in the current formulation. We thus do not replicate the two other cases considered in [186], which employ very large blur kernels. We note that large blur kernels could possibly be handled by our framework via downsampling of the input images, though we do not pursue this option here.

Figure 7.2: Four training images from the CVG Granada data set.

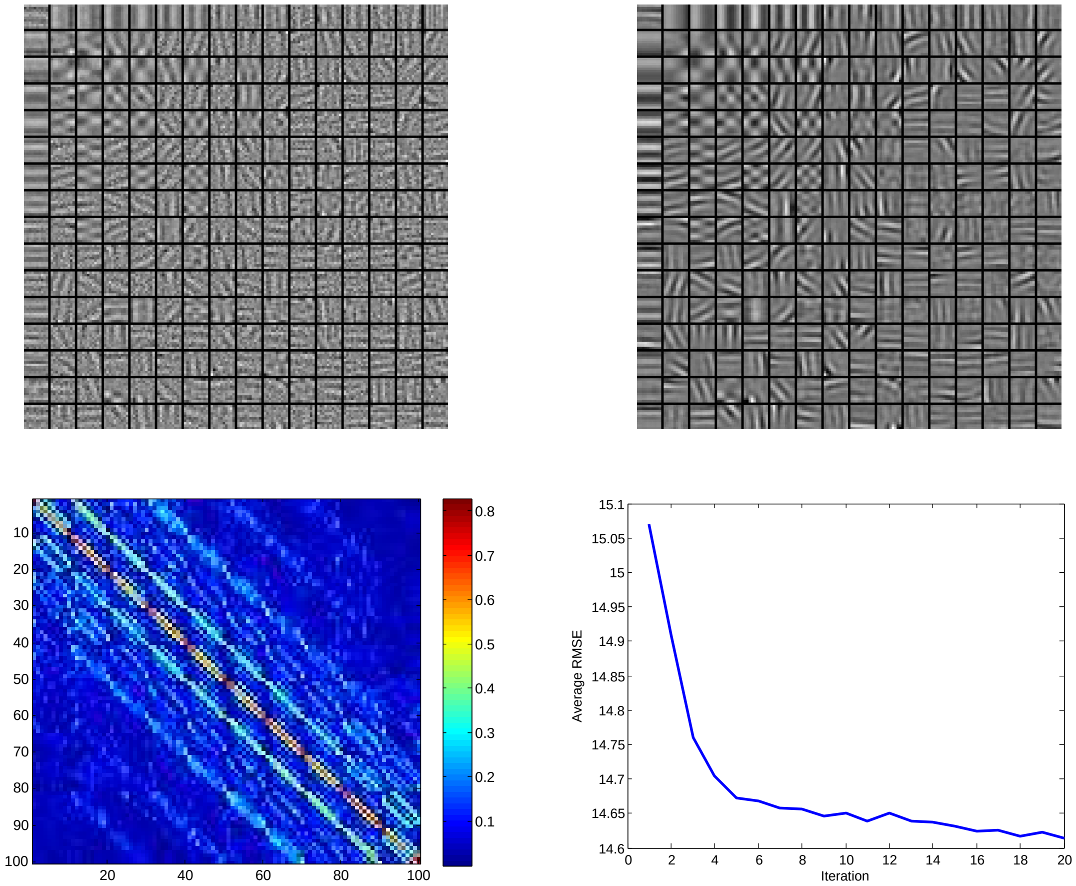


Figure 7.3: Results of the hard thresholding training algorithm for image deblurring. Top left: trained $\mathbf{\Omega}$. Top right: trained $\mathbf{D}$. Bottom left: absolute value of the coefficients in $\mathbf{D\Omega}$. Bottom right: Error evolution during the algorithm iterations (y-axis is the average RMSE of the recovered patches). Training was performed for 20 iterations, using 320,000 training signals. Omega is of size 256×100. Blurring kernel is a 5×5 Gaussian with standard deviation 1.5, Gaussian noise has standard deviation 8.25.

As can be seen, our results in both cases surpass ForWaRD and LPA-ICI in raw RMSE by a small margin, loosing only to the AKTV. Visually, our result in Fig. 7.4 maintains more of the noise than the other methods, though subjectively it also appears less ”processed”, and we note that lines and curves, for instance, appear straighter and less ”jaggy”. Continuing with Fig. 7.5, our result in this case seems more visually pleasing than that of ForWaRD and LPA-ICI, and reproduces more fine details (see for instance the field area at the top right). Compared to the AKTV, our result maintains slightly more noise, though it also avoids introducing the artificial smear and ”brush stroke” effects characteristic of the AKTV, and likely associated with its steering regularization kernel.

7.4.3 Discussion

Compared to the other methods, our deblurring process is very simple and efficient, and involves no parameter tuning. In these respects, the ForWaRD algorithm is the most comparable to our system as it is fast and its parameters can be automatically tuned, as described in [188]. The ForWaRD algorithm is also the most similar to our work as it is based on a scaling (shrinkage) process of the image coefficients in the Fourier and wavelet domains. The LPA-ICI and AKTV, on the other hand, both involve parameters which must be manually tuned to optimize performance. Also, while the LPA-ICI is relative fast, the AKTV in particular is extremely computationally intensive, requiring e.g., in the case shown in Fig. 7.4, at least 12 minutes to achieve a reasonable result, and nearly an hour to reproduce the final result shown in the figure. In comparison, our method on the same hardware completed in just 8 seconds, due to the diversion of most of the computational burden to the offline training phase. Furthermore, our recovery method is highly parallelizable, and can likely be optimized to achieve real-time performance.

Another notable difference between our method and the others is its ”model-less" nature, as previously mentioned. Indeed, all three methods (the ForWaRD, LPA-ICI and AKTV) assume accurate knowledge of the blurring kernel, which is typical of deconvolution frameworks. Our method is fundamentally different, as it replaces this assumption with a very different one — the availability of a set of training images undergoing the same degradation, which implicitly represent the convolution kernel. In practice, the difference between these two modeling paradigms may not be as large as it seems, as in both cases, a real-world application would require either a prior calibration process or an online degradation estimation method. However, in some cases, acquiring a training set for our method may be a simpler and more robust process (e.g., using a pair of low quality and high quality equipment) than a precise measurement of the point spread function.

Finally, our method is inherently indifferent to boundary issues, which plague some deconvolution methods. Our deconvolution process can be applied with no modification to images undergoing non-circular convolution, and will produce no visible artifacts near the image borders. Of the three methods we compare to, only the AKTV provides a similar level of immunity to boundary conditions.

Full deblurring results for the seven standard test images are summarized in Table 7.1. We compare our results to those of the ForWaRD algorithm, which we choose due to its combination of efficiency, lack of manual parameter tuning, and relation to our method. The thresholding results in these tables were produces using the same trained dictionaries used to produce the results in Figs. 7.4 and 7.5. The ForWaRD results were generated using the Matlab package available at [190].

7.5 Summary and Conclusions

This work has presented a novel technique for training the analysis and synthesis dictionaries of a thresholding-based image recovery process. Our method assumes a hard-thresholding operator, which leads to ℓ^0-sparse representations. We exploit this exact sparsity to design a simple training algorithm based on a sequence of

Image	Degraded	Thresh.	ForWaRD
Barbara	17.64	**15.44**	15.84
Camera.	17.36	**13.28**	13.46
Chem.Plant	14.45	**10.51**	11.48
House	8.76	**3.93**	5.03
Lena	10.78	**6.69**	7.44
Peppers	10.81	**6.90**	7.33
Man	12.34	**8.89**	9.54

Image	Degraded	Thresh.	ForWaRD
Barbara	16.57	**14.72**	14.81
Camera.	17.78	11.99	**11.82**
Chem.Plant	15.09	8.78	**8.58**
House	4.94	**2.43**	2.50
Lena	8.92	**5.45**	5.54
Peppers	8.60	**5.94**	6.06
Man	11.12	**7.67**	7.73

Table 7.1: Deblurring results for seven standard test images, using the degradation and dictionary parameters from Figs. 7.4 (left) and 7.5 (right). All values in the tables represent RMSE.

rank-one approximations, in the spirit of the K-SVD algorithm.

The training process simultaneously learns the dictionaries *and* the threshold values, making the resulting recovery process simple, efficient, and parameterless. Thresholding-based recovery is also naturally parallelizable, enabling for substantial acceleration. The proposed thresholding technique was applied to small-kernel image deblurring, where it was found to match or surpass leading dedicated deconvolution methods, and loose only to the highly computationally demanding AKTV. Also, our recovery process is stable under boundary condition changes, which some deconvolution methods are sensitive to.

A unique characteristic of our framework is its example-based approach to the degradation modeling process. Whereas most deconvolution and regularized inversion processes assume explicit knowledge of the signal degradation, our method assumes no prior knowledge of this process, and implicitly learns it from pairs of examples. Our approach can thus be applied in cases where an exact model of the degradation is unavailable, but a limited training set can be produced in a controlled environment.

7.6 Future Directions

Our work gives rise to several possible improvements and future research directions. First, the block-based nature of the recovery process imposes a limit on the size of the convolution kernels which can be handled, a limitation which could be approached by incorporating downscaling and upscaling operations within the recovery scheme. Alternatively, larger dictionaries could be trained using structured dictionary models such as the sparse dictionary [33].

To handle images of arbitrary size, our method employs block-processing followed by an averaging step. A possible technique to improve recovery quality is therefore to incorporate knowledge of the block averaging step in the training process, as suggested in [178]. Such a modification adds significant complexity to the training phase, but no additional complexity to the recovery process. Indeed, in [178] this approach is found to provide an additional gain in quality compared to the simpler approach.

Other straightforward extensions include training spatially-dependent dictionaries to handle non-translation-invariant degradations of a fixed pattern, and training single dictionary pairs for multiple noise levels. Multi-scale thresholding is also an attractive option which could improve performance as well as assist in handling wider-supported degradations. Multi-scale processing could be implemented e.g., using variable-sized blocks, by thresholding in a multi-scale transform domain, or by training dictionaries with a multi-scale structure.

Finally, extending the process to more general thresholding operators remains an open question, with the potential of dramatically improving results. Specifically, developing a unified framework which would perform both dictionary training and threshold operator adaptation is an exciting possibility with far-reaching potential.

7.A Quadratic Programming Atom Update

In this appendix we describe the formulation of the atom update process (7.8) as a constrained Quadratic Programming (QP) problem. We begin with the update task

$$\{\hat{\mathbf{d}}_j, \hat{\mathbf{w}}_j\, \hat{\lambda}_j\} = \underset{\mathbf{d}_j,\mathbf{w}_j,\lambda_j}{\text{Argmin}} \; \|\mathbf{E}_j - \mathbf{d}_j S_{\lambda_j}(\mathbf{w}_j^T \mathbf{Y})\|_F^2 \quad \text{Subject To} \quad \|\mathbf{w}_j\|_2 = 1 \ ,$$

and take a block-coordinate-relaxation approach in which $\mathbf{d}_j$ is updated independently of $\mathbf{w}_j$ and λ_j. In this scheme, updating $\mathbf{d}_j$ is a simple least-squares task given by

$$\mathbf{d}_j = \mathbf{E}_j \boldsymbol{\gamma}_j / (\boldsymbol{\gamma}_j^T \boldsymbol{\gamma}_j) \ , \tag{7.14}$$

with $\boldsymbol{\gamma}_j = S_{\lambda_j}(\mathbf{Y}^T \mathbf{w}_j)$.

Moving to the update of $\mathbf{w}_j$ and λ_j, in the QP approach we constrain the update such that it maintains the partitioning of the training signals about the threshold. Thus, we split $\mathbf{Y}$ to the signals $\mathbf{Y}^J$ that survive the current threshold and the remaining signals $\mathbf{Y}^{\overline{J}}$, and similarly split $\mathbf{E}_j$ to $\mathbf{E}_j^J$ and $\mathbf{E}_j^{\overline{J}}$, obtaining:

$$\begin{aligned} \{\hat{\mathbf{w}}_j, \hat{\lambda}_j\} \ = \ & \underset{\mathbf{w}_j,\lambda_j}{\text{Arg}\min} \; \|\mathbf{E}_j^J - \mathbf{d}_j \mathbf{w}_j^T \mathbf{Y}^J\|_F^2 + \|\mathbf{E}_j^{\overline{J}}\|_F^2 \\ & \text{Subject To} \quad |\mathbf{w}_j^T \mathbf{y}_i| \geq \lambda_j \;\; \forall i \in J \\ & \qquad\qquad\quad\;\; |\mathbf{w}_j^T \mathbf{y}_i| < \lambda_j \;\; \forall i \in \overline{J} \\ & \qquad\qquad\quad\;\; \|\mathbf{w}_j\|_2 = 1 \end{aligned} \quad . \tag{7.15}$$

The constraints ensure that the signal partitioning is maintained by the update process. Note that due to the constraining, J is constant in the optimization.

To bring the problem to QP form, we recall that the norm constraint on $\mathbf{w}_j$ is an arbitrary normalization choice which we can replace, e.g., with a fixed value for λ_j. Thus, we choose to lift the norm constraint on $\mathbf{w}_j$ and instead fix the threshold λ_j at its current value. Indeed, the outcome of this optimization can be subsequently re-scaled to satisfy the original unit-norm constraint. Adding the fact that $\mathbf{E}_j^{\overline{J}}$ is fixed in the above optimization (as J is fixed), the update task can be written as:

$$\begin{aligned} \hat{\mathbf{w}}_j &= \underset{\mathbf{w}_j}{\text{Argmin}} \ \|\mathbf{E}_j^J - \mathbf{d}_j \mathbf{w}_j^T \mathbf{Y}^J\|_F^2 \\ & \qquad \text{Subject To} \quad |\mathbf{w}_j^T \mathbf{y}_i| \geq \lambda_j \quad \forall i \in J \\ & \qquad\qquad\qquad\quad |\mathbf{w}_j^T \mathbf{y}_i| < \lambda_j \quad \forall i \in \overline{J} \end{aligned} \quad .$$

This formulation does not yet constitute a QP problem, as the first set of constraints is clearly non-convex. To remedy this, we must add the requirement that the coefficients $\mathbf{w}_j^T \mathbf{y}_i$ *do not change sign* during the update process, for the signals in the set J. In other words, we require that $\mathbf{w}_j$ does not "change sides" relative to the signals in $\mathbf{Y}^J$. While this choice adds further constraining to the problem, in practice many local optimization techniques would be oblivious to the discontinuous optimization regions anyway, and we thus accept the added constraints in return for a manageable optimization task. Of course, an important point about this specific choice of constraints is that it necessarily leads to a non-empty feasible region, with the current $\mathbf{w}_j$ constituting a good starting point for the optimization.

With the updated set of constraints, the optimization domain becomes convex, and the problem can be formulated as a true QP problem. To express the new constraints, we denote by $\sigma_i = \text{sign}(\mathbf{w}_j^T \mathbf{y}_i)$ the signs of the inner products of the signals with the *current* atom. We can now write the update process for $\mathbf{w}_j$ as:

$$\begin{aligned} \hat{\mathbf{w}}_j &= \underset{\mathbf{w}_j}{\text{Argmin}} \ \|\mathbf{E}_j^J - \mathbf{d}_j \mathbf{w}_j^T \mathbf{Y}^J\|_F^2 \\ & \qquad \text{Subject To} \quad \sigma_i \mathbf{w}_j^T \mathbf{y}_i \geq \lambda_j \qquad\quad \forall i \in \mathbf{Y}_j \\ & \qquad\qquad\qquad\quad -\lambda_j < \mathbf{w}_j^T \mathbf{y}_i < \lambda_j \quad \forall i \in \bar{\mathbf{Y}}_j \end{aligned} \quad . \tag{7.16}$$

This problem is a standard QP optimization task, and can be solved using a variety of techniques. Once $\mathbf{w}_j$ is computed according to (7.16), we restore the original constraint on $\mathbf{w}_j$ by normalizing $\{\mathbf{w}_j, \lambda_j\} \to \{\alpha_j \mathbf{w}_j, \alpha_j \lambda_j\}$ with $\alpha_j = 1/\|\mathbf{w}_j\|_2$, and compute $\mathbf{d}_j$ using (7.14), which concludes the process.

7.B Rank-One Approximation Solution

In this appendix we consider the solution to the problem

$$\operatorname*{Argmin}_{\mathbf{d},\mathbf{w}} \; \|\mathbf{E}-\mathbf{d}\mathbf{w}^T\mathbf{Y}\|_F^2 \quad \text{Subject To} \quad \|\mathbf{w}\|_2 = 1 \;, \tag{7.17}$$

where $\mathbf{E},\mathbf{Y} \in \mathbb{R}^{N\times R}$, and are assumed to be full-rank. To derive the solution, we first assume that $\mathbf{Y}\mathbf{Y}^T = \mathbf{I}$ (i.e., $\mathbf{Y}^T$ is a tight frame). In this case we have[1]:

$$\begin{aligned}
\|\mathbf{E}-\mathbf{d}\mathbf{w}^T\mathbf{Y}\|_F^2 &= \operatorname{tr}\left\{\mathbf{E}^T\mathbf{E} - 2\mathbf{E}^T\mathbf{d}\mathbf{w}^T\mathbf{Y} + \mathbf{Y}^T\mathbf{w}\mathbf{d}^T\mathbf{d}\mathbf{w}^T\mathbf{Y}\right\} \\
&= \operatorname{tr}\left\{\mathbf{E}^T\mathbf{E} - 2\mathbf{Y}\mathbf{E}^T\mathbf{d}\mathbf{w}^T + \mathbf{Y}\mathbf{Y}^T\mathbf{w}\mathbf{d}^T\mathbf{d}\mathbf{w}^T\right\} \\
&= \operatorname{tr}\left\{\mathbf{E}^T\mathbf{E} - 2\mathbf{Y}\mathbf{E}^T\mathbf{d}\mathbf{w}^T + \mathbf{w}\mathbf{d}^T\mathbf{d}\mathbf{w}^T + \mathbf{Y}\mathbf{E}^T\mathbf{E}\mathbf{Y}^T - \mathbf{Y}\mathbf{E}^T\mathbf{E}\mathbf{Y}^T\right\} \\
&= \operatorname{tr}\left\{\mathbf{E}^T\mathbf{E} - \mathbf{Y}\mathbf{E}^T\mathbf{E}\mathbf{Y}^T\right\} + \operatorname{tr}\left\{\mathbf{Y}\mathbf{E}^T\mathbf{E}\mathbf{Y}^T - 2\mathbf{Y}\mathbf{E}^T\mathbf{d}\mathbf{w}^T + \mathbf{w}\mathbf{d}^T\mathbf{d}\mathbf{w}^T\right\} \\
&= \operatorname{tr}\left\{\mathbf{E}^T\mathbf{E} - \mathbf{Y}\mathbf{E}^T\mathbf{E}\mathbf{Y}^T\right\} + \|\mathbf{E}\mathbf{Y}^T - \mathbf{d}\mathbf{w}^T\|_F^2 \;.
\end{aligned}$$

Since the left term is constant in the optimization, we find that when $\mathbf{Y}^T$ is a tight frame, (7.17) is equivalent to:

$$\operatorname*{Argmin}_{\mathbf{d},\mathbf{w}} \; \|\mathbf{E}\mathbf{Y}^T - \mathbf{d}\mathbf{w}^T\|_F^2 \quad \text{Subject To} \quad \|\mathbf{w}\|_2 = 1 \;. \tag{7.18}$$

This is a standard rank-one approximation of $\mathbf{E}\mathbf{Y}^T$, and its solution is given by the singular vector pair corresponding to the largest singular value of $\mathbf{E}\mathbf{Y}^T$.

For a general full-rank $\mathbf{Y}$, we use the SVD of $\mathbf{Y}$, and write $\mathbf{Y} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. We denote the singular values on the diagonal of $\mathbf{S}$ by $s_1 \ldots s_N$, and let $\boldsymbol{\Delta} = \operatorname{diag}(s_1^{-1},\ldots,s_N^{-1})$. We note that the matrix

$$\widetilde{\mathbf{Y}} = \boldsymbol{\Delta}\mathbf{U}^T\mathbf{Y} = \boldsymbol{\Delta}\mathbf{S}\mathbf{V}^T = \mathbf{I}_{N\times R}\mathbf{V}^T$$

satisfies $\widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T = \mathbf{I}$, and thus $\widetilde{\mathbf{Y}}^T$ is a tight frame.

Returning to problem (7.17), we can now write

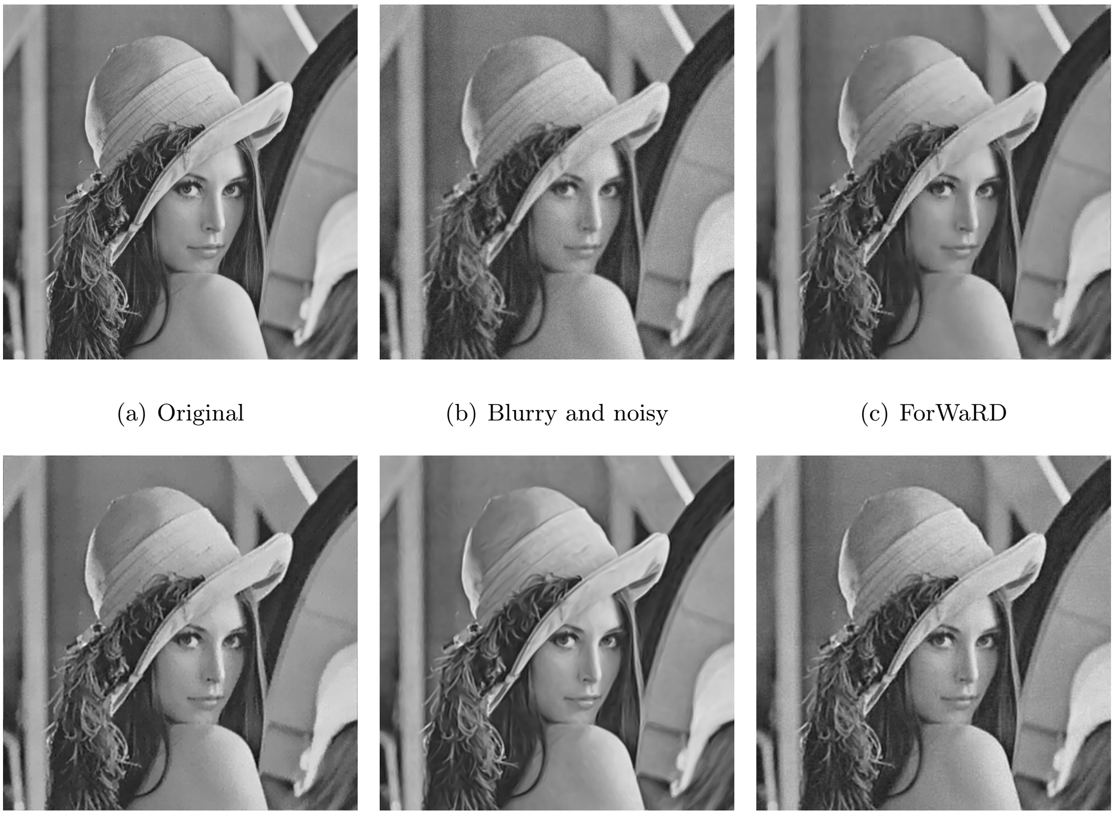
$$\|\mathbf{E}-\mathbf{d}\mathbf{w}^T\mathbf{Y}\|_F^2 \;=\; \|\mathbf{E}-\mathbf{d}\mathbf{w}^T(\boldsymbol{\Delta}\mathbf{U}^T)^{-1}\boldsymbol{\Delta}\mathbf{U}^T\mathbf{Y}\|_F^2 \;=\; \|\mathbf{E}-\mathbf{d}\mathbf{w}^T\mathbf{U}\boldsymbol{\Delta}^{-1}\widetilde{\mathbf{Y}}\|_F^2 \;,$$

[1] For simplicity of presentation, we slightly abuse notation by allowing differently-sized matrices to be summed within the trace operator. These should be interpreted summing the matrix traces.

which leads to the optimization task:

$$\operatorname*{Argmin}_{\mathbf{d},\mathbf{w}} \; \|\mathbf{E} - \mathbf{d}\mathbf{w}^T\mathbf{U}\boldsymbol{\Delta}^{-1}\widetilde{\mathbf{Y}}\|_F^2 \; . \tag{7.19}$$

Since $\widetilde{\mathbf{Y}}^T$ is a tight frame, (7.19) can be solved for $\mathbf{d}$ and $\tilde{\mathbf{w}}^T := \mathbf{w}^T\mathbf{U}\boldsymbol{\Delta}^{-1}$ using (7.18). Once $\tilde{\mathbf{w}}^T$ is computed, the computation is completed by setting $\mathbf{w}^T = \tilde{\mathbf{w}}^T\boldsymbol{\Delta}\mathbf{U}^T$, and renormalizing the obtained $\mathbf{d}$ and $\mathbf{w}$ such that $\|\mathbf{w}\|_2 = 1$. The resulting procedure is summarized in Algorithm 7.1.



(a) Original

(b) Blurry and noisy

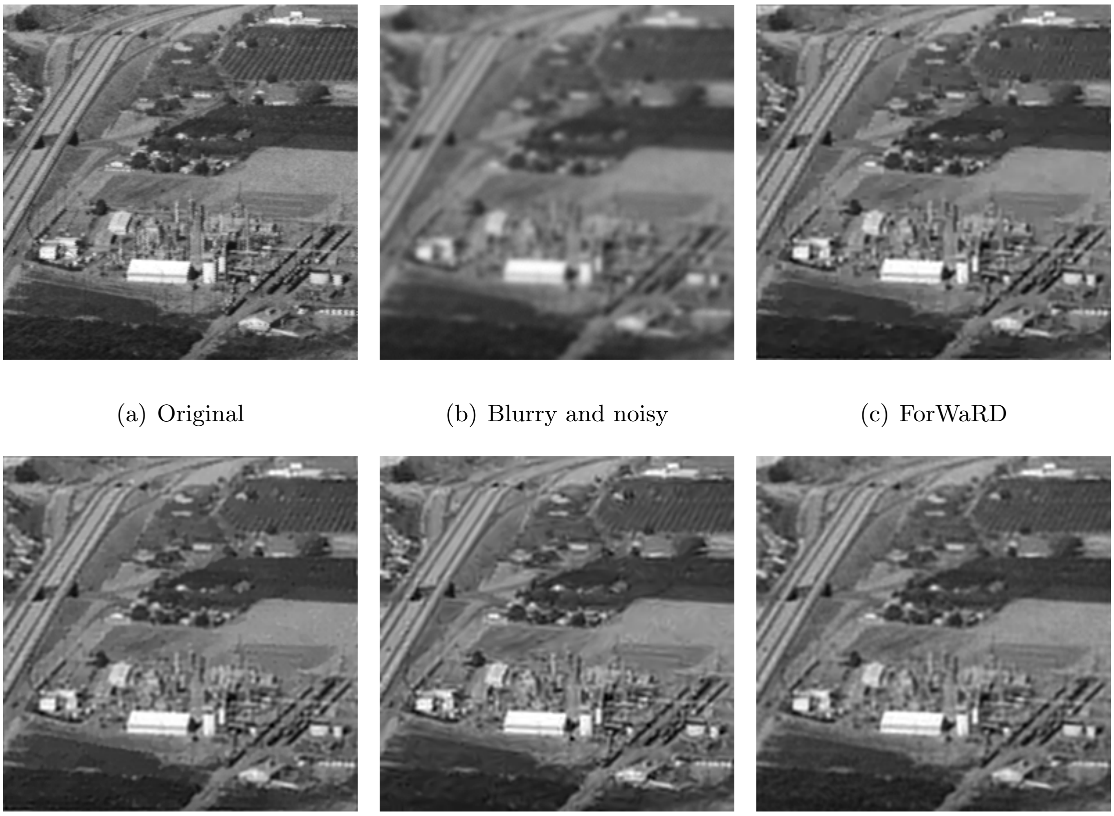
(c) ForWaRD

(d) LPA-ICI

(e) AKTV

(f) Thresholding

Figure 7.4: Deblurring results for *Lena*. Blurring kernel is a 5×5 Gaussian with standard deviation 1.5, additive noise is white Gaussian with standard deviation 8.25 (BSNR=15dB). RMSE values are 10.78 (blurry), 7.55 (ForWaRD), 6.76 (LPA-ICI), 6.12 (AKTV) and 6.69 (Thresholding). Thresholding parameters: block size is 10×10, dictionary size is 256×100.



(a) Original

(b) Blurry and noisy

(c) ForWaRD

(d) LPA-ICI

(e) AKTV

(f) Thresholding

Figure 7.5: Deblurring results for *Chemical Plant*. Blurring kernel is an 11 × 11 Gaussian with standard deviation 1.75, additive noise is white Gaussian with standard deviation 1.15 (BSNR=30dB). RMSE values are 15.09 (blurry), 8.98 (ForWaRD), 8.98 (LPA-ICI), 8.57 (AKTV) and 8.78 (Thresholding). Thresholding parameters: block size is 12 × 12, dictionary size is 576 × 144.

Chapter 8

Discussion and Conclusions

Dictionary-based signal modeling is a powerful and widely successful approach for describing natural signal phenomena. The modeling approach is based on profound notions of simplicity and conciseness, and has deep connections with fundamental concepts such as dimensionality reduction and minimal description length. The idea of describing signals through a dictionary of elementary atoms, controlled by sparsity forces, has had a profound impact on the research community, with applications spanning a wide scope of fields and tasks.

Analysis and synthesis models. The two main incarnations of the dictionary-based models are the *analysis* and *synthesis* models, which seek sparsity in either the decomposition or reconstruction of a signal over the set of atoms, respectively. As we have seen in this work, these two complementary approaches are *not* equivalent once overcomplete dictionaries are involved. Through a geometrical description, we have characterized a large set of signals for which the two are bound to differ, which we named *MAP principal signals*. Specifically of interest were the analysis principal signals, which are orthogonal to many rows in the analysis dictionary, and are treated most effectively by this model. The plentitude of these signals, however — exponential in the dictionary size — indicates that no similarly-sized synthesis model could effectively treat all these signals at once. Thus, an unavoidable gap is revealed between the two modeling approaches. Simulation results, depicted in Figs. 2.2–2.4, demonstrate the extent of this gap.

Reviewing our results for the analysis-synthesis gap, we note that they have a clear *worst-case* nature, focusing on the signals for which the two models differ the most. Indeed, tighter relations could be discovered for less sparse, non-principal signals. However, the realization of this fundamental gap between the two formulations opens the door to new research opportunities focused on the analysis model. Combined with additional indications of the potential of this model, such as those provided in Fig. 2.5, the formalization of this gap revives interest in the analysis model, which has been overshadowed in the past decade by the widespread success of the synthesis approach. With new tools acquired from the vast literature on synthesis models, new works on the analysis model are beginning to emerge, among which are two chapters in this thesis. As the interest in analysis models continues to grow, we expect additional works to gradually appear and explore the full potential of this exciting new field.

Dictionary design and parametric dictionaries. Applying a dictionary-based model in practice requires the selection of a concrete dictionary which describes the signals of interest. This choice is clearly critical to the performance of the dictionary-based approach, as its name so distinctly suggests. In this thesis we have given much attention to the selection of the dictionary. We have outlined the main ingredients in designing effective dictionaries — namely localization, geometric invariance, and adaptivity — and discussed the two main dictionary design paths — the analytic and the learning paths.

Many analytic dictionaries have been proposed over the years. Among the most notable are the Fourier, wavelet, and curvelet dictionaries (Figs. 3.1, 3.3, 3.4). Such dictionaries are designed around a specific well-understood family of signals (e.g., smooth, piecewise-smooth, or multi-dimensional piecewise-smooth, respectively), and deliver optimality for this simplified signal class. Analytic dictionaries typically provide good localization and geometric invariance, and are efficient and well-structured. On the other hand, such dictionaries lack one key property — adaptivity — due to the generic mathematical assumptions made in the design process. Adaptivity is the essence of learned dictionaries, which aim to capture more subtle signal behaviors through the example-based training process. Notable contributions in this area include the PCA, MOD, and K-SVD, as well as Olshausen and Field's experiments (Figs. 3.1, 3.2, 3.7). Learned dictionaries typically lack explicit structure and are less efficient than analytic dictionaries, however, the finer adaptation to the signal data leads to superior results in many applications.

A key conclusion from this discussion was the identification of a rising need for new dictionary structures which could merge the advantages of the two design paradigms. This need is most adequately addressed by *parametric dictionaries*, which are dictionaries described by a relatively small, well-defined set of values. Such dictionaries have the potential of combining structure, efficiency, and geometric invariances with adaptivity provided by the parameter tuning. Several such dictionaries have been recently proposed, among which we note the union-of-orthobases, semi-multiscale, and image-signature dictionaries (Eqs. (4.2) and (4.3), and Fig. 3.7, respectively). We expect such approaches to draw increasing attention in the coming years, with new dictionary designs providing a variety of blends of structure and adaptivity.

Sparse dictionaries. Specific efforts in the direction of parametric dictionary design have led to the development of the *sparse dictionary* model, proposed in this thesis as a particular flexible, adaptive and efficient dictionary structure for sparse signal representation (Eq. (4.4)). Underlying this model is the idea of a global set of *sub-atomic signals* whose combinations explain the formation of all observable dictionary atoms, using the same sparsity rules as those governing signal creation. The sparse dictionary combines efficiency and compact structure with a high degree of adaptivity, and, by supporting a variable number of parameters, provides a nearly smooth transition from analytic to fully-trained dictionaries. An added benefit of this structure, which was illustrated in Fig. 4.5, is its improved generalization ability in the presence of few and noisy examples. This property can become critical when handling large and high-dimensional signal data, where substantial training sets are infeasible.

The sparse dictionary structure was tested with 3-D computed tomography data, where it was found to provide equivalent or superior denoising results — at substantially shorter run-times — compared to a non-structured trained dictionary (Tables 4.1 and 4.2). Indeed, the sparse structure is particularly useful for such multi-dimensional data, where a fully unconstrained dictionary requires an impractical number of examples for effective training. Thus, much of the success of the sparse dictionary in this case can be associated with its improved generalization ability, due to the small and noisy nature of the training set.

A second application of the sparse dictionary was presented in Chapter 5, where the compact representation of the dictionary was exploited to design a novel adaptive image compression scheme. The uniqueness of the proposed system is in the replacement of the fixed dictionary, commonly used in transform-based compression schemes, with an online-learned, *input-adaptive* trained dictionary, sent as part of the compressed data. Such set-ups, to the best of the authors' knowledge, have so far been regarded as impractical due to the cost of transmitting the dictionary. Our system was shown to provide a consistent gain over JPEG compression, though below JPEG2000 performance (Fig. 5.7). While indeed below state-of-the-art, the described system remains significant in that it demonstrates the feasibility of the adaptive approach for generic image compression, positioning it as a viable alternative to traditional fixed-dictionary schemes.

Analysis dictionary training. While dictionary training for synthesis-based models has received thorough attention in the literature, the quest for a dictionary *specific to analysis models* is a recent and challenging undertaking. In this thesis, analysis dictionary training has been explored from two directions, corresponding to two novel forms of the analysis model — the ℓ^0 *analysis model* and the *analysis-synthesis thresholding* model. The two models are of particular interest as both employ exact sparsity measures parallel to the well-established ℓ^0 synthesis model, and thus allow harnessing similar methodology and approaches in the analysis setting. The results of these efforts demonstrate the potential and usefulness of employing such modern algorithmic machinery in analysis-based frameworks, motivating further research into this promising new direction.

The ℓ^0 *analysis model*, which describes signals as orthogonal to sets atoms in the analysis dictionary, has emerged as a natural outcome of the geometrical interpretation of the ℓ^1 analysis model. This interpretation has characterized the analysis principal signals as having many vanishing inner-products with the dictionary, meaning that they are ℓ^0-analysis sparse. This is parallel to the synthesis model, where the principal signals of the ℓ^1 formulation are ℓ^0-synthesis sparse. This view has led to the development of an efficient K-SVD-like training method for the ℓ^0 analysis approach, which involves a minimum-singular-value task in place of the maximum-singular-value one in the original K-SVD. The resulting *Analysis K-SVD* algorithm was shown to recover underlying dictionaries from training examples with high accuracy (Figs. 6.2 and 6.3). Additional experiments with natural images have demonstrated the recovery of localized and oriented dictionary atoms (Fig. 6.5), indicating the ability of the process to reveal fundamental behaviors in the training data. Indeed, additional research is required to develop applications for this new model, as well as more rigorous mathematical tools for handling it. Nonetheless, the results presented here show the potential in the ℓ^0 analysis path, and are expected to raise interest in this new approach.

The *analysis-synthesis thresholding model* (Eq. (7.4)) was proposed in this thesis as an extension to the widely-popular hard thresholding denoising method introduced nearly two decades ago. The process utilizes a pair of analysis and synthesis dictionaries, and can accommodate a variety of recovery tasks by lifting the pseudo-inverse constraint between the two dictionaries. We presented a training method for simultaneously learning both dictionaries for a specific inverse problem from pairs of origin and degraded examples. In this way, the method substitutes the need for a precise degradation model with a training process which concludes it from examples. The effectiveness of the process was demonstrated for small-kernel image deblurring, where it was found to be competitive with recent dedicated deconvolution methods (Table 7.1, Figs. 7.4 and 7.5). Compared to alternative methods, the thresholding approach provides a particularly simple, efficient, parameterless, and readily parallelizeable recovery process, and is inherently stable to boundary conditions. Also, though our implementation assumes a stationary process, the thresholding framework can equally support more complex degradations by utilizing different dictionaries for different regions in the image (though the spatial pattern of the degradation must be known). We thus find the thresholding-based process to be a simple, flexible and effective option for inverse problem solution. A variety of possible improvements, such as employing parametric dictionaries to handle larger image blocks, or simultaneous training of the dictionaries and the shrinking functions, provide additional opportunities to enhance this process and expand its applicability, making it an appealing technique for signal restoration and recovery.

Epilogue. This thesis has been but one step in a long journey of signal modeling methodology and applications, dating back over half a decade. The two main directions set by this work are the analysis modeling path, and the parametric dictionary design path. As additional research accumulates, we expect both directions to mature and become essential tools in signal modeling. Many future directions and objectives have been mentioned throughout the text, and promise new opportunities, challenges and successes in both fields.

References

[1] M Elad, P Milanfar, and R Rubinstein. Analysis versus synthesis in signal priors. *Inverse Problems*, 23(3):947–968, 2007.

[2] 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.

[3] M Elad and A Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Transactions on Image Processing*, 6 (12):1646–1658, 1997.

[4] R Molina, J Nunez, F J Cortijo, and J Mateos. Image restoration in astronomy: a Bayesian perspective. *IEEE Signal Processing Magazine*, 18(2):11–29, 2001.

[5] A Barron, J Rissanen, and B Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, 1998.

[6] M Li and P Vitanyi. *An introduction to Kolmogorov complexity and its applications.* Springer-Verlag, New York, second edition, 1997.

[7] S Mallat. *A wavelet tour of signal processing: the sparse way.* Academic Press, third edition, 2009.

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

[9] E P Simoncelli and B A Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24(1):1193–1216, 2001.

[10] E P Simoncelli. Statistical modeling of photographic images. In A Bovik, editor, *Handbook of Image and Video Processing.* Academic Press, 2005.

[11] S Mallat and Z Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[12] S S Chen, D L Donoho, and M A Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[13] R Rubinstein, A M Bruckstein, and M Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[14] E P Simoncelli, W T Freeman, E H Adelson, and D J Heeger. Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2 part 2):587–607, 1992.

[15] I W Selesnick, R G Baraniuk, and N C Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151, 2005.

[16] E J Candès and D L Donoho. Curvelets – a surprisingly effective nonadaptive representation for objects with edges. *Curves and Surfaces*, 1999.

[17] E J Candès, L Demanet, D L Donoho, and L Ying. Fast discrete curvelet transforms. *Multiscale Modeling & Simulation*, 5:861–899, 2006.

[18] M N Do and M Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12):2091–2106, 2005.

[19] Y Lu and M N Do. A new contourlet transform with sharp frequency localization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1629–1632, 2006.

[20] Y M Lu and M N Do. Multidimensional directional filter banks and surfacelets. *IEEE Transactions on Image Processing*, 16(4):918–931, 2007.

[21] D Labate, W Lim, G Kutyniok, and G Weiss. Sparse multidimensional representation using shearlets. In *Wavelets XI (Proceedings of SPIE)*, volume 5914, pages 254–262, 2005.

[22] B A Olshausen and D J Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997.

[23] K Engan, S O Aase, and J Hakon Husoy. Method of optimal directions for frame design. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 5:2443–2446, 1999.

[24] M S Lewicki and T J Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[25] 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.

[26] P Sallee and B A Olshausen. Learning sparse multiscale image representations. *Advances in Neural Information Processing Systems*, 15:1327–1334, 2003.

[27] S Lesage, R Gribonval, F Bimbot, and L Benaroya. Learning unions of orthonormal bases with thresholded singular value decomposition. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 5:293–296, 2005.

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

[29] K Engan, K Skretting, and J H Husøy. Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation. *Digital Signal Processing*, 17(1): 32–49, 2007.

[30] H Lee, A Battle, R Raina, and A Y Ng. Efficient sparse coding algorithms. *Advances in neural information processing systems*, 19:801–808, 2007.

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

[32] M Aharon and M Elad. Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, 2008.

[33] R Rubinstein, M Zibulevsky, and M Elad. Double sparsity: learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564, 2010.

[34] J Mairal, F Bach, J Ponce, and G Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[35] B Ophir, M Lustig, and M Elad. Multi-scale dictionary learning using Wavelets. *IEEE Selected Topics in Signal Processing*. To appear.

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

[37] G Davis, S Mallat, and M Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, 1997.

[38] Y C Pati, R Rezaiifar, and P S Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. *1993 Conference Record of The 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44, 1993.

[39] D L Donoho, Y Tsaig, I Drori, and J L Starck. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. *Technical Report – Statistics, Stanford*, 2006.

[40] T Blumensath and M Davies. Gradient pursuits. *IEEE Transactions on Signal Processing*, 56(6):2370–2382, 2008.

[41] D Needell and J A Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[42] D L Donoho and M Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via L_1 minimization. *Proceedings of the National Academy of Sciences*, 100: 2197–2202, 2003.

[43] I Daubechies, M Defrise, and C De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

[44] M Elad. Why simple shrinkage is still relevant for redundant representations? *IEEE Transactions on Information Theory*, 52(12):5559–5569, 2006.

[45] M Elad, B Matalon, and M Zibulevsky. Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization. *Applied and Computational Harmonic Analysis*, 23(3):346–367, 2007.

[46] 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.

[47] S Sardy, A G Bruce, and P Tseng. Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics*, 9(2):361–379, 2000.

[48] K Schnass and P Vandergheynst. Dictionary preconditioning for greedy algorithms. *IEEE Transactions on Signal Processing*, 2007.

[49] D L Donoho and M Elad. On the stability of the basis pursuit in the presence of noise. *Signal Processing*, 86(3):511–532, 2006.

[50] J A Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

[51] M. Elad and I. Yavneh. A plurality of sparse representations is better than the sparsest one alone. *IEEE Transactions on Information Theory*, 55(10):4701–4714, 2009.

[52] Y Li. A globally convergent method for ℓ^p problems. *SIAM Journal on Optimization*, 3: 609–629, 1993.

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

[54] G Peyré. *Literature review on sparse optimization*, 2008. Available online at http://www.ceremade.dauphine.fr/~peyre/cs-tv/OptimReview.pdf.

[55] L I Rudin, S Osher, and E Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[56] P Blomgren and T F Chan. Color TV: total variation methods for restoration of vector-valued images. *IEEE Transactions on Image Processing*, 7(3):304–309, 1998.

[57] M Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing*, 11(10):1141–1151, 2002.

[58] S Roth and M J Black. Fields of experts. *International Journal of Computer Vision*, 82 (2):205–229, 2009.

[59] R R Schultz and R L Stevenson. A Bayesian approach to image expansion for improved definition. *IEEE Transactions on Image Processing*, 3(3):233–242, 1994.

[60] C Bouman and K Sauer. A generalized Gaussian image model for edge-preserving MAP estimation. *IEEE Transactions on Image Processing*, 2(3):296–310, 1993.

[61] S Farsiu, M D Robinson, M Elad, and P Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, 2004.

[62] S Farsiu, M Elad, and P Milanfar. Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing*, 15(1):141–159, 2006.

[63] X Li, Z Wei, L Xiao, Y Sun, and J Yang. Compressed sensing image reconstruction based on morphological component analysis. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 2129–2132, 2009.

[64] J Mairal, M Elad, and G Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.

[65] J Mairal, F Bach, J Ponce, G Sapiro, and A Zisserman. Non-local sparse models for image restoration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

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

[67] S Raboy. On the recovery of missing samples via sparsity conditions. Master's thesis, Electrical Engineering Department, The Technion – Israel Institute of Technology, February 2007.

[68] J Yang, J Wright, T Huang, and Y Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 99(11):2861–2873, 2010.

[69] R Zeyde, M Protter, and M Elad. On single image scale-up using sparse-representation. In *Curves and Surfaces*, 2010.

[70] M Zibulevsky and B A Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 13(4):863–882, 2001.

[71] R Gribonval and S Lesage. A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In *Proceedings of the 14th European Symposium on Artificial Neural Networks (ESANN)*, pages 323–330, 2006.

[72] M D Plumbley, T Blumensath, L Daudet, R Gribonval, and M E Davies. Sparse representations in audio and music: From coding to source separation. *Proceedings of the IEEE*, 98(6):995–1005, 2010.

[73] S A Abdallah and M D Plumbley. Unsupervised analysis of polyphonic music by sparse coding. *IEEE Transactions on Neural Networks*, 17(1):179–196, 2006.

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

[75] J Shtok, M Elad, and M Zibulevsky. Sparsity-based sinogram for low-dose computed tomogrpahy. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

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

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

[78] I W Selesnick and K Y Li. Video denoising using 2D and 3D dual-tree complex wavelet transforms. In *Wavelets: Applications in Signal and Image Processing X (Proceedings of SPIE)*, volume 5207, pages 607–618, 2003.

[79] A L da Cunha, J Zhou, and M N Do. The nonsubsampled contourlet transform: theory, design, and applications. *IEEE Transactions on Image Processing*, 15(10):3089–3101, 2006.

[80] G Easley, D Labate, and W Lim. Sparse directional image representations using the discrete shearlet transform. *Applied and Computational Harmonic Analysis*, 25:25–46, 2008.

[81] M Elad, B Matalon, J Shtok, and M Zibulevsky. A wide-angle view at iterated shrinkage algorithms. In *Wavelets XII (Proceedings of SPIE)*, volume 6701, 2007.

[82] R Vidal, Y Ma, and S Sastry. Generalized principal component analysis (GPCA). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959, 2005.

[83] I Horev, O Bryt, and R Rubinstein. Adaptive Image Compression Using Sparse Dictionaries. *Technical Report, CS – Technion*, 2011.

[84] M Elad, J L Starck, P Querre, and D L Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19:340–358, 2005.

[85] M Genussov. Transcription and classification of audio data by sparse representations and geometric methods. Master's thesis, Electrical Engineering Department, The Technion – Israel Institute of Technology, 2010.

[86] B Ophir, M Elad, N Bertin, and M D Plumbley. Sequential minimal eigenvalues – an approach to analysis dictionary learning. In *European Signal Processing Conference (EUSIPCO)*, 2011.

[87] M Yaghoobi, S Nam, R Gribonval, and M E Davies. Analysis operator learning for overcomplete cosparse representations. In *European Signal Processing Conference (EUSIPCO)*, 2011.

[88] S Roth and M J Black. Fields of experts: A framework for learning image priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 860–867, 2005.

[89] E P Simoncelli. Bayesian denoising of visual images in the wavelet domain. *Lecture Notes in Statistics*, pages 291–308, 1999.

[90] A Chambolle, R A DeVore, N Lee, and B J Lucier. Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transactions on Image Processing*, 7(3):319, 1998.

[91] D M Higdon, J E Bowsher, V E Johnson, T G Turkington, D R Gilland, and R J Jaszczak. Fully bayesian estimation of gibbs hyperparameters for emission computed tomography data. *IEEE Transactions on Medical Imaging*, 16(5):516–526, 1997.

[92] E J Candès and D L Donoho. Ridgelets: a key to higher-dimensional intermittency? *Philosophical Transactions A: Mathematical, Physical and Engineering Sciences*, 357(1760): 2495–2509, 1999.

[93] J A Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[94] B A Olshausen and D J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[95] 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.

[96] R Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[97] D L Donoho. High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension. *Discrete and Computational Geometry*, 35(4):617–652, 2006.

[98] M Elad and A M Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002.

[99] T Strohmer and R W Heath. Grassmannian frames with applications to coding and communication. *Applied and Computational Harmonic Analysis*, 14(3):257–275, 2003.

[100] P J Huber. *Robust statistics.* Wiley, New York, 1981.

[101] J W Cooley and J W Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19:297–301, 1965.

[102] D J Field. What is the goal of sensory coding? *Neural Computation*, 6(4):559–601, 1994.

[103] A K Jain. *Fundamentals of digital image processing.* Prentice-Hall, 1989.

[104] I T Jolliffe. *Principal component analysis.* Springer, New York, second edition, 2002.

[105] R A DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[106] J B Allen and L R Rabiner. A unified approach to short-time Fourier analysis and synthesis. *Proc. IEEE*, 65(11):1558–1564, 1977.

[107] W B Pennebaker and J L Mitchell. *JPEG still image data compression standard.* Springer, New York, 1993.

[108] D Gabor. Theory of communication. *J. Inst. Electr. Eng*, 93(26):429–457, 1946.

[109] M J Bastiaans. Gabor's expansion of a signal into Gaussian elementary signals. *Proc. IEEE*, 68(4):538–539, 1980.

[110] A Janssen. Gabor representation of generalized functions. *J. Math. Anal. and Applic.*, 83 (2):377–394, 1981.

[111] I Daubechies, A Grossmann, and Y Meyer. Painless nonorthogonal expansions. *J. Math. Phys.*, 27:1271, 1986.

[112] I Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, 1990.

[113] H G Feichtinger and K Gröchenig. Banach spaces related to integrable group representations and their atomic decompositions, part I. *J. Funct. Anal.*, 86(2):307–340, 1989.

[114] H G Feichtinger and K Gröchenig. Gabor wavelets and the Heisenberg group: Gabor expansions and short time Fourier transform from the group theoretical point of view. *Wavelets: A Tutorial in Theory and Applications, C.K. Chiu (ed.)*, pages 359–397, 1992.

[115] H G Feichtinger and K Gröchenig. Gabor frames and time-frequency analysis of distributions. *J. Funct. Anal.*, 146(2):464–495, 1997.

[116] J Wexler and S Raz. Discrete gabor expansions. *Signal processing*, 21(3):207–221, 1990.

[117] S Qian and D Chen. Discrete gabor transform. *IEEE Trans. Sigal Process.*, 41(7):2429–2438, 1993.

[118] J G Daugman. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research*, 20(10):847–856, 1980.

[119] J G Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A*, 2(7): 1160–1169, 1985.

[120] J G Daugman. Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoust. Speech Sigal Process.*, 36(7):1169–1179, 1988.

[121] M Porat and Y Y Zeevi. The generalized Gabor scheme of image representation in biological and machine vision. *IEEE Trans. Patt. Anal. Machine Intell.*, 10(4):452–468, 1988.

[122] P Burt and E Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*, 31(4):532–540, 1983.

[123] I Daubechies. *Ten lectures on wavelets.* Society for Industrial Mathematics, 1992.

[124] Y Meyer and D Salinger. *Wavelets and operators.* Cambridge University Press, 1995.

[125] J Morlet and A Grossman. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.*, 15:723–736, 1984.

[126] Y. Meyer. Principe dŠincertitude, bases hilbertiennes et algèbres dŠopérateurs. *Séminaire Bourbaki*, (662), 1985-86.

[127] P G Lemarie and Y Meyer. Ondelettes et bases hilbertiennes. *Rev. Mat. Iberoamericana*, 2(1-2):1–18, 1986.

[128] I Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math*, 41(7):909–996, 1988.

[129] S G Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Anal. Machine Intell.*, 11(7):674–693, 1989.

[130] S Mallat and W L Hwang. Singularity detection and processing with wavelets. *IEEE Trans. Inf. Theo.*, 38(2):617–643, 1992.

[131] S Mallat and S Zhong. Characterization of signals from multiscale edges. *IEEE Trans. Patt. Anal. Machine Intell.*, 14(7):710–732, 1992.

[132] R R Coifman, Y Meyer, and V Wickerhauser. Wavelet analysis and signal processing. *Wavelets and their Applications*, pages 153–178, 1992.

[133] G Beylkin. On the representation of operators in bases of compactly supported wavelets. *SIAM Journal on Numerical Analysis*, pages 1716–1740, 1992.

[134] G P Nason and B W Silverman. The stationary wavelet transform and some statistical applications. *Lecture Notes in Statistics 103: Wavelets and Statistics (Ed. A. Antoniadis and G. Oppenheim)*, pages 281–299, 1995.

[135] R R Coifman and D L Donoho. Translation-invariant de-noising. *Lecture Notes in Statistics 103: Wavelets and Statistics (Edited by A. Antoniadis and G. Oppenheim)*, pages 125–150, 1995.

[136] S S Chen, D L Donoho, and M A Saunders. Atomic decomposition by basis pursuit. *Technical Report – Statistics, Stanford*, 1995.

[137] 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.

[138] D L Donoho. Wedgelets: nearly minimax estimation of edges. *Annals of statistics*, 27(3): 859–897, 1999.

[139] M B Wakin, J K Romberg, H Choi, and R G Baraniuk. Wavelet-domain approximation and compression of piecewise smooth images. *IEEE Transactions on Image Processing*, 15 (5):1071–1087, 2006.

[140] R M Willett and R D Nowak. Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging. *IEEE Trans. Med. Imaging*, 22(3):332–350, 2003.

[141] V Chandrasekaran, M B Wakin, D Baron, and R G Baraniuk. Representation and compression of multi-dimensional piecewise functions using surflets. *IEEE Transactions on Information Theory*, 55(1):374–400, 2009.

[142] M Yaghoobi, L Daudet, and M E Davies. Parametric dictionary design for sparse coding. *IEEE Transactions on Signal Processing*, 57(12):4800–4810, 2009.

[143] L Ying, L Demanet, and E J Candès. 3D discrete curvelet transform. In *Wavelets XI (Proceedings of SPIE 5914)*, volume 5914, pages 351–361, 2005.

[144] E J Candès and D L Donoho. Continuous curvelet transform: I. Resolution of the wavefront set. *Appl. Comput. Harmon. Anal*, 19(2):162–197, 2005.

[145] A Woiselle, J L Starck, and M J Fadili. New 3D data representations: applications in astrophysics. *Applied and Computational Harmonic Analysis*. To appear.

[146] M N Do and M Vetterli. Contourlets: a new directional multiresolution image representation. In *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, volume 1, pages 497–501, 2002.

[147] R Eslami and H Radha. Translation-invariant contourlet transform and its application to image denoising. *IEEE Transactions on Image Processing*, 15(11):3362–3374, 2006.

[148] E LePennec and S Mallat. Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing*, 14(4):423–438, 2005.

[149] G Peyré and S Mallat. Surface compression with geometric bandelets. In *ACM Transactions on Graphics (Proc. SIGGRAPH 05)*, volume 24, pages 601–608, 2005.

[150] N Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10(3):234–253, 2001.

[151] G Kutyniok and D Labate. Resolution of the wavefront set using continuous shearlets. *Trans. Amer. Math. Soc.*, 361:2719–2754, 2009.

[152] V Velisavljevic, B Beferull-Lozano, M Vetterli, and P L Dragotti. Directionlets: anisotropic multidirectional representation with separable filtering. *IEEE Transactions on Image Processing*, 15(7):1916–1933, 2006.

[153] S Mallat. Geometrical grouplets. *Applied and Computational Harmonic Analysis*, 26(2): 161–180, 2009.

[154] K Skretting and K Engan. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing.* To appear.

[155] K Engan, B D Rao, and K Kreutz-Delgado. Frame design using FOCUSS with method of optimal directions (MOD). *Proceedings of the Norwegian Signal Processing Symposium*, pages 65–69, 1999.

[156] T Blumensath and M Davies. Sparse and shift-invariant representations of music. *IEEE Transactions on Speech and Audio Processing*, 14(1):50, 2006.

[157] P Jost, P Vandergheynst, S Lesage, and R Gribonval. MoTIF: An efficient algorithm for learning translation invariant dictionaries. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, 2006.

[158] O Bryt and M Elad. Compression of facial images using the K-SVD algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–283, 2008.

[159] R R Coifman and M V Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2(2)):713–718, 1992.

[160] H H Szu, B A Telfer, and S L Kadambe. Neural network adaptive wavelets for signal representation and classification. *Optical Engineering*, 31:1907, 1992.

[161] W J Jasper, S J Garnier, and H Potlapalli. Texture characterization and defect detection using adaptive wavelets. *Optical Engineering*, 35:3140, 1996.

[162] M Nielsen, E N Kamavuako, M M Andersen, M F Lucas, and D Farina. Optimal wavelets for biomedical signal compression. *Medical and Biological Engineering and Computing*, 44 (7):561–568, 2006.

[163] R Rubinstein, M Zibulevsky, and M Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *Technical Report – CS Technion*, 2008.

[164] J C Bezdek and R J Hathaway. Some notes on alternating optimization. *Lecture Notes in Computer Science*, 2275:187–195, 2002.

[165] A J Smola and B Scholkopf. Sparse greedy matrix approximation for machine learning. *Proceedings of the 17th International Conference on Machine Learning*, pages 911–918, 2000.

[166] R A Horn and C R Johnson. *Topics in matrix analysis.* Cambridge University Press, 1991.

[167] E J Im. *Optimizing the Performance of Sparse Matrix-Vector Multiplication.* PhD thesis, University of California, 2000.

[168] S F Cotter, R Adler, R D Rao, and K Kreutz-Delgado. Forward sequential algorithms for best basis selection. *IEEE Proceedings – Vision, Image and Signal Processing*, 146(5): 235–244, 1999.

[169] R Rubinstein, M Zibulevsky, and M Elad. Accelerating sparse-coding techniques using sparse dictionaries. *Technical Report – CS Technion*, 2009.

[170] The NIH Visible Human Project. http://www.nlm.nih.gov/research/visible/.

[171] M Aharon, M Elad, and A M Bruckstein. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear Algebra and Its Applications*, 416(1):48–67, 2006.

[172] Sparse K-SVD Toolbox. http://www.cs.technion.ac.il/~ronrubin/software.html.

[173] D S Taubman and M W Marcellin. *JPEG2000: Image compression fundamentals, standards and practice.* Kluwer Academic Publishers Norwell, MA, 2001.

[174] K S Gurumoorthy, A Rajwade, A Banerjee, and A Rangarajan. A method for compact image representation using sparse matrix and tensor projections onto exemplar orthonormal bases. *IEEE Transactions on Image Processing*, 19(2):322–334, 2010.

[175] J Zepeda, C Guillemot, and E Kijak. Image compression using sparse representations and the iteration-tuned and aligned dictionary. *IEEE Transactions on Image Processing.* Submitted.

[176] I Horev. Adaptive image compression using Sparse K-SVD. *Project Report, Signal and Image Processing Lab (SIPL), EE Technion*, 2010.

[177] S Nam, M Davies, M Elad, and R Gribonval. Cosparse analysis modeling-uniqueness and algorithms. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011.

[178] Y Hel-Or and D Shaked. A discriminative approach for wavelet denoising. *IEEE Transactions on Image Processing*, 17(4):443–457, 2008.

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

[180] D L Donoho, I M Johnstone, G Kerkyacharian, and D Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(2):301–369, 1995.

[181] S G Chang, B Yu, and M Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Transactions on Image Processing*, 9(9):1532–1546, 2000.

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

[183] F Abramovich, Y Benjamini, D L Donoho, and I M Johnstone. Adapting to unknown sparsity by controlling the false discovery rate. *The Annals of Statistics*, 34(2):584–653, 2006.

[184] E Haber and L Tenorio. Learning regularization functionals — a supervised training approach. *Inverse Problems*, 19:611–626, 2003.

[185] The CVG Granada image database. http://decsai.ugr.es/cvg/dbimagenes/.

[186] H Takeda, S Farsiu, and P Milanfar. Deblurring using regularized locally-adaptive kernel regression. *IEEE Transactions on Image Processing*, 17(4):550–563, 2008.

[187] Regularized kernel regression-based deblurring. http://users.soe.ucsc.edu/~htakeda/AKTV.htm.

[188] R Neelamani, H Choi, and R Baraniuk. ForWaRD: Fourier-wavelet regularized deconvolution for ill-conditioned systems. *IEEE Transactions on Signal Processing*, 52(2):418–433, 2004.

[189] V Katkovnik, K Egiazarian, and J Astola. A spatially adaptive nonparametric regression image deblurring. *IEEE Transactions on Image Processing*, 14(10):1469–1478, 2005.

[190] Fourier-wavelet regularized deconvolution. http://dsp.rice.edu/software/forward.

קלט מעל מילון שמותאם *באופן פרטני* לתמונה, ואשר נשלח כחלק מהמידע הדחוס. בכך מהווה השיטה טכניקת דחיסה ראשונה מסוגה המבוססת על מילונים נלמדים ומתאימה לתמונות כלליות.

בחלק האחרון של התיזה, אנו חוזרים למודל האנליזה ומתייחסים לשאלה של אימון מילון ייחודי למודל זה. זהו תחום חדש למדי, המושפע רבות מההצלחה הניכרת של מודל הסינטיזה המקביל, והצובר תאוצה בין היתר הודות לתוצאות התיאורטיות עבור מודל האנליזה שהוזכרו מעלה. אנו מציגים בעבודה זו שתי גישות לבעיית האימון, המתאימות לשתי גישות חדשות למודל האנליזה. המשותף לשתי הגישות הוא השימוש במדד דלילות מדוייק – המשמש גם את מודל הסינטיזה – והמאפשר לכן להשתמש בכלים ותובנות שהצטברו במחקר הרב על מודל הסינטיזה, עבור מודל האנליזה.

הגישה הראשונה מאמנת מילון עבור מודל ה-ℓ^0 החדש של האנליזה, שהעניין בו מתעורר בין היתר כתוצאה מההבנות הגיאומטריות שתוארו קודם. על פי תוצאות אלה, האותות המועדפים על מודל האנליזה במקרה ה-ℓ^1 (הנפוץ) מציגים גם תכונות של דלילות ℓ^0. מצב זה דומה מאוד למצב במודל הסינטיזה, בו האותות המועדפים עבור מקרה ה-ℓ^1 דלילים גם במדד ℓ^0. לאור ההצלחה הרבה של מודל הסינטיזה עם ℓ^0, מסקנות אלה מספקות מוטיבציה רבה לחקור את מודל האנליזה עם מדד דלילות זה גם כן. בעבודה אנו מציגים את מודל האנליזה עם מדד ℓ^0, ומציעים אלגוריתמים לייצוג דליל של אות וכן לאימון מילון עבור המודל. תוצאות נסיוניות מאששות את היכולת של אלגוריתם האימון לשחזר ביעילות מילון אנליזה מתוך דוגמאות של אותות דלילים. כמו כן, נסיונות עם תמונות טבעיות מדגימות את יכולת האלגוריתם לזהות מבנים משמעותיים בהתנהגותן של תמונות, כגון התנהגות מקומית וכיוונית.

הגישה השנייה מאמנת זוג מילוני אנליזה וסינטיזה עבור שחזור אותות מבוסס-סף, המפותח בעבודה זו כהכללה של תהליך ניקוי הרעשים הידוע, עבור בעיות שחזור כלליות. אנו מציגים בעבודה אלגוריתם המאמן בו-זמנית את מילוני האנליזה והסינטיזה, מתוך זוגות של דוגמאות מקוריות ופגומות. התהליך המוצע מוביל למנגנון יעיל ופשוט המאפשר פיתוח תהליכי שחזור ותיקון עבור מגוון בעיות. יתרון מרכזי של השיטה הוא שתהליך האימון מקבע את כל הפרמטרים המעורבים בתהליך השחזור, כולל ערכי הסף, ולכן פעולת השחזור פשוטה, יעילה וחסרת פרמטרים. כמו כן, ייחודה של השיטה בכך שהיא אינה דורשת מידע מוקדם לגבי תהליך ההנזקות של האותות, ובמקום זאת, מסיקה את פרטי התהליך מתוך זוגות הדוגמאות עצמם במהלך האימון. סדרת ניסיונות בשחזור תמונות מטישטוש מראה כי השיטה המוצעת מובילה לתוצאות מוצלחות ותחרותיות עם שיטות קיימות, כאשר סיבוכיות השחזור קטנה משמעותית משיטות ייעודיות לשחזור מטישטוש.

מספר דוגמאות בהם מודל האנליזה מספק תוצאות טובות יותר ממודל הסינטיזה המקביל, מה שמעורר עניין מחודש בשיטת האנליזה, שהעניין בה ירד בשנים האחרונות לאור ההתעניינות הרבה בשיטת הסינטיזה החדשה יותר. התוצאות של מחקר זה מדגימות את הפוטנציאל והחשיבות שבמודל האנליזה, ומספקות מספר תובנות שימושיות לגבי המודל.

בחלק המרכזי של המחקר אנו מתמקדים במרכיב העיקרי של שני המודלים הללו – המילון. מרכיב זה מייצג למעשה את התגשמות כל הידע וההבנה שלנו על התנהגותם של האותות, ובחירתו של המילון מכתיבה את הצלחת המודל כולו. בתיזה אנו דנים בשתי הגישות המרכזיות הרווחות לתכנון מילונים – אנליזה הרמונית ולמידה חישובית – כאשר אנליזה הרמונית שואפת לקרב את האותות המעניינים על ידי משפחה של *פונקציות מתימטיות* בעלות התנהגות ידועה היטב, ואילו למידה חישובית שואפת ללמוד את התנהגות האותות מתוך *אוסף דוגמאות* המציג את ההתנהגות הרצויה של האותות. היתרונות העיקריים של הגישה הראשונה הינם היכולת להוכיח *אופטימאליות* של המילון עבור משפחת האותות המקורבת, וכן האפשרות לפתח אלגוריתמים יעילים למימוש המילון, המתקבל על פי רוב באופן אנליטי. לעומת זאת, היתרון המרכזי של גישת הלימוד הינו ההתאמה הטובה יותר של המילון המתקבל לאותות האמיתיים בהם המערכת מטפלת, מה שמוביל במקרים רבים לתוצאות טובות יותר בפועל. לאחרונה צוברת תאוצה מגמה נוספת, השואפת *למזג* את היתרונות של שתי הגישות באמצעות *מילונים מבניים*. מילונים מבניים הינם מילונים המאופיינים על ידי מספר פרמטרים הקטן ממספר האיברים במטריצת המילון, ומאפשרים, באמצעות תכנון נבון של מבנה המילון, להשיג מגוון של תכונות והתנהגויות רצויות.

בעבודה זו אנו מציעים את *המילון הדליל* כמילון מבני השואף למזג בין היתרונות של שתי הגישות לתכנון מילונים. המילון הדליל מתקבל כהרכבה של מילון אנליטי יעיל עם מטריצה דלילה נלמדת, ובאופן זה מספק שילוב של יעילות, מבניות, וכן יכולת לימוד של המילון. כמו כן, באמצעות שינוי מספר המקדמים במטריצה הדלילה, המילון הדליל מאפשר לנוע באופן כמעט רציף בין מילון אנליטי לחלוטין (מטריצה דלילה מאוד) לבין מילון מאומן מלא (מטריצה צפופה). בכך, מספק המבנה המוצע גשר של ממש בין הגישה האנליטית והגישה הנלמדת. יתרונות נוספים של המילון הדליל הינם יכולת הכללה משופרת בתהליך האימון, קיום ייצוג יעיל של המילון, וכן היכולת לכפות מבנים משמעותיים על המילון, כתלות ביישום. אנו מדגימים את השימושיות של המילון הדליל עבור ניקוי רעשים בתמונות טומוגרפיה ממוחשבת (CT), ומראים כי המבנה הדליל מספק תוצאות שקולות או טובות מהמבנה הנלמד המקביל, כאשר המבנה הדליל מהיר בהרבה. בפרק נוסף אנו מתארים יישום של המילון הדליל לדחיסה, בצורת מערכת שלמה לדחיסת תמונות כלליות המבוססת על המילון הדליל. ייחודה של השיטה בכך שהיא מקודדת כל תמונת

תקציר

מודלים של אותות הם מאבני הבניין של תחום עיבוד האותות והתמונות המודרני, ומשמשים למגוון רחב של מטלות שחזור, ייצוג, ניתוח ושיפור. מודל של אות מספק *תאור מתמטי* של התנהגות האותות המעניינים במערכת באופן המאפשר להבדיל אותם מכלל האותות האפשריים. לאור המורכבות הרבה של אותות טבעיים, מודלים אלו הינם *מקורבים* מיסודם. המטרה של תחום מידול האותות הינו לפתח מודלים מדוייקים ככל האפשר, המתארים בנאמנות את התנהגותם של אותות אמיתיים.

מבין מגוון המודלים שפותחו בסיפרות, *מודלים מבוססי מילון* נוחלים הצלחה רבה במגוון רחב של יישומים. מודלים אלו מאמצים גישה של פירוק האות למרכיבים, ומתארים אותו באמצעות *מילון* של אותות בסיסיים הידועים כ*אטומים*. שתי גישות משלימות רווחות בסיפרות למידול אותות מבוסס מילון. *מודל האנליזה* מתאר אותות במונחים של *מכפלות פנימיות* של האותות עם האטומים במילון. לעומתו, *מודל הסינטיזה* נוקט גישה הפוכה, ומתאר אותות כ*צירופים ליניאריים* של האטומים. הכוח הבסיסי ביסודם של שני מודלים אלו הינו *דלילות* – דהיינו, דעיכה מהירה של מקדמי הייצוג מעל המילון. שני המודלים מוכיחים את עצמם כיעילים ביותר במגוון רחב של בעיות עיבוד אות ותמונה, ומביאים לתוצאות מובילות בתחומן ביישומים רבים, ביניהם הסרת רעש, דחיסה, שחזור צבע מפסיפס, השלמת חורים, הגדלה, חישה דלילה (compressed sensing), ועוד.

עבודה זו עוסקת במספר פנים של מודלי האנליזה והסינטיזה. העבודה פותחת בשאלת *הקשר* בין שתי הגישות מבוססות-המילון, המתעוררת עקב הדמיון המתמטי הרב בין השתיים. במקרה ההפיך, כלומר כאשר המילון מהווה מטריצה הפיכה, קל להראות ששני המודלים מתלכדים. עם זאת, במקרה הבלתי הפיך – כאשר מספר האטומים עולה על מימד האות – אנו מראים תוך שימוש בכלים גיאומטריים כי על אף הדמיון האלגברי, שתי הגישות למעשה שונות מהותית זו מזו, עם פער ניכר שמפריד בין השתיים. בפרט, אנו מראים כי לכל זוג מודלים של אנליזה וסינטיזה בעלי גודל אסימפטוטי דומה, קיים בהכרח מספר עצום (מעריכי) של אותות עליהם שני המודלים יתנהגו באופן שונה מהותית. כמו כן, אנו מביאים

אני חש זכות גדולה במיוחד להוקיר ולהודות באופן אישי לחברי הקרוב והיקר יעקב כגן. אני מודה ליעקב מכל לב על השנים הרבות של חברות בלתי מסוייגת, עזרה מסורה, שיחות מרתקות, ותמיכה אינסופית. תודתי העמוקה ביותר שלוחה ליעקב. לא יכולתי לבקש חבר טוב ממנו.

לבסוף, עבודה זו לא הייתה מתאפשרת ללא המשפחה המסורה, התומכת, החמה והמעודדת ביותר. להוריי, ישראל ובלהה, אני שולח את תודתי והערכתי העמוקות ביותר על התמיכה הבלתי נדלית, העידוד העיקש, העצות האיתנות, והאהבה ללא גבול. הייתם לי לעוגן, למשענת, למורי דרך, למודל ללכת לאורו, ועל כל אלה ועוד אני מוקיר לכם תודה עמוקה. לאחיותיי, יעל, טלי ותמי, תודה על התמיכה, ההבנה, ההערכה, ועל הרגעים השמחים הרבים. תודות מקרב לב לדודותיי ניצה ונורית על העצות הרבות והחשובות, על העידוד, התמיכה, העזרה, והאהבה האדירה. תודה עמוקה גם לסבי וסבתי האהובים, אליוט ולוסי, על האמונה שלהם בי, על התמיכה הרבה, על העזרה, ועל האהבה המסורה. תודות לכל בני משפחתי שאת שמם לא ציינתי כאן באופן אישי – אהבתי שלוחה לכולכם.

אני מודה לטכניון ולמשרד המדע על התמיכה הכספית הנדיבה

המחקר נעשה בהנחיית פרופ׳ מיכאל אלעד בפקולטה למדעי המחשב בטכניון

תודות

כתיבת חיבור זה התאפשרה הודות לעזרתם הנדיבה של אנשים רבים, ואני שמח ונרגש להוקיר להם תודה כאן.

בראש ובראשונה, התודה וההוקרה הגדולות ביותר שמורות למורה ולמנחה שלי, פרופ׳ מיכאל אלעד. מיקי, זו הייתה זכות גדולה לעבוד עם אדם כה מסור, מלומד, מתחשב, אבחנתי, מעשיר ומעורר השראה. אין מילים לתאר את הערכתי לעזרתך הרבה, לתמיכתך, לעצותיך ולהדרכתך האיתנה לכל אורך דרכי האקדמית, ועל כל אלו ועוד, תודתי העמוקה נתונה לך.

תודה מיוחדת שלוחה גם לדר׳ מיכאל ציבולבסקי, עימו הייתה לי ההזדמנות הנדירה לשתף פעולה. אני מודה מקרב לב לדר׳ ציבולבסקי על השיחות הפוריות והרעיונות מאירי העיניים, על העזרה הנדיבה, התובנות המעמיקות, והיחס החם לאורך שנותיי בטכניון.

אני רוצה לנצל הזדמנות זו ולהודות למספר אנשים מהם למדתי רבות במהלך השנים, ואשר אני חב להם חלק ניכר מן הידע וההבנה שצברתי עד היום. תודתי והערכתי העמוקות נתונות לפרופ׳ אלפרד ברוקשטיין, פרופ׳ רון קימל, פרופ׳ אברהם סידי, פרופ׳ עירד יבנה, פרופ׳ שמואל פלג ופרופ׳ נפתלי תשבי, אשר העשירו אותי בידע מרתק ויקר מפז, והטמיעו בי את הלהט למדע.

תודות חמות שלוחות לירון חונן וליאנה כץ מן המעבדה לעיבוד תמונה גיאומטרי, על התמיכה הנפלאה והסיוע יוצא הדופן לאורך השנים. תודות גם לפרופ׳ דוד מלאך ולנמרוד פלג מן המעבדה לעיבוד אותות ותמונות, על שיתופי הפעולה הפוריים והמועילים. לבסוף, תודה מיוחדת שמורה לירדנה קולט, מזכירת לימודי מוסמכים בפקולטה למדעי המחשב, על העזרה המסורה והתמיכה האינסופית לאורך כל שנותיי בטכניון.

אני רוצה להודות מאוד לכל חבריי ומכריי מהטכניון, אשר הפכו את תקופת הלימודים המאתגרת לחוויה מהנה ומעשירה במיוחד. מבין אלו אציין באופן אישי את רועי אנגלברג, יניב חמו, קרן וואקנין, סבטלנה רבוי ואירנה זברסקי, אשר כל אחד מהם הינו חבר יקר, מוערך וקרוב, שהשפיע באופן משמעותי על חיי. תודה מיוחדת שלוחה לידידי ועמיתי אורי בריט, על השיחות המועילות, שיתופי הפעולה הפוריים, והעזרה הרבה לאורך השנים.

מודלים דלילים של אנליזה וסינטיזה בעיבוד תמונה

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

רן רובינשטיין

מוגש לסנט הטכניון – מכון טכנולוגי לישראל

אלול תשע״א חיפה ספטמבר 2011

מודלים דלילים של אנליזה וסינטיזה בעיבוד תמונה

רן רובינשטיין