Deconvolution and Sparsity based Image Restoration

Muhammad Hanif

11 November 2015

A thesis submitted for the degree of Doctor of Philosophy of The Australian National University
Declaration

The work presented in this thesis is my own, except where stated otherwise with a reference. To my best knowledge this work do not contains substantial material previously published or accepted for the degree of higher learning by me or by someone else. This work has been carried out in collaboration with my supervisor Dr. Abd-Krim Seghouane at the College of Engineering and Computer Sciences, the Australian National University (ANU) and at National ICT Australia (NICTA) Canberra Research Laboratory. Most of the technical discussions presented in this thesis are based on the following publications:

- Muhammad Hanif and A. K. Seghouane. **Blind Image Deblurring Using Non-negative Sparse Approximation.** IEEE Int. Conf. on Image Processing (ICIP-2014)

- Muhammad Hanif and A. K. Seghouane. **An Effective Image Restoration using Kullback-Leibler Divergence.** IEEE Int. Conf. on Image Processing (ICIP-2014)


- Muhammad Hanif and A. K. Seghouane. **An EM Based Hybrid Fourier-Wavelet Image Deconvolution Algorithm** IEEE Int. Conf. on Image Processing (ICIP-2013)


Muhammad Hanif
11 November 2015
To my family and parents with love. Especially, to my late mom, Saleha who passed away very early in her life, without enjoying the fruits of her work.
Acknowledgments

I am indebted to many people who made this thesis possible. Firstly, I would like to offer thanks to my family, friends, colleagues, and teachers who believed in me and provided their continuous support and encouragement.

Specially my sincerest thanks to my supervisor, Dr. Abd-Krim Seghouane for his unwavering support and invaluable guidance, which were essential to the fruitful completion of the thesis. Working with him has been a great pleasure and I would always appreciate his concern towards my growth as a researcher. I express my deepest thanks to my supervisor at Australian National University (ANU) Dr. Jochen Trumpf for helpful advice and financial arrangements, and Dr. Wray Buntine for being part of my supervisory panel. Special thanks to my colleagues Dr. Adnan Shah, and Muhammad Usman Khalid for all their suggestions, support, and discussions.

Finally, but by no means the least, I my grateful to my family and parents for their unconditional support and care during the entire time of PhD research. To my loving father Abdul Hanan, for his endless moral support and having full belief in my abilities. To my loving wife Sadia for her understanding, tolerance, and sacrifice throughout my study life. Many thanks to my lovely children Ume Salma, Muhammad Ahmad and Muhammad Ibrahim who suffered from my absence many weekends due to my busy schedules.
Abstract

Deconvolution and sparse representation are the two key areas in image and signal processing. In this thesis the classical image restoration problem is addressed using these two modalities. Image restoration, such as deblurring, denoising, and inpainting belongs to the class of ill-posed linear inverse problems, which requires a proper regularization for a credible solution. The aim is to develop techniques that are stable, practical and require a minimum amount of prior knowledge. The two main approaches that we focused upon in this thesis are image deconvolution for blurred image restoration and dictionary learning algorithms for sparse image denoising and inpainting.

In the first approach, iterative least square and maximum likelihood based deconvolution methods, are derived for image deblurring application. Three novel methods are presented i) a hybrid Fourier-wavelet deblurring (HFW) method based on expectation maximization (EM) approach, ii) sparse non-negative matrix approximation (SNMA), and iii) Kullback-Leibler divergence minimization (KLD). For HFW, the main objective function was split into Fourier domain deconvolution and wavelet domain denoising, to avoid the computational burden of handling blurring matrix in wavelet domain. Further, the wavelet coefficients were modelled using the class of Gaussian scale mixture (GSM) model, which represent the heavy tailed distribution. The SMNA and KLD are designed for a more challenging task of blind image deconvolution (BID), where either no or very little prior information about the original image and blurring operator is provided. In SMNA an explicit blur estimation and strict positive constraint on the observed and original image, are utilized to retrieve the latent original image. The third method is derived using successive minimization of KLD between a model and a desired family of probability distributions. This algorithm can be viewed as cascaded EM in information geometric terms.

In the second approach, dictionary learning methods with sparsity constraint on original image, are designed to address the image denoising and inpainting problem. Recently, the sparse representation emerged as a useful regularization in ill-posed linear inverse problems. The main assumption in this direction, is that the original image has a sparse representation over some dictionary. Three novel dictionary learning algorithms are outlined. In the first method, an orthogonal dictionary based
on the profile-likelihood estimate is derived with single Eigen decomposition. Most of the dictionary learning algorithms confined the sparsity constraint to the sparse coding stage. In the second method, we look at the impact of enforcing the sparsity constraint also in the dictionary update stage. Within this framework different constraints such as smoothness of dictionary atoms and other can also be enforced to enhance the dictionary strength. In the last method, we looked into the double sparsity constraint, where the strength of explicit and implicit dictionaries is combined for efficient data representation. This model is based on a sparse representation of dictionary atoms over an implicit base dictionary. The advantage of this dictionary structure is evident from the experimental simulations.
Contents

Acknowledgments vii

Abstract ix

1 Introduction 1
   1.1 Thesis statement .................................. 4
   1.2 Mathematical model ................................ 5
   1.3 Organization ....................................... 6

2 Part 1: Image Deconvolution 9
   2.1 What is blur? ....................................... 10
   2.2 Image deconvolution ................................. 11
      2.2.1 Non-Blind deconvolution ...................... 12
      2.2.2 Blind image deconvolution ................... 12
   2.3 Literature Review .................................. 13

3 EM based Hybrid Fourier-Wavelet Image Deblurring 17
   3.1 Expectation Maximization Algorithm ............... 18
      3.1.1 EM in image processing ....................... 20
   3.2 Gaussian Scale Mixture ............................. 21
   3.3 Related Work ...................................... 22
   3.4 Proposed Algorithm ................................ 24
      3.4.1 Wavelet Based Deconvolution ................. 25
      3.4.2 Hybrid Fourier Deconvolution ............... 26
      3.4.3 E-Step ......................................... 27
      3.4.4 M-Step ......................................... 27
      3.4.5 E2-Step ........................................ 27
      3.4.6 M2-Step ........................................ 29
   3.5 Simulation Results ................................ 29
   3.6 Conclusion ......................................... 33

4 Image Debluring with Sparse Non-Negative Matrix Factorization 35
   4.1 Low rank approximation ............................ 36
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2</td>
<td>Non-negative Matrix Factorization (NMF)</td>
<td>37</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Sparse NMF</td>
<td>40</td>
</tr>
<tr>
<td>4.3</td>
<td>Related Work</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Proposed Algorithm</td>
<td>43</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Non-negative Sparse Approximation</td>
<td>44</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Blur Kernel Estimation</td>
<td>45</td>
</tr>
<tr>
<td>4.5</td>
<td>Experimental Analysis</td>
<td>46</td>
</tr>
<tr>
<td>4.6</td>
<td>Conclusion</td>
<td>48</td>
</tr>
<tr>
<td>5</td>
<td>Information Geometric Blind Image Debluring</td>
<td>51</td>
</tr>
<tr>
<td>5.1</td>
<td>Kullback-Leibler Divergence</td>
<td>52</td>
</tr>
<tr>
<td>5.1.1</td>
<td>KLD and Likelihood Estimation</td>
<td>53</td>
</tr>
<tr>
<td>5.2</td>
<td>Related literature</td>
<td>54</td>
</tr>
<tr>
<td>5.3</td>
<td>Proposed Approach</td>
<td>56</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Estimation by successive minimization of KL divergence</td>
<td>58</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Blind image restoration by KL divergence minimization</td>
<td>62</td>
</tr>
<tr>
<td>5.4</td>
<td>Experimental Results</td>
<td>66</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Non-Blind Deconvolution</td>
<td>67</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Blind Deconvolution</td>
<td>69</td>
</tr>
<tr>
<td>5.5</td>
<td>Conclusion</td>
<td>74</td>
</tr>
<tr>
<td>6</td>
<td>Part 2: Sparse Representation based Image Restoration</td>
<td>75</td>
</tr>
<tr>
<td>6.1</td>
<td>Dictionary learning</td>
<td>78</td>
</tr>
<tr>
<td>6.1.1</td>
<td>MOD</td>
<td>80</td>
</tr>
<tr>
<td>6.1.2</td>
<td>K-SVD</td>
<td>80</td>
</tr>
<tr>
<td>6.2</td>
<td>Dictionary learning in image processing</td>
<td>81</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Image denosing</td>
<td>82</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Image in-painting</td>
<td>83</td>
</tr>
<tr>
<td>7</td>
<td>Maximum Likelihood Based Dictionary Learning for Image Restoration</td>
<td>85</td>
</tr>
<tr>
<td>7.1</td>
<td>Related Methods</td>
<td>86</td>
</tr>
<tr>
<td>7.2</td>
<td>Problem Formulation</td>
<td>87</td>
</tr>
<tr>
<td>7.3</td>
<td>Proposed Method</td>
<td>89</td>
</tr>
<tr>
<td>7.4</td>
<td>Application Results</td>
<td>90</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Filling-in Missing Pixels</td>
<td>91</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Noisy Image Restoration</td>
<td>92</td>
</tr>
<tr>
<td>7.5</td>
<td>Conclusion</td>
<td>93</td>
</tr>
<tr>
<td>Chapter</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>8</td>
<td>Penalized Rank-1 Matrix Approximation for Dictionary Learning</td>
<td>95</td>
</tr>
<tr>
<td>8.1</td>
<td>Related Work</td>
<td>96</td>
</tr>
<tr>
<td>8.2</td>
<td>Mathematical formulation</td>
<td>99</td>
</tr>
<tr>
<td>8.3</td>
<td>Sequential dictionary update with enforced sparsity</td>
<td>101</td>
</tr>
<tr>
<td>8.4</td>
<td>Sequential dictionary update with enforced constraints</td>
<td>103</td>
</tr>
<tr>
<td>8.4.1</td>
<td>Update with smooth dictionary atoms</td>
<td>103</td>
</tr>
<tr>
<td>8.4.2</td>
<td>Update with enforced sparsity and smooth dictionary atoms</td>
<td>106</td>
</tr>
<tr>
<td>8.5</td>
<td>Experimental results</td>
<td>107</td>
</tr>
<tr>
<td>8.5.1</td>
<td>Synthetic data</td>
<td>108</td>
</tr>
<tr>
<td>8.5.2</td>
<td>Image denoising</td>
<td>111</td>
</tr>
<tr>
<td>8.5.3</td>
<td>Image in-painting</td>
<td>112</td>
</tr>
<tr>
<td>8.5.4</td>
<td>Image compression</td>
<td>114</td>
</tr>
<tr>
<td>8.6</td>
<td>Conclusion</td>
<td>115</td>
</tr>
<tr>
<td>9</td>
<td>Double Sparsity for Image Restoration</td>
<td>119</td>
</tr>
<tr>
<td>9.1</td>
<td>Motivation</td>
<td>120</td>
</tr>
<tr>
<td>9.2</td>
<td>Mathematical Modeling</td>
<td>121</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Double Sparsity</td>
<td>123</td>
</tr>
<tr>
<td>9.3</td>
<td>Proposed Algorithms</td>
<td>125</td>
</tr>
<tr>
<td>9.3.1</td>
<td>Sequential learning of sparse dictionaries</td>
<td>125</td>
</tr>
<tr>
<td>9.3.2</td>
<td>Sequential dictionary learning with smooth constraint</td>
<td>129</td>
</tr>
<tr>
<td>9.4</td>
<td>Experimental results</td>
<td>131</td>
</tr>
<tr>
<td>9.4.1</td>
<td>Dictionary training</td>
<td>131</td>
</tr>
<tr>
<td>9.4.2</td>
<td>Image in-painting</td>
<td>133</td>
</tr>
<tr>
<td>9.4.3</td>
<td>Image denoising</td>
<td>135</td>
</tr>
<tr>
<td>9.5</td>
<td>Conclusion</td>
<td>138</td>
</tr>
<tr>
<td>10</td>
<td>Conclusion</td>
<td>141</td>
</tr>
<tr>
<td>10.1</td>
<td>Summary</td>
<td>141</td>
</tr>
<tr>
<td>10.2</td>
<td>Future Work</td>
<td>143</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
<td>145</td>
</tr>
<tr>
<td>A</td>
<td>Maximum Likelihood estimation via Kullback-Leibler divergence</td>
<td>147</td>
</tr>
<tr>
<td>A.1</td>
<td>Minimization w.r.t. $p(x)$</td>
<td>148</td>
</tr>
<tr>
<td>A.2</td>
<td>Minimization w.r.t. $\theta$</td>
<td>148</td>
</tr>
<tr>
<td>A.2.1</td>
<td>Minimization w.r.t. $p(s)$</td>
<td>149</td>
</tr>
<tr>
<td>A.2.2</td>
<td>Minimization w.r.t. $\theta_2$</td>
<td>150</td>
</tr>
<tr>
<td>B</td>
<td>Penalized rank-1 matrix approximation</td>
<td>151</td>
</tr>
</tbody>
</table>
# List of Figures

1.1 Original images used for the simulations results. ........................................ 7
1.2 Thesis layout. ......................................................................................... 8

2.1 Blurred and noisy image observation. ...................................................... 10

3.1 \textit{Exp-2: Visual comparison.} ................................................................. 31
3.2 \textit{Exp-3: ISNR for different BSNR levels.} ............................................... 32

4.1 \textit{Exp-4: Visual comparison.} ................................................................. 48
4.2 \textit{Exp-2: ISNR for different BSNR levels.} ............................................... 48

5.1 Non-blind deconvolution visual comparisons for the \textit{Exp. 2}. .................... 69
5.2 Non-blind ISNR for different BSNR levels for \textit{Exp. 5} ................................. 70
5.3 Blind deconvolution visual comparison for \textit{Exp. 1} ................................... 72
5.4 Blind deconvolution ISNR for different BSNR levels for \textit{Exp. 3} ............... 72
5.5 Estimated image and PSF’s ........................................................................ 73

6.1 Sparse representation of a data set $X$ in terms of $D$ and $A$. ..................... 76
6.2 Learned Dictionary: (a) images, (b) the learned dictionary. ......................... 82

7.1 Missing Pixel: Visual comparison. .............................................................. 91
7.2 Image Restoration: Visual comparison. ..................................................... 93

8.1 Average dictionary atoms retrieved at each iteration for $s=3$, SNR=20dB. .... 109
8.2 Average dictionary atoms retrieved at each iteration for $s=4$, SNR=30dB. ... 110
8.3 Comparison in terms of data-fitting for different sparsity levels. ................... 110
8.4 Visual comparison for image restoration. .................................................. 113
8.5 Missing Pixel: Visual comparison. ............................................................ 115
8.6 Visual comparison for image compression example ...................................... 116

9.1 Visual comparison of text removal. ............................................................ 134
9.2 Visual comparison for random scratches removal. ..................................... 135
9.3 Visual comparisons for Fill-in missing image pixels. ................................... 136
9.4 Double sparsity: Image denoising visual comparison. .................................. 138
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Experimental Setup</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>ISNR (in dB) Comparisons with other state of the art methods</td>
<td>31</td>
</tr>
<tr>
<td>3.3</td>
<td>ISNR for Levin Data</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>Proposed algorithm</td>
<td>45</td>
</tr>
<tr>
<td>4.2</td>
<td>Experimental Setup</td>
<td>46</td>
</tr>
<tr>
<td>4.3</td>
<td>ISNR (in dB) Comparisons with other state of the art methods</td>
<td>47</td>
</tr>
<tr>
<td>4.4</td>
<td>ISNR for Levin Data</td>
<td>49</td>
</tr>
<tr>
<td>5.1</td>
<td>Experimental Setup</td>
<td>66</td>
</tr>
<tr>
<td>5.2</td>
<td>Non-Blind Deconvolution ISNR Comparison</td>
<td>68</td>
</tr>
<tr>
<td>5.3</td>
<td>Blind Deconvolution ISNR Comparison</td>
<td>71</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison for Levin data in terms of SSE</td>
<td>73</td>
</tr>
<tr>
<td>5.5</td>
<td>Comparison with VBSK</td>
<td>74</td>
</tr>
<tr>
<td>5.6</td>
<td>Comparison of estimated PSF in terms of PSNR</td>
<td>74</td>
</tr>
<tr>
<td>7.1</td>
<td>Maximum Likelihood Orthogonal Dictionary Learning algorithm</td>
<td>90</td>
</tr>
<tr>
<td>7.2</td>
<td>Filling-in missing pixels comparison in terms of SSD</td>
<td>92</td>
</tr>
<tr>
<td>7.3</td>
<td>Noisy image restoration ISNR Comparisons</td>
<td>92</td>
</tr>
<tr>
<td>8.1</td>
<td>Sequential dictionary learning with enforced sparsity</td>
<td>104</td>
</tr>
<tr>
<td>8.2</td>
<td>Sequential dictionary learning with regularized dictionary atoms</td>
<td>105</td>
</tr>
<tr>
<td>8.3</td>
<td>Dictionary learning with regularized atoms and enforced sparsity</td>
<td>107</td>
</tr>
<tr>
<td>8.4</td>
<td>Average number of restored dictionary atoms</td>
<td>108</td>
</tr>
<tr>
<td>8.5</td>
<td>ISNR comparisons for noisy image restoration</td>
<td>111</td>
</tr>
<tr>
<td>8.6</td>
<td>SSD comparison for fill-in missing pixels example</td>
<td>114</td>
</tr>
<tr>
<td>8.7</td>
<td>Comparison in terms of PSNR for image compression example</td>
<td>116</td>
</tr>
<tr>
<td>9.1</td>
<td>Double sparsity controlled dictionary learning algorithm</td>
<td>128</td>
</tr>
<tr>
<td>9.2</td>
<td>The double sparsity algorithm with a smooth constraint</td>
<td>132</td>
</tr>
<tr>
<td>9.3</td>
<td>Scratches and text removal comparison in terms of SSD</td>
<td>134</td>
</tr>
<tr>
<td>9.4</td>
<td>Double sparsity: Fill-in missing pixels comparison in terms of SSD</td>
<td>136</td>
</tr>
<tr>
<td>9.5</td>
<td>Double sparsity: Noisy image restoration ISNR Comparisons</td>
<td>137</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

"Essentially, all models are wrong, but some are useful."
— George E. P. Box

‘All is number’, Pythagoras once taught. In his philosophy it was numbers and their nature that make things clear, either by them self or in their relation to other things. The current digital era with all its revolution in the form of digital computers, digital cameras, digital telephones, digital television, and digital processors attest to the foresight of the Pythagoras theory. In particular this is reflected in the advancement in sensor technology, storage efficiency and processing capabilities in the last few decades. These revolutionized the digital image processing research and its applications in different fields of science. Images are the primary pool of information in nearly every branch of science most importantly, medical, astronomy, remote sensing, surveillance, computer vision, and photography. From smart phones in our pockets to the space ship in space, images are everywhere.

Vision is no doubt an important human sense; the world for us is actually what we see. In human learning and perception, images and their information content play the central role. Our mind began its first lesson from our very first image, the clearer the images became the more we started understanding. The human brain can be seen as the first image processing unit, where the shapes in the visual frames were segmented and interpreted accordingly, as we increased in age.

The genesis of vision concept can be traced to the work of 11th century Arab polymath, Ibn al-Haytham, also known as Alhazen or Alhacen, whose scientific exploration of the vision system, significantly impacted on the initial concepts of the human vision system. For the very first time Ibn al-Haytham introduced the ‘scientific methods’ to question the Greek theories and used experimental
Introduction

evidence to develop accurate theory of vision Smith [2005] Hockney [2006]. Ibn al Haytham negates the Greek theories of vision of that time, known as extramission, intromission and the combination of the two, by empirical understanding, deduced from his scientific experimentation Smith [2001]. His seminal work documented in seven volume treatises, Kitab-al-Manazir (Book of Optics), laid the foundation of the modern theory of optics and explains the principal concepts related to the two dimension (2D) pictorial representations of three dimension (3D) scenes Lindberg [1981]. In this book Ibn al Haytham constantly linked sight and vision with the properties of light, along with the behaviour of light and the physiology of the eye. His work and design became the root source of optics understanding in the west up to the 17th century, until the Kepler theory of retinal image Lindberg [1967].

Modern science also adopted the human visual learning process and images become the primary source of information in almost every research field. The use of images by humans can be traced back a thousand years or more. From the beginning, images hold the vital position in information exchange, either in the form of representing an idea, a thought or an emotion or testimony to an important moment or event. The rapid developments in technology made the image processing convenient and less expensive and contributed to the overwhelming growth of digital imaging systems.

The importance of images in this modern era is well reflected from its applications range; entertainment industry (including photography, television, fashion magazine), remote sensing, medical, astronomy, surveillance, microscopy, classification, optics, geophysical prospecting, super resolution and more.

Images are complex signals and are difficult to handle mathematically. With growing applications and interest, the imaging sensors, image storage and processing become an essential part of the research community. The imaging science basically consist of three main components sensing, processing and interpretation Chan and Shen [2005]. The sensing or acquisition are composed of imaging systems and all the related physical mechanisms that generate the image, for example X-rays machines, computed tomography (CT) scanners, functional magnetic resonance imaging (fMRI) detectors and Diffusion tensor imaging (DTI) systems. Image processing includes the computational methods and techniques applied on the data presented by the image in search of some specific information, for example methods for image denoising, object segmentation and classification etc. The last part of the interpretation deals with the decision based on the extracted or processed information from the image, which is mostly the ultimate goal of whole process.
Although a lot of effort and work has been done in creating the efficient and flaw-
less imaging systems, the contamination of image data by corruption or degrada-
tion, such as noise, blurring, aliasing, illumination and colour imperfections, cannot
be ruled out and are always present in one form or in other. The spatial resolution
and degradations due to imaging system, environment, storage or transmission are
always challenging in image processing applications. Degradation or noise is an un-
wanted signal that can interfere with the actual or original measured signal. From
one angle, the degradation itself acts as a signal that conveys information regarding
the type, source and extent of the corruption, for example blurring indicates lowpass
filtering or out-of-focus camera setting. Three possible directions can be adopted to
cope with such degradation in images, hardware (array and sensors), imaging en-
vironment (imaging systems, conditions), and software routines (image processing
tools). All these avenues have been extensively explored by research community
and there are many success stories. In these three possible approaches, the software
solution i.e. image processing methods, is cost efficient and has been exploited to
improve the image quality during the last few decades.

Nevertheless rapid improvement has been observed in digital imaging, yet the ob-
servation process in never perfect. The image restoration is still an active and open
research problem. There are a number of factors which keep the image restoration
still alive:

- Physical limitations of imaging system, sensor and storage: the denser CCD ar-
  rays in modern cameras with a small number of pixels cause noise. The longer
  exposure can reduce the noise effect but on the other side cause a blurring
effect. Also the compression techniques in storage contribute to the problem.

- Imaging environment: sometimes it is impossible, costly or dangerous to create
  the perfect and stable imaging environment, like in medical imaging, the move-
  ment of the patient or part of the body, heart beat, or blood flow is out of our
  control. Also in the case of astronomy, remote sensing or deep water imaging it
  is very costly to use the proper imaging equipment. Similarly, problems occur
  when dealing with ancient or antique photographs, which cannot be retake.

- Image transmission: even with optimized imaging system in perfect conditions,
  images can be degraded in the transmission process.

In view of the above mentioned limitations and the increase in the usage of image
based information it is no surprise that image restoration is an active and important
Introduction

Image restoration mainly refers to the group of methods and techniques, specifically designed to remove or reduce the degradations in the observed image.

Image restoration is basically an ill-posed and/or ill-conditioned linear inverse problem, which requires strong and meaningful assumptions on the latent image for a plausible estimate. The efficiency of an image restoration method is related to its ability of identifying and modelling the degradation process, and utilizing these characteristics to differentiate the latent image information from the corruption. How much an image can be restored?, it totally depends on the knowledge about the original image, image contents, degree of degradation, corruption source, and accuracy of problem modelling and implementation. The development made in the image restoration domain can be attributed to better insight into image content, modelling and appropriate deployment of these models, in the relevant applications [Gonzalez and Woods 2002]. This model adaptation starts from the simple L2 norm local differences smoothing, spanning through edge preserving smoothness and wavelet based sparsity to the very recent sparse and redundant based modelling, which produced state-of-the-art results in many image processing applications. The evolution and employment of these reliable and strong modelling techniques enable the image processing community to effectively handle the problems, ranging from denoising, sampling, deblurring, inpainting, and reconstruction in inverse problems, all the way to more advance issues of clustering, classification, detection and compression.

In literature parametric and non-parametric methods, have been proposed to address image degradation problems. In parametric methods the estimation is based on some specific imaging model, where the model parameters are treated as unknowns and are computed from the noisy observation. In contrast, non-parametric methods follow the regression theory and structure the estimation model from the observed noisy data.

1.1 Thesis statement

In this thesis we are focused on image restoration degraded by blur and additive noise. Image denosing and deblurring are classical and challenging image restoration problems. The two main techniques we adopted to address these problem were image deconvolution and sparse representation. Broadly speaking, this thesis can be divided in two main sections; 1) deconvolution based methods, where blurring and noise are the main degradation sources; and 2) sparse representation based image
restoration, where noise and image inpainting are the main issues to address.

## 1.2 Mathematical model

Mathematical model is a description of a system using mathematical concepts and language to express the relation between input(s) and output(s). Modelling the problem, in some logical form, provides a platform to represent the prior knowledge about the problem in hand and the path we are adopting to address that problem. Mathematical models are used to reshape the concepts about the underlying problem in such a way that it can distinguish the ‘original’ signal or data from the ‘corruption’ or unwanted ‘noise’ signal. This separation of ‘original’ and ‘unwanted’ signal requires sufficient insight into the characteristics of both these entities. In this thesis, expression of such information in mathematical form to differentiate the underlying latent image from the corruption, either convolutive or additive, is termed as image model.

In mathematics language an image can be seen as a function $x$ defined on a subset $\Omega$ from $\mathbb{R}$. For each data point $(u, v) \in \Omega$, $x(u,v)$ is a number (grayscale) or a vector (colour), given as

$$
\text{Grayscale} \rightarrow x : \Omega \rightarrow \mathbb{R}
$$

$$
\text{Colour} \rightarrow x : \Omega \rightarrow \mathbb{R}^3
$$

The observed image, $y$, usually represents a degraded / corrupt version of the original scene $x$. The degradation process mostly consists of a linear corruption operator, $h$, followed by some additive noise $n$, such that

$$
y(u,v) = \int h(u,v,s) x(s) ds + n(u,v)
$$

The linear corruption is represented by the convolution of $h$ and $x$, where $h$ is known as point spread function (PSF). This convolution produces the blurring effect and $n$ represent the system or measurement noise. In image restoration community the blurring operator $h$ is mostly assumed spatial-invariant, which simplifies the observation model

$$
y(u,v) = \int h(s) x(u-s, v-r) ds + n(u,v)
$$

$$
= h * x + n
$$
Stepping into the discrete world of digital devices and computers, the digital image is treated as matrix, where the subset \( \Omega \) is replaced by a two-dimensional (2D) array of size \( U \times V \). Each data point in this 2D matrix is called pixel, which is the basic unit of image. In discrete form the observed image can written as,

\[
y(u, v) = \sum \sum \{ h(s, r) x(u - s, v - r) \} + n(u, v)
\]

(1.1)

In the matrix-vector form (1.1) can be written as

\[
Y = HX + N
\]

(1.2)

where \( H \in \mathbb{R}^{UV \times UV} \) is a block circulant with circulant block (BCCB) matrix created from \( h \); and \( Y, X \) and \( N \) are vectors formed by lexicographical ordering of \( y, x \) and \( n \), respectively. The 2-D discrete Fourier transform (DFT) can be used to optimally diagonalized the BCCB matrix, which allow eq. (1.2) to be expressed in discrete frequency domain.

In section 1 of this thesis image deconvolution based methods are presented where the image degradation includes both the blurring operator and additive noise. Where as in the section 2 the operator \( H \) either acts as a pixel masking (inpainting) or identity (denosing) operator, the details are given in respective sections. As a linear inverse problem, the image restoration tasks, such as de-noising, deblurring and inpainting, required a good image prior in order to differentiate between the true image contents and the degradation. In this thesis we used different image prior such as heavy-tailed Gaussian distribution for natural images and non-negativity in section 1 and sparse prior in section 2. Each proposed method is evaluated against state-of-the-art methods in respective chapters. The images used for the simulations in this thesis are shown in figure 1.1.

### 1.3 Organization

In methodological point of view, this thesis is divided into two parts. In section 1 the image deconvolution based approaches are presented to address image deblurring and denoising problem. This part contains four chapters–Chapter 2 to 5. The basics concepts of image deconvolution and a brief literature review are presented

---

1. In a circulant matrix a basic row of entries is repeated with a periodic shift in the positions of its entries. Now, in a block circulant with circulant block (BCCB) matrix a basic row of blocks is repeated with a periodic shift in the position and each block is a circulant matrix.

2. The vector is formed by stacking either the columns or rows of image in lexicographical ordered; i.e., \( \mathbb{R}^{U \times V} \rightarrow \mathbb{R}^{UV} \)
In chapter 2, from chapter 3 to 5, three novel image deblurring algorithms and their comparison with other state-of-the-art methods are presented.

In section 2 of this thesis, we focused on the sparseland model for image restoration. The main application area is image denoising and image inpainting. This part is also consist of four chapters—Chapter 6 to 9. In chapter 6 the basics of sparseland are introduced and a brief review of the related literature is presented. Three new sparsity-based dictionary learning algorithms, with applications in image restoration, are outlined in chapter 7 to chapter 9. The thesis is concluded in chapter 10. The overall thesis organization is depicted in figure 1.2.
Introduction

Figure 1.2: Thesis layout
Part 1: Image Deconvolution

This first part of thesis presents restoration techniques for images contaminated mainly by blurring operator and some additive noise, shown in figure 2.1.

Blur and noise are the two common degradation sources in image applications. The blurring kernel, also called point spread function (PSF) is convolved with the original image and produces a low pass (broadening) kind of effect, mostly caused during image formation process. Generally, the relative motion of camera and object, out of focus optical system, or atmospheric turbulence cause the blurring effects. The removal of blurring effects, in the presence of noise is an ill-conditioned linear inverse problem, where the solution is highly sensitive to the noise. Mathematically, the observed blurred and noisy image, can be model as

\[ y = h \ast x + n \]  \hspace{1cm} (2.1)

where \( x \) is the latent original image, \( h \) represents the non-negative blurring PSF whose size is small compared to the \( x \) size and \( \ast \) is the convolution operator. The system or observation noise is represented by \( n \) and \( y \) is the observed image. The blurring kernel \( h \) is independent of the noise \( n \), which represents system or observation distortions. In matrix-vector form Eq. (2.1) can be represented as,

\[ Y = HX + N \]

where \( H \) is a BCCB matrix and \( Y, X, N \) are vectors formed by lexicographically ordering of \( y, x, \) and \( n \); respectively. Image deconvolution is a strong mathematical tool that can be utilized to estimate a plausible solution for this ill-posed deblurring problem. One main reason of deconvolution success in image deblurring is the independence of blurring effects from other types of image distortions. Deconvolution exploits this independence and counters the effects of blurring operator using some prior information about the image, imaging system or PSF in the form of constraints on acceptable estimate.
This part consists of a total of three chapters — Chapter 3 to 5. For the reader’s convenience, the basic concepts and a brief general review of the related literature are presented in this chapter. The method specific literature and the related concepts are given in each chapter respectively.

### 2.1 What is blur?

Blur is one of the main image degradation. The main sources of blurring in an image are camera defocus, relative motion of camera and object, and the random atmospheric turbulence. In image processing literature blurring operator is mostly attributed to the convolution of a PSF with the image data. The PSF represent the non-random spreading (blurring) of a light point caused by the optical system. In frequency domain, PSF is the inverse Fourier transform of optical transform function (OTF), which describes the response of a linear and space invariant system to an impulse signal.

The theoretical modelling of blur is of fundamental importance in image deconvolu-
§2.2 Image deconvolution

Deconvolution is referred to the process of resolving or decomposing a set of overlapping peaks into their separate additive components [Cornwell and Braun 1989, Press et al. 1992], the overlapping is mostly caused by convolution operator. Image deconvolution is a well-studied approach used to restore images attenuated by the unwanted convolution of blurring PSF. In mathematical terms, deconvolution is an ill-posed linear inverse problem [Hansen 1997]; i.e., a small variation in the input generates a large modification in the output, or even completely different output. As the output of an ill-posed problem does not continuously depends on the input(s), so the direct inversion of blur can cause uncontrolled amplification of the additive noise [Bertero and Boccacci 1998]. In addition to ill-posedness, image deconvolution is also singular, i.e., the solution may not be unique; many different estimates of image and blur can result in the same optimal solution [Kundur and Hatzinakos 1996]. However, most image-blur pairs may corresponds to estimate with ringing or other distortion, which can be eliminated in search of a plausible solution. Depending on the choice of the prior, the analytical solution may be difficult to obtained, so most results are based on approximation [Wang et al. 2012].

Broadly speaking image deconvolution techniques fall into two main categories,
Part 1: Image Deconvolution

blind and non-blind, depending on the availability of information about the degradation process. In case the information or an estimate of blurring PSF is available, then the deblurring is known as non-blind deconvolution. On other hand, if not any or very little information about the PSF or original image is provided, then the restoration is termed as blind deconvolution, which is a comparatively challenging problem to solve.

2.2.1  Non-Blind deconvolution

In many imaging applications the statistical information of the degradation such as blur or noise, is available and can be utilized accurately in the deconvolution process, for example in astronomy and microscopy it is easy to model the blurring PSF. Most of the methods in this category follow the maximum likelihood (ML) approach, where the solution is sought in terms of available blur information.

2.2.2  Blind image deconvolution

In many applications the observed degraded image is the only available data without any or very limited information about original image or blurring PSF. In such situations the BID based methods are the appropriate techniques to restore the original image, where the missing parameters are estimated from the information within the observed data set [Levin et al. 2011b]. BID is a fast developing area mainly due to its importance and applications [Levin et al. 2011b]. Although the general concept of blind deconvolution was outlined in [Oppenheim et al. 1968] during late 1960’s, the word was used the first time in [Stockham et al. 1975] during mid 1970’s. Blind deconvolution hit the imaging community directly for first time in [Cannon 1976; Ayers and Dainty 1988]. The principal concept of BID stands by the observation that blurred images always contains the information about the blurring PSF, i.e., the light spreading of fine points or line structure which makes the PSF [Weiss and Freeman 2007]. In most BID techniques the PSF is handled as a function with unknown parameters that are computed from the image data [Wang et al. 2012]. In last two decades this line of research has been extensively explored with many outstanding works and some very important results reported, but it still remains an open research problem. Even with constant attention from the research community, there are still limitations in terms of robustness, stability, convergence and uniqueness [Levin et al. 2011b].

Most of the BID methods stem from concepts in estimation theory, numerical analysis, and linear algebra [Campisi and Egiazarian 2007]. Depending on the order of
parameter estimations, BID methods can be divided into two classes, i) priori PSF estimation and ii) simultaneous PSF and image estimation. In the former case the PSF is estimated a priori and then used with any image restoration methods to estimate the image, like in Ghiglia et al. [1993] Chalmond [1991]; where in the later case, both the image and PSF are estimated concurrently, in the same iteration. Methods in the first category are comparatively simple but limited to the parametric blur Levin et al. [2011b]. Most of the BID methods belong to the later class, which are more complicated. For a detailed review readers are referred to Kundur and Hatzinakos [1996] Campisi and Egiazarian [2007] Levin et al. [2011b].

### 2.3 Literature Review

Image deblurring has a rich literature with methods from a single image to multiple images and from a single source to multiple channel imaging. Broadly, the image deblurring methods presented so far fall into two main classes; stochastic and deterministic Wang et al. [2012]. In stochastic methods, images are modelled as a sample function of random fields and estimate the latent original image as the most probable realization Campisi and Egiazarian [2007]. Whereas, the deterministic methods estimate the image parameters by norm minimization of some objective function Chan and Shen [2005]. As an ill-posed problem, a credible estimate in deconvolution problems requires some strong assumptions in terms of constraints on latent image, imaging system and blur. These constraints are important entities and have strong reflections on the final solution. In literature the constraint based solution, is termed as regularization techniques Tikhonov and Arsenin [1977], which span a vast set of mathematical and heuristic concepts You and Kaveh [1999]. In Bayesian framework these constraints are formulated in the form of prior information and the parameters estimation, is framed as minimization problem Bishop [2006]. The main objective term in this case, consist of a data fitting term and a prior or regularization function. The most commonly used regularization includes prior density distribution of original image and blurring PSF Roth and Black [2005]. In probabilistic framework the inclusion of prior distribution corresponds to the maximum a posteriori (MAP) technique where the unknown parameters are estimated by maximizing the posterior distribution Kay [1993] Bishop [2006]. The MAP based methods have been successfully applied in image deconvolution problems Katsaggelos and Lay [1991], given as

\[
(\hat{x}, \hat{h}, \hat{\Omega})_{MAP} = \arg \max_{x,h,\Omega} P(y|x,h,\Omega)P(x|\Omega)P(h|\Omega)P(\Omega)
\]
where $\Omega$ is the set of unknown parameters.

Blurred image restoration has been addressed with different techniques in literature. The recent techniques includes methods using image prior \cite{Xu_2013, Levin_2006, Danielyan_2012, Zhang_2011}, multiple images \cite{Chow_2001, Zhang_2014, Harmeling_2010}, and hardware based methods \cite{Grossberg_2004, Levin_2007}. In \cite{Roth_2005} a review of different prior models on the original image is presented. Although the majority of the image deblurring algorithms are designed for shift-invariant blurs, in practice blurring can also be shift-variant \cite{Ayan_2010, Sorel_2008, Kubota_2005}. In practical applications the PSF is mostly unknown or is very ill modelled, therefore the blurred image restoration naturally favours blind deconvolution and the literature is mostly dominated by BID techniques. The ill-conditioned nature of image deconvolution also plays a part in this, as a small perturbation in PSF severely affect the estimation, although there exits some nonblind techniques \cite{Chantas_2008, Katkovnik_2005, Chantas_2006} which overcome this difficulty to some extent \cite{Almeida_2009}.

In the last two decades, BID has been explored extensively by the image and signal processing community, with some very profound solutions, such as \cite{Neelamani_2004, Danielyan_2012, Levin_2011a, Bioucas-Dias_2006, Fergus_2006, Shan_2008, Likas_2004}, and more. A range of recent algorithms have stressed the utilization of image statistics to characterize the underlying latent image \cite{Wang_2012}. These priors, aid to counter the ill-conditioned nature of BID, but selection of proper prior is not an easy task \cite{Sang_2010, Favaro_2014}.

Several blur specific techniques are also suggested to handle the BID problem, such as for motion blur \cite{Levin_2006} and defocus \cite{Simonov_2009}. This class of methods work for the PSF with well-known parametric form that can be characterized by its frequency domain zeros \cite{Chicoski_2007}. The length of blurring PSF depends on the space between zeros. In \cite{Cai_2012, Cai_2009} a straight line segment with uniform intensity, is used to model the one-dimensional motion blur. Similarly in \cite{Yin_2006, Oliveira_2007} a circle with uniform intensity is used for defocus blur model. Unfortunately, these model based techniques are not suitable in BID as the real blurs are more complex than the simple models, for example, motion blur due to sensor or object movement is far more complex than the simple straight line motion \cite{Almeida_2009}. 
In Seghouane [2011] a ML estimation based approach, with multivariate distribution prior for image and PSF, is given for blind image deblurring. A motion deblurring algorithm with heavy-tailed prior on image distribution and L1 norm for sparse blur is outlined in Shan et al. [2008], with special attention on ringing artifacts. The consistency of gradient profile across a different range of images, also attracts attention in many reported work. The use of such sparse gradient favours sharp images Wang et al. [2012], but required additional constraints Levin et al. [2011b]. In Fergus et al. [2006] natural image statistics in conjunction with ensemble learning is used to estimate the motion blur, using image gradient as prior. The same image statistics is also used in Levin [2006] where the image is classify in blurred and non-blurred regions and the blur direction is estimated from the minimal derivative variation. A mixture of bilateral filtering, shock filter, and gradient thresholding is given in Cho and Lee [2009] for blind motion deblurring. In Krishnan et al. [2011] a different $\ell_1/\ell_2$ regularization, where $\ell_2$ norm for image gradient and $\ell_1$ for weighting, is used for BID.

Another line of work focused on the variational based approaches to handle more general blurs and non quadratics image priors Likas and Katsaggelos [2004, Molina et al. [2006]. A variational Bayesian approach with a suboptimal Gaussian blur model is presented in Molina et al. [2006, Babacan et al. [2009]. Total variation (TV) regularization with edge preservation is employed in Favaro [2014, Chan and Wong [1998]. A hierarchical TV based parameters estimation is suggested in Babacan et al. [2009]. A learning base approach which starts with main edges and gradually moves to fine details is presented in Shan et al. [2008]. However, these methods are limited to small blur sizes and failed in large or sever blurs. Recently, TV with split Bregman iterations Bregman [1967, Goldstein and Osher [2009] for various blurs is presented in Li et al. [2012b]. The accuracy of these TV methods depends on the complexity of optimization problems and the ability in penalizing the piece-wise smooth image information.

In some applications often more than one image of the same observation is available this additional image set can facilitate the parameters estimation. In Kubota and Aizawa [2005, Sroubek and Flusser [2005] regularized algorithms are presented for multi-image blind deconvolution. In Rav-Acha and Peleg [1992] two observed images are used to estimate the motion blur and the latent image. Similarly, in Yuan et al. [2007] a combination of noisy and blur images are utilized to get an estimate of original clean image.
In contrast to the normal practice of simultaneous MAP estimation of image and blur, Levin [Levin et al., 2011a] proposed the MAP estimation of only the blur kernel for BID. In [Levin et al., 2011b] it is argued that in addition to prior selection, the estimator also play a vital role in the MAP based BID solution. The number of unknowns in PSF estimation is comparatively much less than image due to the size difference, therefore a credible MAP estimate of PSF, marginalizing over image, is more likely to be obtained. Following this path and to avoid the trivial MAP estimation, many novel techniques are suggested in recent years. In [Krishnan et al., 2011] Almeida and Almeida [2009] methods are presented with strong regularizations in the initial stages and gradually relaxing in iterations which shows improved results. Some other additions are explicitly sharp edge detection [Jia, 2007] Whyte et al. [2011] Xu et al. [2013], normalized sparsity constraint on images [SparseBID] and sparse representation with wavelets [Cai et al., 2012].
EM based Hybrid Fourier-Wavelet Image Deblurring

In the literature images are mostly modelled as multidimensional random processes, mainly due to the inherent randomness in the imaging process or scene under observation. Therefore, image restoration can be seen as a problem of statistical inference, where the goal is to estimate certain unknown parameters of probability density function (PDF) from the observed image or data distribution. In the last few decades, maximum likelihood estimate (MLE) gained more attention in the image processing community due to its autonomous behaviour, compared to other traditional methods. Specifically in blurred image restoration, MLE has been successfully used to estimate the unknown blur and noise parameters. In MLE the data distribution PDF is evaluated at an observed data sample, conditioned on some parameters, and seeks the parameter’s values that can best explain the observed data distribution. Intuitively MLE is an appealing choice in many data analysis problems, however for data with missing, hidden or incomplete observations, the direct evaluation of MLE can be difficult; if not possible. The difficulties mainly arise due to the lack of closed form solution.

In this chapter we framed blurred image restoration (deconvolution) as missing or incomplete data case and used one of the classical statistical tool, the expectation maximization (EM) algorithm, to address this missing data problem. A hybrid Fourier-Wavelet based deconvolution approach is used, where the sparse property of wavelet coefficients is modelled using the class of Gaussian Scale Mixture (GSM). The GSM represents the heavy-tailed statistical distribution, suitable for natural images. The underlying original image and noise parameters are estimated by alternating EM iterations based on available and hidden data sets, where regularization is introduced using an intermediate variable. The resulting optimization problem is computationally efficient and converge in few iterations.
This chapter is organized as: Section 3.1 introduce the general EM algorithm framework followed by a short description about the GSM class distribution in section 3.2. In section 3.3 a brief overview of some related literature are presented. The proposed image deblurring algorithm is presented in section 3.4. The experimental evaluation of the proposed technique and its comparisons with some state-of-the-art methods are presented in the section 3.5 followed by the concluding remarks in the last section 3.6.

3.1 Expectation Maximization Algorithm

Data analysis with missing or incomplete observation is a longstanding statistical research problem, which transform a relatively simple analysis into a complex problem. Here we are considering the case of missing or incomplete data, due to imperfection with or limitations to the observation process. The increase in complexity is due to the fact that most of the classic and modern statistical techniques are designed for complete data. One direct and simple approach is to drop incomplete observations from the statistical analysis, but unfortunately situations arise where missing data cannot be ignored for one reason or another. In statistical data analysis, the term incomplete implies the existence of two distributions $A$ and $B$ and a many-to-one mapping from $A$ to $B$ McLachlan and Krishnan [2008]. The observed data $b$ is a realization from $B$ and the corresponding $a$ in $A$ is not directly observed but indirectly through $b$. The $a$ is mostly referred as complete data distribution and $b$ as observed or incomplete data distribution. Given $b$ there are many possible $a$ that will satisfy $b$, therefore an engineered approach is required to pick the right $a$ Bishop [2006]. Incomplete data analysis has been a major area of statistical research over the past few decades. A good portion of research is devoted to handling the missing data in observation Orchard and Woodbury [1972] Dempster et al. [1977] Wu and Tang [2010] Amari [1995] and the reference within.

One of the successful statistical tool to deal and infer the missing information in observed data is the expectation maximization (EM) algorithm. Although the name EM was first tossed up in Dempster et al. [1977], the underlying theoretical foundation of EM approach, was first outlined in Orchard and Woodbury [1972]. It is a general iterative algorithm for parameter estimation through ML when some of the random variables involved, are consider missing or incomplete. The core theory of EM algorithm is to efficiently compute the MLE when the observed data can be considered incomplete, especially when optimizing the likelihood is analytically in-
tractable. The EM incorporate some auxiliary function and variable, such that it has similar behaviour to MLE, but comparatively easier to maximize. EM works with the relationship between the unknown data model parameters and the missing data. The general EM principle is, as noted in [2008 McLachlan and Krishnan, 2002 Howell]: with an estimate of missing values the calculation of model parameters would be easy, similarly with the updated model parameters it would be possible to obtain unbiased estimate for the missing data. This suggests an alternative iterative process where the model parameters, e.g. variance, covariance, mean etc., are estimated in the first step and then, using the updated parameters in a regression equation, the missing data is estimated. The filled-in data set is then used in the first step for parameters re-calculation, and so on. In reality EM do not find the actual values for the missing data part, but in fact estimate what the model parameters (e.g. variance, covariance, mean etc.) would be, without going through that step [2010 Enders]. On convergence EM estimate sufficient model parameters for any kind of data analysis in the subsequent steps. Below a short description of EM, is provided for those who may be rusty on this or new to the topic.

The EM algorithm involves two basic steps; in the first step the expectation of observed data is calculated conditioned over the current parameters values and in the second step the parameters values are re-estimated by maximizing the conditional expectation, calculated in the first step, hence the algorithm is named EM. Let \( Y \) be a random vector with joint density \( P(Y; \theta) \), where \( \theta \) is some set of parameters \( \theta \in \Theta \). In case of missing data, only a function of complete data \( Y \) is observed, denoted as \( (Y_{\text{obs}}, Y_{\text{miss}}) \), representing the observed and missing part; respectively. This can be expressed by

\[
P(Y; \theta) = P(Y_{\text{obs}}, Y_{\text{miss}}; \theta) = P_1(Y_{\text{obs}}; \theta)P_2(Y_{\text{mis}}|Y_{\text{obs}}; \theta)
\]  

(3.1)

where \( P_1 \) and \( P_2 \) represent the joint density of \( Y_{\text{obs}} \) and \( Y_{\text{miss}} \) (given \( Y_{\text{obs}} \)), respectively. The log likelihood of equation (3.1) can be written as

\[
L(\theta; Y_{\text{obs}}) = \log(P(Y; \theta)) = \log(P_1(Y_{\text{obs}}; \theta)P_2(Y_{\text{mis}}|Y_{\text{obs}}; \theta))
\]

\[
= \log P_1(Y_{\text{obs}}; \theta) + \log P_2(Y_{\text{mis}}|Y_{\text{obs}}; \theta)
\]

rearranging,

\[
L(\theta; Y_{\text{obs}}) = L(\theta, Y) - \log(P_2(Y_{\text{mis}}|Y_{\text{obs}}; \theta))
\]

\[
l_{\text{obs}}(\theta, Y_{\text{obs}}) = l(\theta, Y) - \log P_2(Y_{\text{mis}}|Y_{\text{obs}}; \theta)
\]
EM suits more in the situations when maximizing the observed data log likelihood ($l_{\text{obs}}$) is more complicated than maximizing the complete data log likelihood ($l(\theta; Y)$). Since in missing or hidden data case $Y$ is not fully observed, so the direct maximization of $l(\theta; Y)$ is not possible. In EM algorithm, the maximization of $l(\theta; Y)$ is achieved through the maximization of its conditional expectation, given the observed data. Starting from some arbitrary initial value of parameters, $\theta^0$, the conditional expectation is evaluated iteratively with the current estimate of parameters,

$$Q(\theta; \theta^0) = E[l(\theta; Y)|Y_{\text{obs}}]$$

$Q(\theta; \theta^i)$ is then maximized with respect to $\theta$ to estimate $\theta^{(i+1)}$ for the next iteration, such that

$$Q(\theta^{(i+1)}; \theta^i) \geq Q(\theta^{(i-1)}; \theta^i) \text{ for all } \theta \in \Theta.$$ 

The first step is estimation/expectation (E) and the second step is maximization (M). These steps are repeated alternately until convergence or some stopping criteria, such as $|l(\theta^{(i+1)}; Y) - l(\theta^{(i)}; Y)| < \epsilon$, where $\epsilon$ is a small positive integer. The EM algorithm is an unbiased and efficient estimator, it is simple and easy to implement and numerically more stable. These key properties make EM a better choice for the complex maximum likelihood computations. A possible side effect is the slow rate of convergence, the dependency on initial condition, and model specific nature.

### 3.1.1 EM in image processing

The adaptability and usefulness of the EM algorithm has been reported in a diverse range of image restoration applications. The first appearance of EM in image processing literature can be trace back to 1982 when it was first used for PET images in Shepp and Vardi [1982], and for blurred image restoration EM was first introduced in Katsaggelos and Lay [1991]. Since then the application of EM in the image processing community flourished with impressive results. Now the key word EM and image alone in IEEE electronic database returned over 1941 results in the period from 2000 to 2014 with 518 journals and around 1413 conference publications.

EM algorithm has been successfully employed for MAP based blur and noise parameters estimation in image restoration. A pioneer work in this direction, where the degradation (blur) identification and image restoration is performed in one step, is proposed in Lagendijk et al. [1988] Lagendijk et al. [1990] Katsaggelos and Lay [1991]. This line of work is then extended in Biemond and Lagendijk [1989] by parametric

---

1Based on the online search on http://www.ieee.org/index.html, dated 21 May, 2015.
modelling of blur and image model and for the partially known blur in [Mesarovic et al. 1996]. A little different approach, where EM based image deblurring using Gibbs priors is proposed, with correlated Gaussian noise is presented in [Hebert and Lu 1995] and with Poisson noise in [Hebert and Leahy 1991]. Recently EM is employed for noise parameter estimation in mixed probability distributions, for example Poisson-Gaussian [Jezierska et al. 2014]. In addition to blurred image restoration EM is also used in other image applications, for example photometric stereo [Wu and Tang 2010], hyperspectral and multi band [Zhang 2014] image restoration, Pulse-Echo ultrasound image restoration [Ng et al. 2007], content based image retrieval [Yuan and Zhang 2010], and Gaussian mixture model based image segmentation [Nikou et al. 2010].

3.2 Gaussian Scale Mixture

Wavelets have been used intensively in data analysis and statistical applications. Wavelet descriptors are considered the best representative of a large class of real world images [Mallat 1998]. In wavelet domain, generally natural images have non-Gaussian regularities and contain inter and intra-subband dependencies. These wavelet coefficients have highly kurtotic marginal distribution and exhibit strong correlation with the neighbouring coefficients. In literature different Markov random fields (MRF) models [Romberg et al. 2001] have been proposed to capture the non-Gaussian marginal behaviour based on the local descriptors, among them the field of exports models [Roth and Black 2005] which shows impressive performance for image denosing.

Recently, scale mixture models [Wainwright and Simoncelli 2000] have also been used to capture the non-Gaussian local structure of wavelet coefficients. These models proved to be a good choice for data with high mutual correlation and higher order dependencies. Specifically, Gaussian scale mixtures (GSM) is successfully employed with good description of local image statistics [Bioucas-Dias 2006], which leads to efficient parameter estimation and data inference models in wavelet domain. The GSM is constructed as normal variance mixture model with the covariance matrix involves a stochastic scaling factor with a given prior distribution. The GSM is constructed as normal variance mixture model with the covariance matrix involves a stochastic scaling factor with a given prior distribution. In GSM model the densities are symmetric and zero mean, having leptokurtoic marginal densities, which represents heavier tails than Gaussian [Wainwright and Simoncelli 2000]. The marginal and joint statistics of GSM distributions [Andrews and Mallows 1974] can account for both the marginal and pairwise joint empirical distributions of wavelet coefficients. In GSM, the wavelet coefficients within a window correspond to linear
combination of a zero-mean Gaussian random vector and a hidden multiplier, known as scale factor.

Let the probability density function (PDF), \( P_Y(Y) \), of some random variable \( Y \) is symmetric about zero and the derivative satisfy

\[ (d/dy)^k p_Y(y) \geq 0; \text{ for } y > 0 \]

Then there exist independent variables \( S \) and \( G \), with \( G \) being a standard normal variable or vector, such that

\[ Y = \sqrt{S}G \]  

(3.2)

where the multiplier \( S \) is an independent positive scalar random variable and \( G \) is a zero-mean Gaussian random vector with covariance matrix \( C_G \). The GSM model characterizes the set of real-valued random vectors that can be expressed as eq.(3.2). For \( Y \) to have the unique GSM representation, the mean of \( S \) must be specified. Without loss of generality, we can assume that \( S \) has unit mean \( \text{P}[S = 1] \). This implies that the covariance matrix \( C_Y = E[S]C_G = C_G \). Accordingly, the density of \( S \), \( P_S(S) \), is determined through the integral.

\[ P(Y) = \int_S N_Y(0,SC_G)P_S(S)ds \propto \int_S \frac{1}{\sqrt{|C_G|}} \exp(-\frac{Y^T C_G^{-1} Y}{2S}) P_S(S)ds \]

Many well known data distributions, such as the Student’s-t, the-stable, the generalized Gaussian and symmetrized Gamma distributions can be considered as special case of GSM model [Andrews and Mallows [1974] Rakvongthai et al. [2010].

Recently, some extensions of GSM based models in wavelet domain are suggested in the literature, including oriented-adaptive GSM [Hammond and Simoncelli [2008], complex GSM [Rakvongthai et al. [2010], space variant GSM [Guerrero-Colon et al. [2008], and GSM for de-rotated complex coefficients [Miller and Kingsbury [2008].

3.3 Related Work

Blurred image restoration is a longstanding problem in the image processing community, with application in remote sensing, medical, astronomy, photography, and more. The main goal of deblurring is to reconstruct a plausible estimate of the original scene from its degraded observed version. This restoration process is critical to many image processing applications [Kundur and Hatzinakos [1996]. Image deconvolution proved to be the best image restoration technique [Cichoski and Amari]
3.3 Related Work

In addition to noise removal, it also accounts for the undesired point response function which models the imaging system. Deconvolution is a well-studied, classical research task, but using wavelets it comes up with some problems, the main hurdle lies in the wavelet domain representation of convolution operator Figueiredo and Nowak [2003].

Image deconvolution is an ill-posed linear inverse problem and direct inversion leads to a dramatic amplification of observed noise, or some times is impossible. The key to cope with this kind of problem is the proper incorporation of prior knowledge about the original image into the restoration process Campisi and Egiazarian [2006]. In recent literature many possible solutions have been suggested, mostly under regularization or Bayesian frameworks Seghouane [2011] Lagendijk et al. [1990] Shan et al. [2008] Mariana and Luis [2010] Danielyan et al. [2012]. The heart of such techniques is the a-priori knowledge expressed by the prior/regularization term. However, modelling a proper prior for real world images is quite a challenging task Roth and Black [2005]. Relevant classes of priors include, Gaussian, sparse, and heavy-tailed priors. With Gaussian priors the resulting optimization problem is convex and the optimal solution can be derived in closed form Kay [1993]. In case of image gradient, sparse priors are preferred over Gaussian because sparse priors opts to concentrate gradient at a small number of pixels, leaving most of the image pixels unchanged Roth and Black [2005]. The advantage of using sparse priors has been reported in Bioucas-Dias [2006] Figueiredo and Nowak [2001], the main benefit is that it produces sharp edges and efficiently deals with unwanted ringing artifacts.

Recent research in natural image statistics revealed that natural images and their wavelet coefficients exhibit strongly heavy-tailed distribution. The usefulness of heavy-tailed priors for natural images are already exercised in image denoising Strela et al. [2000] Portilla et al. [2003], motion deblurring Shan et al. [2008] Fergus et al. [2006], blind deconvolution Bioucas-Dias [2006], image restoration Figueiredo and Nowak [2001] and video matting Apostoloff and Fitzgibbon [2005]. Some example of this priors class includes the generalized Gaussian Moulin and Liu [1999], the Jeffreys noninformative prior Yang and Berger [1997], and the Gaussian mixture (GM) Wan and Nowak [1999].

The GSM has been considered as a unifying framework for heavy-tailed distributions. In GSM model, the densities are symmetric and zero mean, having leptokurtoic marginal densities, which represents heavier tails than Gaussian Wainwright and Simoncelli [2000]. The marginal and joint statistics of GSM distributions Andrews and
Mallows [1974] can account for both the marginal and pairwise joint empirical distributions of wavelet coefficients. In GSM, the wavelet coefficients, within a window corresponds to linear combination of a zero-mean Gaussian random vector and a hidden multiplier, known as scale factor.

### 3.4 Proposed Algorithm

Blurred image deconvolution aims at recovering the original image $x$ from its degraded observed version $y$. The class of observation herein considered, is described by the standard linear observation plus Gaussian noise model

$$y = Hx + n$$

where $H$ is a square block-Toeplitz matrix accounting for space invariant blur and $n$ is a sample of zero-mean white Gaussian noise of variance $\sigma_n^2$, that is, $n \sim \mathcal{N}(0, \sigma_n^2 I)$, where $I$ denotes an identity matrix. In this proposed method the original image $x$ and noise $n$ are assumed unknown.

We are specially interested in the case where $H$ models the space-invariant periodic convolution in image domain. Therefore $H$ can be diagonalized by the two-dimensional (2-D) discrete Fourier transform (DFT),

$$H = F^T D F.$$

where $F$ represents the 2D DFT, $(.)^T$ denotes conjugate transpose, and $D$ is the diagonal matrix containing the DFT coefficients of the convolution operator represented by $H$. This representation facilitate the convolution by $H$ in image domain by a simple point wise multiplication in DFT domain, i.e.,

$$Hx = F^T D F x$$

$$= F^T D X$$

where $X$ denotes the DFT of $x$.

Mostly $H$ is ill posed or even non-invertible and direct inversion results in severe noise amplification. Therefore, some regularized procedure is required to get an acceptable estimate of the original image. A widely used choice is MAP estimation, defined as,

$$\hat{x} = \arg \max_x \{ \log p(y \mid x) - \log p(x) \},$$

(3.4)
If the prior \( p(x) \) is considered zero mean Gaussian then the MAP estimation can be written as [Bishop 2006],

\[
\hat{x} = \arg \max_y \left\{ -\frac{1}{2\sigma_n^2} \| Hx - y \|^2 - x^T C_x x \right\},
\]

(3.5)

\[
= C_x H^T (\sigma_n^2 I + HC_x H^T)^{-1} (y - H).
\]

where \( C_x \) is the covariance matrix of \( p(x) \).

### 3.4.1 Wavelet Based Deconvolution

As discussed, wavelet based approaches are considered best in many real world image processing methods. Switching the image deconvolution problem from DFT domain to wavelet domain will clearly be an advantage. In wavelet based estimation, the original image \( x \) is represented in terms of orthogonal wavelet expansion, provides sparse representation [Mallat 1998]. Let assume a given wavelet forward and reverse transforms be represented by matrices \( W \) and \( P \), respectively, and \( \omega = Wx \) the wavelet coefficients of \( x \). We assumed that the system has perfect reconstruction property; i.e., \( PW = I \) and \( P = W^T \).

The MAP estimation (3.4), can be re-expressed in wavelet domain as,

\[
\hat{\omega} = \arg \max_{\omega} \left\{ \log p(y | \omega) - \log p(\omega) \right\},
\]

(3.6)

\[
= \arg \max_{\omega} \left\{ -\frac{1}{2\sigma_n^2} \| HP\omega - y \|^2 - \log p(\omega) \right\},
\]

In case of blur, i.e., \( H \neq I \), the solution of (3.6) leads to hard computational problem, mainly due to matrix \( HP \). This is because of the facts that, unlike \( H \), \( HP \) is not block-circulant, thus can not be diagonalized by DFT. To avoid this problem a possible solution, presented in [Figueiredo and Nowak 2003], is to decompose the noise, \( n \), into two different Gaussian noises, i.e,

\[
n = \alpha Hn_1 + n_2
\]

where \( \alpha \) is a positive parameter, and \( n_1 \) and \( n_2 \) are independent noises, such that,

\[
p(n_1) \sim N(0, I)
\]

(3.7)

\[
p(n_2) \sim N(0, \sigma_n^2 I - \alpha^2 HH^T)
\]
This decomposition of noise, in addition to computational ease, also decouples denoising step from deconvolution.

### 3.4.2 Hybrid Fourier Deconvolution

We formulate our image deconvolution approach in wavelet domain, where GSM model is used to represent the wavelet distribution. The direct manipulation of blurring operator $H$ and wavelet matrix is avoided by decomposing noise into two different noises. The image model (3.3) can be rewritten as,

\[
\begin{align*}
    y &= Hz + n_2 \\
    z &= P\omega + an_1
\end{align*}
\]  

(3.8)

Further we modeled $\omega$ as GSM, decomposing its density into a linear combination of zero-mean Gaussian densities, i.e.,

\[
\omega = \sqrt{S}G.
\]  

(3.9)

where $S$ and $G$ are random variables with densities $p_S(S)$ and $\mathcal{N}(0, I)$, respectively. With this set of equations we can perform image denoising and deconvolution in two separate steps. From (3.8) clearly if we had $z$, we would be in pure denoising problem. Therefore, treating $z$ as an intermediate hidden variable, a cascaded EM algorithm can be used to get an estimate of $\omega$.

Let $z$ and $(y, z)$, plays the role of missing and complete data, respectively, in our 1st EM iteration. The EM algorithm yields an estimate, $\hat{z}$, by two iterative steps, until convergence criterion.

- **E-Step:** In this step conditional expectation of complete data is computed, using observed data and current estimate $\hat{z}_t$,

  \[
  Q(z, \hat{z}_t) = E[\log p(y, z | \omega) | y, \hat{z}_t].
  \]

- **M-Step:** Update the estimate as,

  \[
  \hat{z}_{t+1} = \arg \max_{\omega} \{Q(z, \hat{z}_t)\}.\]
3.4.3 E-Step.

The conditional expectation of intermediate complete data condition on \( \hat{z} \) can be written as,

\[
Q(z, \hat{z}_t) = E[\log p(y \mid z) + \log p(z \mid \omega) \mid \hat{z}_t]
= -\frac{\|y - Hz\|_2^2}{2\zeta} - \frac{\|z - P\omega\|_2^2}{2\alpha^2}
\]

(3.10)

where \( \zeta = \sigma^2 I - \alpha^2 HH^T \).

3.4.4 M-Step.

The M-Step consists of maximizing (3.10) with respect of \( z \), to get the next iteration parameters i.e.,

\[
\hat{z}_{(t+1)} = \arg\max_\hat{z} Q(z, \hat{z}_t)
\]

An estimation of mean value, ignoring terms that do not depend on \( z \), can be calculated as Bishop [2006],

\[
\hat{\omega}_{t+1} = P\hat{\omega}_t + \frac{\alpha^2}{\zeta^2} F^T D^T (Fy - DFP\hat{\omega}_t).
\]

(3.11)

The \( \hat{z} \) represents the intermediate deblurred image estimate. Now to deal with the noise another iteration of EM algorithm is run on the estimated deblurred image. Let \( u \) and \( (z,u) \) represent missing and complete data distributions, then in EM terms the next iterations consists of the following two steps.

3.4.5 E2-Step.

Having the intermediate deblurred estimation \( \hat{z} \), the next step is to handle the noisy artifacts.

\[
Q_2(\omega, \hat{\omega}_t) = E[\log p(Wz \mid \omega) + \log p(\omega \mid u) + \log p(S) \mid \hat{\omega}_t]
= -\frac{\|Wz - \omega\|_2^2}{2\alpha^2} - \frac{1}{2} \omega^T U_t \omega.
\]

(3.12)

where \( S_t = \text{diag}\{E[s_i^{-1} \mid \hat{\omega}_t]\} \), as \( s_i \) are assumed zero mean and uncorrelated, so it has values only on the diagonal. As \( p(\theta \mid S) \sim \mathcal{N}(0, S) \), so
\begin{align*}
E[S^{-1} | \omega] &= \int_0^\infty S^{-1} p(S | \omega) ds \\
&= \int_0^\infty S^{-1} \frac{p(\omega | z)p(S)}{p(\omega)} ds \\
&= \frac{1}{p(\omega)} \int_0^\infty S^{-1} p(\omega | S)p(S)ds 
\end{align*}
(3.13)

from definition,

\begin{align*}
p(\omega | S) &= \frac{1}{2\pi} e^{-\frac{1}{2} \omega^2 S} \\
\frac{dp(\omega)}{d\omega} &= -\frac{\omega}{2} p(\omega | S) \\
\frac{dp(\omega)}{dw \omega} &= -S^{-1} p(\omega | S)
\end{align*}

Substituting in (8) and exchanging the integral with the derivative, we obtain.

\begin{align*}
E[S^{-1} | \omega] &= -\frac{1}{\theta p(\omega)} \frac{dp(\omega)}{d\omega} 
\end{align*}
(3.14)

Eq. (3.14) is very important, since it allows us to compute \(E[S^{-1} | \omega]\) directly from \(p_\omega\) without the explicit knowledge of the density \(p_S\) \cite{Bioucas-Dias2006}. Moreover, if the prior, \(p_\omega(\omega) \propto e^{-\phi(\omega)}\), is induced by a thresholding/shrinking function (T) such that \(\omega = T(z)\) and \(T^{-1}\) exist for \(|\omega| > 0\), then

\begin{align*}
E[S^{-1} | \omega] &= -\frac{1}{\omega p(\omega)} \frac{dp(\omega)}{d\omega} \\
&= \frac{\phi'(\omega)}{\omega} \\
&= \frac{T^{-1}(\omega) - \omega}{\omega^2}
\end{align*}
(3.15)

Eq. (3.15) allow us to compute \(E[S^{-1} | \omega]\) directly from a given thresholding/shrinking function without the knowledge of the density \(p_\omega\) or the density \(p_S\) \cite{Bioucas-Dias2006}. 

### Simulation Results

Table 3.1: Experimental Setup.

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Blur</th>
<th>σ²_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gaussian, std=1.6</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>9x9 Uniform</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>9 Pixel Motion blur</td>
<td>0.45</td>
</tr>
<tr>
<td>4</td>
<td>1/(1 + x₁² + x₂²), x₁, x₂ = −7, .., 7</td>
<td>2</td>
</tr>
</tbody>
</table>

3.4.6 M2-Step.

The M2-Step consists in maximizing (3.12) with respect of ω, i.e.,

\[
\hat{\omega}_{t+1} = \arg \max_{\omega} Q_2(\omega, \hat{\omega}_t),
\]

\[
= \arg \max_{\omega} \left[ -\frac{\|W_{z_t} - \omega\|_2^2}{2\sigma^2} - \frac{1}{2} \omega^T S_t \omega \right]
\]

\[
= \arg \min_{\omega} \left[ \omega^T (\alpha^2 S_t + I) \omega - 2 \omega^T W_{z_t} \right]
\]

\[
= \left( \alpha^2 S_t + I \right)^{-1} W_{z_t}
\]

This estimate of original image is then passed to the next iteration for the intermediate deblurred image restoration, until a stopping criterion. The main iteration steps involve eq.(3.12) and eq.(3.16).

3.5 Simulation Results.

In this section, we present a set of experimental results illustrating the performance of our technique and comparisons with other methods of the same kind. Five different image degradation scenarios are examined, where in each case the original image is degraded by convolving a blur function, h, followed by additive random noise with variance σ²_n, summarized in Table-3.1. The comparisons are done using simulated blurred and noisy versions of some well known grayscale test images. In all the experiments, we employed Haar (Danbechies-2) wavelets; other wavelets always lead to very similar results, and the blur operator is applied via the Fast Fourier Transform (FFT). For the GSM multiplier prior density, p(S), we choose non-informative prior or objective prior, as this kind of prior does not require the fitting of any parameters to the noisy observation and is efficient to implement [Portilla et al. 2003]. We compared our algorithm with some state of the art image restoration methods: including, KL-BIR [Seghouane 2011], WaveGSM [Bioucas-Dias 2006], HQD [Shan et al. 2008], and BM3D [Danielyan et al. 2012]. KL-BIR is basically a blind image restoration method, but here we pass the mean of simulated blurring PSF as input to make it non-blind.
EM based Hybrid Fourier-Wavelet Image Deblurring

In the experiments we are interested in noise suppression and blur removal in our final restored image. The Improvement-in-Signal-to-Noise-Ratio (ISNR) is used as an objective measure of the quality for the restored images, defined as

$$\text{ISNR} = 10 \log_{10} \frac{\| x - y \|^2}{\| x - \hat{x} \|^2},$$

(3.17)

where $x$, $y$, and $\hat{x}$ are respectively the original, observed and estimated images.

The initial value of $\sigma_n^2$ is sampled from a normal distribution and ISNR is computed for different values of Blur Signal to Noise Ratio (BSNR), defined as

$$\text{BSNR} = 10 \log_{10} \frac{\| Hx \|^2}{\| n \|^2},$$

(3.18)

For each value of BSNR we generated 20 random noises and plot the average of ISNR values. Similar to Portilla et al. [2003], the GSM multipliers, $S$, are obtained by sampling with logarithmically uniform spacing, which required fewer samples than linear sampling. In case of logarithmic representation, $\log(S)$, the Jeffrey’s improper prior corresponds to a constant. In our tests we used the interval $[\log(S_{\text{min}}), \log(S_{\text{max}})]$, to generate 17 samples of $\log(S)$, with a step size of 2. The values of $S_{\text{min}}$ and $S_{\text{max}}$ were chosen in consideration that left and right tails of posterior are properly covered by integration interval. We used the interval of [-23, 9], however, we noticed very slight performance difference by shifting the interval values in range of ±20. We observed that for the same BSNR value our method always produce better restoration results and is computationally faster. As a guide, on a Window 7 workstation with 2.66 GHz Intel Core2 Quad CPU, our method takes only 1.074 seconds in our MATLAB implementation for $256 \times 256$ image.

For WaveGSM, similar to Bioucas-Dias [2006], Garrote prior is used as GSM density, which leads to the wavelet domain thresholding rule,

$$T(x) = \left( \frac{x^2 - \lambda^2}{x} \right)_{+},$$

(3.19)

where $(x)_{+}$ represent “the positive part of”, i.e., $(x)_{+} = x$, if $x > 0$, and $(x)_{+} = 0$, if $x \leq 0$.

This threshold value is then used to calculate the mean value of GSM density. For BM3D [3] and HQD [4] we used the available online codes with the default parameters. The obtained ISNR values are compared in table 3.2. Clearly our proposed method outperform the other methods, this was expected because, in contrast, we addressed the noise removal as a separate step, in addition to deblurring. The GSM distribution assumption for wavelet coefficient of latent image efficiently handles the ringing

---

3 http://www.cs.tut.fi/foi/GCF-BM3D
4 http://www.cse.cuhk.edu.hk/leojia/projects/motion_deblurring/
### Simulation Results

Table 3.2: ISNR (in dB) Comparisons with other state of the art methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Exp.</th>
<th>Exp.</th>
<th>Exp.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cameraman.</td>
<td>Barbara</td>
<td></td>
</tr>
<tr>
<td>BSNR</td>
<td>29</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>WaveGSM</td>
<td>4.52</td>
<td>7.012</td>
<td>5.31</td>
</tr>
<tr>
<td>KL-BIR</td>
<td>4.91</td>
<td>10.93</td>
<td>6.03</td>
</tr>
<tr>
<td>HQD</td>
<td>0.668</td>
<td>-2.891</td>
<td>3.07</td>
</tr>
<tr>
<td>BM3D</td>
<td>4.31</td>
<td>10.45</td>
<td>-</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>5.22</td>
<td>11.20</td>
<td>6.83</td>
</tr>
</tbody>
</table>

Figure 3.1: Exp-2: Visual comparison.
artifacts, which is a problem in many image deconvolution methods. WaveGSM addressed this problem by using heavy-tailed priors, but the M step of this method involves solution of linear system to avoid the direct manipulation of blurring PSF and wavelet transformation matrix \((HW^\top)\), which introduced blocky artifacts. HQD method copes with both blocky and ringing artifacts but the local prior assumption unchanged the nearly smooth regions of blurred image, which fails in the restoration of images with very fine edges, also HQD is mainly focused on motion deblurring and fails on other kind of blurs. BM3D produced promising results but required more iterations. The KL-BIR is mainly image restoration, which addressed the deblurring of an image in Fourier domain, however this kind of estimation tends to suffer from unpleasant ringing artifacts near the edges and image boundaries. FMD is mainly motion deblurring algorithm, which fails severely to deal with other kinds of blurs. The visual comparisons of results for Exp-2 is presented in Fig.1. Finally, in figure 2, we plot the evolution of ISNR along different values of BSNR for Exp.3 with lena image as input. A detailed performance of our algorithm, in terms of ISNR, for the online available degraded image data set \(^5\) is presented in Table 3.3.

\(^5\)www.wisdom.weizmann.ac.il/~levina/papers/LevinEtalCVPR09Data.zip
3.6 Conclusion

We addressed the blurred image restoration problem in two steps. The first step mainly consists of deblurring, which is efficiently done in Fourier domain. In the second step noisy artifacts are removed in the estimated deblurred image. This step is carried out in wavelet domain, where the wavelet coefficients are modelled as Gaussian scale mixtures (GSM) to account for inter and intra scale dependencies. This hybrid Fourier-wavelet regularization technique results in an optimized and efficient image restoration method. The GSM modelling of neighbouring wavelet coefficients reduce the ringing artifacts and prevent smoother outputs, which are the limitations in many image deblurring algorithms.
Finding an appropriate representation of the data or observations is a challenging task in data analysis applications. A proper representation reveals any hidden structure or relation(s) in the data and helps in reducing the data dimensionality to facilitate the subsequent processing steps. In many image and signal processing tasks, the data is often non-negative and sometimes has smooth or sparse representation. To process such data it is desirable to exploit these additional constraints in order to avoid the absurd or unstable results. In this chapter we address the blurred image restoration problem, using the non-negative and sparsity constraints on the image data and propose a parameter free algorithm to efficiently handle these natural constraints.

Blurring is a common source of image degradation in many image analysis applications. Blind image deconvolution (BID) is an apposite approach for blur removal in real images, where a very little or no information about the degradation (blur and noise) is available. Being an ill-posed linear inverse problem, a regularized and well constrained approach is required, for a credible solution of BID model. Recently sparse representation base modeling emerged as an efficacious tool in the image processing community, with applications as regularizers in inverse problems. In this chapter the sparsity constraint is fused with the non-negative matrix approximation, to address the BID problem. An iterative frame work is developed to estimate the non-negative sparse approximation of the sharp image and blurring kernel. With sparsity constraint, an estimate of the sharp image is obtained without solving the ill-posed deconvolution model. Although similar formulation has been proposed, but unlike other BID methods, the proposed approach is parameter free and benefits from the explicit blurring kernel estimation. The experimental results validate comparatively better performance of proposed methods against the other methods.
This chapter is organised as: First a brief introduction to low rank matrix approximation is presented in section 4.1. A short explanation of non-negative matrix factorization (NMF) and sparse NMF is given in 4.2. A summary of related work in the literature is presented in 4.3. The proposed novel approach is given in 4.4, followed by the experimental results and comparisons with other methods in 4.5. Finally the concluding remarks are given in 4.6.

4.1 Low rank approximation

The recent evolution in sensor and computing technology results in enormous amounts of raw processing data, in almost every branch of science, where extracting the meaningful information from the measurements, observations and understanding the underlying structure, have become an important challenge. To efficiently handle this huge amount of data, the designing of appropriate dimensionality reduction techniques, is an important and challenging task in multivariate data analysis Lathauwer et al. [2000] Fodor [2002]. With the reduction of data dimensionality, such low rank approximation or matrix factorization should also identify the principle components, hidden structure or prominent features in the data Field [1994]. In most situations, the observed data is sorted as matrices or tensors, which can be decomposed into the underlying components, in order to extract the hidden information Cichocki et al. [2009] Gillis and Plemmons [2011]. Matrix factorization has become an important tool in signal processing and linear algebra, with applications in machine learning, image analysis, information retrieval and data mining. It provides an effective and simple approach to find a suitable representation of the data matrix, which typically makes the underlying latent structure in data, explicit to facilitate the subsequent data processing steps. This matrix factorization promotes the embedding of high dimensional data in lower dimensional space and reduces the effects of noise and other artifacts. Generally the matrix factorization is carried out using the popular factorization techniques such as, singular value decomposition (SVD), principal component analysis (PCA), and QR, LU, Cholesky and Eigen decomposition Li et al. [2001]. Each method uses different assumptions, regarding the factor matrices and the underlying structure, the choice of appropriate method is application dependent and a critical choice. However, with the advent of new methods based on random projections and convex optimization, a surge of diverse matrix factorization algorithms, with constraints on rank and/or positivity and/or sparsity have emerged in last two decades.

Generally, low rank approximation of a matrix X consist of estimating its factors U
and \( V \) such that

\[
X = UV
\]

Where \( X \in \mathbb{R}^{m \times n}, U \in \mathbb{R}^{m \times k} \) and \( V \in \mathbb{R}^{k \times n} \), with \( k \) represent the rank of factorization. The factorization can be exact \( (X = UV) \) or approximate \( (X \approx UV) \). The main objective is to summarize and split the information contained in \( X \) into \( k \) factors; the columns of \( U \). The characteristics such as entries, shape and rank of the factors \( U \) and \( V \) depends on the application context. In most applications \( U \) is called basis matrix and \( V \) denoted the coefficients matrix. Generally the matrix factorization algorithms, such as SVD and exponential PCA, minimize a Bregman divergence \cite{dhillon2006} between the non-negative input and its approximation. The Bregman divergence between two matrices \( A \) and \( B \) for a convex function \( F : S \subseteq \mathbb{R} \rightarrow \mathbb{R} \) with continuous first derivates can be written as,

\[
D_F(A, B) = F(A) - F(B) - \nabla F(B)(A - B)
\]

In literature different priors and regularization are imposed for a variety of matrix factorization methods to address both the over-fitting and need for information pooling across rows and columns of component matrices \cite{cichocki2008,chen1998}. Most of these methods, such as SVD, QR, Eigen decomposition, PCA, ICA, etc., differ in terms of the statistical properties attributable to the different constraints imposed on the representative matrices and their underlying structure, however there is no constraint on the sign of entries in the factorized matrices \cite{wang2013}.

### 4.2 Non-negative Matrix Factorization (NMF)

In many data analysis applications often the data to be analysed, is non-negative and their low rank factorization is constrained to be comprised of strictly positive values to match the physical data description and facilitate the subsequent steps by avoiding the absurd or unpredictable results. The basic connotation of non-negativity is: purely additive combination of non-negative components \cite{paatero1994,wang2013}. In many real world observation data, such as images and spectra, the negative values are physically meaningless. The classical matrix factorization methods, such as PCA, SVD, ICA and others, do not guarantee this non-negativity constraint, paved the path for methods with non-negative low rank factors to approximate a non-negative data matrix. In contrast, Non-negative matrix factorization (NMF) is relatively a novel parameter free approach for low rank approximation, which incorporates the non-negativity constraint and represents the
non-negative data, as combination of low rank non-negative basis and activations vectors [Buciu 2008]. NMF was first tossed up in mid-1990’s with the name positive matrix factorization by Paatero and Tapper [Paatero and Tapper 1994] for multivariate data analysis, that resolved the limitations of PCA. Later Lee and Seung [Lee and Seung 1999, Lee and Seung 2001] polished the idea with the argument that the non-negativity is important in human perception and proposed a simple and effective algorithmic procedure based on multiplicative update to find the non-negative representation for non-negative data. The simplicity of their method and interpretability of the results, attracted the attention of researchers in almost every field.

Mathematically, given a non-negative data matrix $X \in \mathbb{R}_+^{m \times n}$, the goal in NMF is to approximate $X$ by a product of two lower rank non-negative matrices $U \in \mathbb{R}_+^{m \times k}$, and $V \in \mathbb{R}_+^{k \times n}$ with $k < \min(m, n)$, by minimizing the following function,

$$\min_{U, V \geq 0} D(X, UV) + R(U, V) = F(U, V)$$

Where $D$ is a similarity measure or loss function that quantify the difference between the data matrix $X$ and the factor matrices $UV$, and $R$ is an optional regularization term, defined to enforce extra properties on $U$ and $V$ such as sparsity or smoothness. The choice of $D$ depends on the probability distribution and structure of the data. The common choices for the loss function $D$ are Frobenius distance, defined as,

$$D : A, B \rightarrow \frac{\text{Tr}(AB^T)}{2} = \frac{1}{2} \sum_{ij} (a_{ij}b_{ij})^2$$

or the Kullback-Leibler divergence [Yang et al. 2011, Kullback and Leibler 1951],

$$D : A, B \rightarrow KL(A \| B) = \sum_{ij} (a_{ij} \log \frac{a_{ij}}{b_{ij}})a_{ij} + b_{ij})$$

The matrix $U$ is called features or basis matrix, contains as its column the basis vectors and the rows of $V$ contain the corresponding weighting components that assign weight to each basis vector. Each data vector $x_i$ is approximated by a linear combination of the basis vectors $u \in U$, weighted by the coefficient vector $v_i$. In the basic NMF algorithm [Lee and Seung 1999] the matrix $U$ and $V$ are updated using a weighted gradient descent, given as

$$U_{i+1} \leftarrow U_i \odot (XV_i^T) \odot (U_iV_iV_i^T)$$

$$V_{i+1} \leftarrow V_i \odot (U_{i+1}^TX) \odot (U_{i+1}^TU_{i+1}V_i)$$
where \( \odot \) is the element wise matrix multiplication and \( \oslash \) represents the element wise matrix division.

In most cases the basis matrix \( U \) is ill-posed or singular, therefore specialized models or algorithms are required for acceptable solution. Many image and signal processing problems can be framed as matrix factorization, however the selection of different cost functions and regularizations leads to different solutions [Tibshirani 1994]. The fundamental models of NMF utilize the least square based cost function to measure the closeness of approximation \( R \).

In literature the NMF algorithms can be divided into four basic categories: Basic (basic non-negativity constraint only) [Paatero 1997, Lee and Seung 1999], Constraint (non-negativity with additional regularization on factors \( U \) and/or \( V \)) [Hoyer 2004, Liu et al. 2012], Structure [Wild et al. 1999] and Generalized NMF [Ding et al. 2010]. The optimization can be obtained using multiplicative updates [Lee and Seung 2001], alternating least square [Benthem and Keenan 2004] or gradient based methods [Lin 2007]. Due to the enhanced interpretability under the non-negativity, NMF has emerged as an imperative tool in multivariate data analysis and been widely used in different fields such as; signal processing [Buciu 2008], computer vision and machine learning [Kotsia et al. 2007], data mining [Ding et al. 2006], blind source separation [Gillis 2014], sound recognition, spectral data analysis [Gillis et al. 2015], image processing [Cichocki et al. 2008, Sandler and Lindenbaum 2011], image classification, clustering [Pompili et al. 2014], and bioinformatics [Gao and Church 2005].

Generally the NMF algorithms are mostly based on the alternative iterative optimization, which have some common limitations, such as the solution sensitivity to the initialization and the non-uniqueness of solution [Donoho and Stodden 2004]. The efficiency of many NMF algorithms depend on the selection of initial matrices \( U \) and/or \( V \) [Boutsidis and Gallopoulos 2008]. Inadequate initialization results in slow convergence and often leads to an irrelevant or incorrect solution. To remedy these limitations, in addition to the non-negativity constraint, some extra auxiliary constraints on the factors \( U \) and/or \( V \) are imposed as regularization [Kim and Choi 2007]. These additional constraints cope with the ill-posedness of NMF and incorporate the additional prior knowledge about the problem. The commonly used constraints are smoothness [Pauca et al. 2006], sparsity [Hoyer 2002, Hoyer 2004] and discreteness etc. The incorporation of such constraints are problem dependent and reflect the need to compensate for the noise or other degradation present in the data.
4.2.1 Sparse NMF

Sparsity based signal and image processing, attracts more attention recently due to the appealing results and is widely used in machine learning, signal processing, neuroscience and statistics [Elad 2010]. In sparse representation the data is modelled as sparse linear combination of active basis elements, which may be pre-defined as wavelets [Mallat 1998] or can be learned from the data as [Aharon et al. 2006, Engan et al. 1999]. Sparsity is an important aspect in practice, that reduces the storage requirement and improves interpretation of the factor matrices. In contrast to the pre-defined basis, the sparse representation, with learned basis is well adopted to natural signals and led to state-of-the-art results in numerous signal processing applications such as audio processing [Fevotte et al. 2009], image classification [Mairal et al. 2008a], image denoising [Elad and Aharon 2006, Mairal et al. 2008b], texture synthesis [Peyre 2009], hyperspectral imaging [Yuan et al. 2015], blind source separation [Rapin et al. 2013] and audio processing [Fevotte et al. 2009].

Although the non-negative constraint naturally leads to a sort of sparseness in NMF, but this sparseness is somewhat a side effect, rather than a goal, as there is no explicit control over the degree of data sparseness [Hoyer 2002]. The inclusion of an explicit or controlled sparsity parameter in NMF, improves the uniqueness of the date decomposition and favors the part based representation, as in general the NMF decomposition is not unique. Generally sparseness can be endorsed in NMF by two different ways, defining a priori the desired sparsity level and adapt the main iteration to guarantee the desired sparsity level [Hoyer 2004] or alternately, a sparse regularization term is added to the main objective function [Li et al. 2001]. With the endorsed sparsity constraint the basic NMF model can be re-written as

$$\min_{U, V \geq 0} D(X, UV) + \alpha J_1(U) + \beta J_2(V)$$

where $J_1$ and $J_2$ are the application dependent sparsity constraint on $U$ and $V$, respectively. The sparsity constraint can be applied on $U$ or $V$ or on both, depends on the application. In the case where each basis vector contributes in each observation, the sparseness is imposed along the column of $U$. If each observation is approximated by a linear combination of few basis vectors, then sparseness is imposed on columns of $V$. Or if sparness is along rows of $V$ then each basis vector is used to approximate a limited number of observation vectors, which is basically related to data clustering. The alpha and beta are regularization parameters, used to balance the tradeoff between data fitting and goodness.
Due to the overall global representation, in some cases the NMF fails to extract the underlying true data elements \cite{Li2001, Hoyer2003}. In such cases the explicit control of sparseness improves the local representation of the data and leads to unique representation. The sparsity was first fused with NMF in \cite{Hoyer2002} to include an adjustable sparseness parameter along with the non-negativity, by employing the $\ell_1$ norm along columns of $V$ as sparsity penalty. This work was then extended in \cite{Hoyer2004} by explicitly adjusting the sparsity parameter by means of nonlinear projection in each iteration based on the relationship between $\ell_1$ and $\ell_2$ norm. These basic sparse NMF frameworks are adopted by many researchers in different applications, such as \cite{Morup2008, Peharz2012}.

## 4.3 Related Work

Blind image deconvolution (BID) is a well defined problem in the image processing community, where the goal is to estimate a sharp version of observed image with no or very little prior information about the blurring kernel and original image. In imaging the blurring artefact is mainly caused by imperfection in image formation processes, which includes atmospheric turbulence, defocusing, and the relative motion of object or imaging device \cite{Kundur1996}. Mostly the imaging process is linear and shift invariant thus deblurring problem can be described as additive linear degradation model given as

$$Y = HX + N$$  \hfill (4.1)

where $X$, $Y$, $H$, and $N$ represents sharp and blurred image, square matrix accounting for space invariant blur and a zero-mean additive white Gaussian noise caused by the imaging system, respectively.

BID is severely ill-posed inverse problem whose solution provides infinite sets of $H$ and $X$ for the fulfilling of $Y$ \cite{Levin2011}. To cope with this problem some prior constraints on $H$ or $X$ must be employed \cite{Campisi2007, Cichoski2007, Amari2007}. In literature BID is mostly addressed using deconvolution based methods, where $H$ is de-convolved with $Y$ to get a sharp version $X$. These methods can be categorized into two groups \cite{Kundur1996}; i) disjoint blur identification and image restoration; ii) estimate blur and sharp image simultaneously in one procedure. The deconvolution process is mostly ill conditioned and very sensitive to the noise that requires a regularized and constrained approach for a plausible estimate of the underlying sharp image.
Recently, sparsity based image deblurring methods have shown significant improvement [Elad et al. 2010]. These methods utilize the sparse representation of image patches as a prior to regularize the ill-posed inverse problem. Although, natural images are usually sub-Gaussian and can not be considered sparse in their native representation, but it is possible to transform an image in such a way that allows to fit sparsity [Bruckstein et al. 2009]. The underlying assumption in this direction is that natural images often contain multiple correlated versions of the same information, which can be approximated by a sparse representation of prototype signal-atoms, such that only few samples are sufficient for its reconstruction [Tropp and Wright 2010].

Ideally this reduction in the dimension of signal should not effect the subsequent processing step. The objective in this case is to find a set of prototype basis that can be used to represent the given image by some sparse combination of few basis. Formally, for a given image data $X \in \mathbb{R}^{n \times L}$ the sparse representation consist of learning a suitable basis matrix, $U \in \mathbb{R}^{n \times K}$ (with $K > n$ in overcomplete case), and $L$ sparse coefficient vectors with minimal number of non-zero coefficients, $V \in \mathbb{R}^{K \times L}$, such that $x_i \approx Uv_i$, where $x_i \in X$ and $v_i \in V$. For an effective sparsity, $U$ should be generalized to be non-orthogonal and over-complete, by either combining different orthonormal basis or learning from the data.

In recent literature many proposed BID algorithms have utilized the sparsity prior. In Zhang et al. [2011] sparsity prior is used to estimate the sharp image and blurring kernel. Similarly, a BID algorithm using a pair of sharp and blur basis with the sparse prior is presented in [Li et al. 2012a], but the requirement of sharp basis weakens its blind nature. A blind motion deblurring method using sparsity of blurring kernel and sharp image is proposed in Cai et al. [2009]. A fast blind deconvolution method using $\ell_1/\ell_2$ regularization on sharp image for sparsity is presented in [Krishnan et al. 2011].

In addition to sparsity constraint, the non-negative assumption over the original image leads to sparse non-negative matrix approximation (SNMA) [Hoyer 2002], with promising results in blind image restoration [Kopriva and Nuzillard 2006, Kopriva 2005]. Non-negative matrix approximation (NMA)\footnote{In literature NMA is better known as non-negative matrix factorization (NMF) but here we used word ‘approximation’ because usually exact factorization is not possible.} is a very efficient and parameter-free gradient descent method for decomposing multivariate data under strictly posi-
4.4 Proposed Algorithm

Consider an observed image, $Y \in \mathbb{R}^{m \times n}$, representing the blurred and degraded version of some original sharp image, $X \in \mathbb{R}^{m \times n}$. In a standard linear model this observation can be represented (as was in eq. (4.1) and repeated here for convenience),

$$Y = HX + N$$

where $H$ is a square matrix accounting for space invariant blur and $N$ is a zero-mean additive white Gaussian noise caused by the imaging system. In the proposed method the blur, $H$, and original image, $X$, are treated as unknowns.

Under the sparse and non-negative factorization assumption over the original image, a $\sqrt{p} \times \sqrt{p}$ image patch, $x_i \in \mathbb{R}^p_+$ with column stacks in lexicographical order, can be re-written as

$$x_i \approx Uv_i$$  \hspace{1cm} (4.2)

where $U \in \mathbb{R}^{p \times K}_+$ (with $K > p$ in overcomplete case) is the K-basis matrix, and $v_i \in \mathbb{R}^K_+$ represents the sparse mixing coefficients vector with minimal number of non-zero elements. For all the overlapping patches (4.2) can be written in matrix form as

$$X = UV$$

with $X \in \mathbb{R}^{p \times L}_+$ and $V \in \mathbb{R}^{K \times L}_+$, where $L$ is the total number of all possible overlapping patches. Here we consider the sparse version of non-negative matrix factorization where sparsity is constrained via the mixing coefficients $V$. In this setting, model (4.1) can be written as,

$$Y = H(UV) + N$$

where $Y \in \mathbb{R}^{p \times L}_+$. Ignoring the noise term, we have three unknowns to estimate $H$, $U$, and $V$. In the proposed method these hidden variables are estimated from the
observed image data, $Y$ by the optimization of following equation.

$$\{ \hat{H}, \hat{U}, \hat{V} \} = \arg \min_{H,U,V} \frac{1}{2} \| Y - HX \|_F^2 + \| X - UV \|_F^2 + \lambda \| V \|_p + \beta \| H \|_2$$  \hfill (4.3)

where $\lambda$ and $\beta$ are the regularization parameters and $\| \cdot \|_p$ is the $\ell_p$ norm, with $0 \leq p \leq 1$. For $p = 1$ the optimization problem (4.3) becomes convex and easy to find a close form solution. Note that (4.3) is not jointly convex in $H$, $U$, and $V$, but it can be made convex by solving with respect to one variable at a time keeping the others fixed. An iterative-alternative approach is a natural way to find a solution in this case, minimizing with respect to one while keeping the other fixed and iterate specific number of times or until some convergence condition [Tosic and Frossard 2011].

### 4.4.1 Non-negative Sparse Approximation

Under the sparse and non-negative factorization assumption over original image, keeping $H$ fixed, an estimate of basis $\hat{U}$ and sparse mixing coefficients $\hat{V}$ can be calculated using the following optimization

$$\hat{U}, \hat{V} = \arg \min_{U,V} \| X - UV \|_F^2 + \lambda \| V \|_1$$

A multiplicative update rule is presented in [Lee and Seung 2001] for alternative estimation of $U$ and $V$. In [Hoyer 2004] a sparse version of the multiplicative update rule is outlined. Following the same line, in this work we impose the sparsity on the mixing coefficients. Under the sparse assumption and keeping $U$ fixed, an estimate of $V$ can be calculated by

$$\hat{V} = V - \lambda U^T(UV - X)$$  \hfill (4.4)

or alternatively the non-negative version of orthogonal matching pursuit (OMP) [Tropp 2004], called NMP [Peharz et al. 2010] can be used to get an update of $V$.

The next step is to estimate the basis matrix using the updated sparse coefficients. Since no extra constraint is imposed on $U$, this update can be estimated using the multiplicative rule of [Lee and Seung 2001],

$$\hat{U} = [U \odot (XV^T)] \odot (UV^T + \lambda I)$$  \hfill (4.5)
Table 4.1: Proposed algorithm

<table>
<thead>
<tr>
<th>Algorithm: BID by Non-negative Sparse Approximation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong> ( Y \in \mathbb{R}_{0+}^{m \times n} ), no. of iterations ( T ), no. of basis ( K ).</td>
</tr>
<tr>
<td><strong>Initializations:</strong> ( X_0 = Y ), ( U_0 \in \mathbb{R}<em>{0+}^{p \times K} ), ( V_0 \in \mathbb{R}</em>{0+}^{K \times L} )</td>
</tr>
<tr>
<td><strong>For</strong> ( i = 1 ) to ( T )</td>
</tr>
<tr>
<td>1: <em>Non-negative Sparse Approximation:</em></td>
</tr>
<tr>
<td>( \text{Find mixing sparse coefficients } V ), using eq.(4.4)</td>
</tr>
<tr>
<td>( \text{Find basis matrix } U ), using eq.(4.5)</td>
</tr>
<tr>
<td>2: *Get an intermediate estimate of } ( \hat{X} ).</td>
</tr>
<tr>
<td>( \hat{X} = \hat{U} \hat{V} )</td>
</tr>
<tr>
<td>3: <em>Blur Kernel Estimation</em></td>
</tr>
<tr>
<td>( \text{Find } h ), using eq. (4.8).</td>
</tr>
<tr>
<td>4: <em>Enhance the estimated image.</em></td>
</tr>
<tr>
<td>( \text{Deblur image } \hat{X} \text{ with new } \hat{h}. )</td>
</tr>
<tr>
<td>5: ( X = \hat{X} )</td>
</tr>
<tr>
<td><strong>end.</strong></td>
</tr>
<tr>
<td><strong>Output:</strong> ( X ), ( h )</td>
</tr>
</tbody>
</table>

Where \( \odot \) and \( \oslash \) represents the element wise multiplication and division, respectively. Equipped with the estimated basis and sparse mixing coefficients, an updated estimate of original image \( \hat{X} \) can be obtained by

\[
\hat{X} = \hat{U} \hat{V}
\]

### 4.4.2 Blur Kernel Estimation

Using the updated version of original image \( \hat{X} \), the blur kernel \( \hat{H} \) can be estimated by optimizing (4.3) with respect to \( H \)

\[
\hat{H} = \arg \min_H \frac{1}{2} \| Y - H \hat{X} \|^2_F + \beta \| H \|^2_F \quad (4.6)
\]

Writing the convolution \( H \hat{X} \) as matrix multiplication \( I \cdot h \), where \( h \) is the blur kernel and \( I \) represents the block matrix with image patches \( x_i \) as columns; thus equation (4.6) can be rewritten as

\[
\hat{h} = \arg \min_h \frac{1}{2} \| Y - Ih \|^2_F + \beta \| h \|^2_F \quad (4.7)
\]
Table 4.2: Experimental Setup.

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Blur Setup</th>
<th>(\sigma_n^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gaussian, with (std = 1.6)</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>(9 \times 9) pixels uniform</td>
<td>0.42</td>
</tr>
<tr>
<td>3</td>
<td>9 pixels horizontal motion</td>
<td>0.615</td>
</tr>
<tr>
<td>4</td>
<td>(1/(1 + x_1^2 + x_2^2)), (x_1, x_2 = -7, \ldots, 7)</td>
<td>2</td>
</tr>
</tbody>
</table>

Equation (4.7) is a linear optimization problem and under the Tikhonov regularization an estimate can be obtained as

\[
\hat{h} = (\hat{I}^T + \beta I)^{-1} \hat{I}^T Y
\]  

The estimated sharp image \(\hat{X}\) can be further enhanced by deblurring with the calculated blur kernel \(\hat{h}\) using any image deblurring method. We used the Lucy-Richardson method [Lucy 1974b] implemented in MATLAB [MATLAB 2014] function \texttt{deconvlucy} and found suitable enhancement in \(\hat{X}\). The enhanced \(\hat{X}\) is passed to the first step of the algorithm for the next iteration. The algorithm is summarized in table 4.1.

4.5 Experimental Analysis

This section presents a set of experimental results validating the performance of the proposed method and its comparisons with other methods. We considered four different image degradation scenarios, where in each case the original image is convolved with a blur function, \(h\), followed by additive random noise with variance \(\sigma_n^2\); given in table 4.2.

The comparisons are done using simulated blurred and noisy version of some well known grayscale test images. We compared our algorithm with some state of the art BID methods: including, NMFSC [Hoyer 2004], SDP [Li et al. 2012a], HQD [Shan et al. 2008], and BMD [Cai et al. 2009]. The Improvement-in-Signal-to-Noise-Ratio (ISNR) is used as an objective measure of the quality for the restored images, defined as

\[
\text{ISNR} = 10 \log_{10} \frac{\| x - y \|^2}{\| x - \hat{x} \|^2}
\]

where \(x\), \(y\), and \(\hat{x}\) are the original, observed and estimated images, respectively.

The ISNR is computed for different values of Blur Signal to Noise Ratio (BSNR), defined as

\[
\text{BSNR} = 10 \log_{10} \frac{\| Bx \|^2}{\| n \|^2}
\]
### Table 4.3: ISNR (in dB) Comparisons with other state of the art methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Exp. Lena</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Exp. Cameraman</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSNR</td>
<td></td>
<td>29</td>
<td>40</td>
<td>15</td>
<td>30</td>
<td></td>
<td>29</td>
<td>40</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>NMFSC</td>
<td></td>
<td>2.52</td>
<td>2.91</td>
<td>3.21</td>
<td>3.82</td>
<td></td>
<td>2.45</td>
<td>3.04</td>
<td>3.15</td>
<td>3.57</td>
</tr>
<tr>
<td>BMD</td>
<td></td>
<td>3.19</td>
<td>3.03</td>
<td>3.83</td>
<td>4.12</td>
<td></td>
<td>3.67</td>
<td>3.57</td>
<td>3.72</td>
<td>4.04</td>
</tr>
<tr>
<td>HQD</td>
<td></td>
<td>0.538</td>
<td>-2.019</td>
<td>3.41</td>
<td>-1.31</td>
<td></td>
<td>1.61</td>
<td>-3.21</td>
<td>2.01</td>
<td>-1.27</td>
</tr>
<tr>
<td>SDP</td>
<td></td>
<td>3.27</td>
<td>3.41</td>
<td>4.21</td>
<td>4.79</td>
<td></td>
<td>3.43</td>
<td>3.14</td>
<td>4.56</td>
<td>4.85</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td>3.51</td>
<td>3.82</td>
<td>4.73</td>
<td>5.09</td>
<td></td>
<td>3.81</td>
<td>3.87</td>
<td>4.87</td>
<td>5.10</td>
</tr>
</tbody>
</table>

For each value of BSNR we generated 20 results and plotted the average of ISNR values. For the proposed method we set $\lambda = 0.01$, $\beta = 0.03$, $\sqrt{p} = 8$ and $K = 100$, the algorithm is iterated 10 times. We observed that for the same BSNR value our method produced comparatively better restoration results and is computationally less expensive. As a guide, on a Window 7 workstation with 2.66 GHz Intel Core2 Quad CPU, our method takes about 4.314 seconds in MATLAB implementation for a $256 \times 256$ image.

For NMFSC and HQD we used the available online codes\(^2\) \(^3\) with the default parameters. Incase of SDP the sharp basis matrix (dictionary) is trained on the original image using the K-SVD algorithm\(^4\) Aharon et al. [2006] and the other parameters are set as suggested (i.e $\lambda = 0.01, T = 5, \beta = 0.01, \tau = 2000$). Similarly for the BMD we used the mentioned values for the parameters (i.e $\lambda = 0.001, \mu = 0.02 \| x \|_{\infty}, \nu = 1$).

The obtained ISNR values for all the methods are compared in table 4.3. The proposed method comparatively produced better results. In HQD the extraction of local smooth structure in blurred image results in instable restoration of images with very fine edges and heavy blur; additionally HQD is mainly focused on motion deblurring and fails to other kind of blurs. BMD produced better results but the algorithm is computationally heavy and requires more iterations. The NMFSC do not estimate the blur kernel explicitly resulting in a low performance in high blurring conditions. The SDP method is not a true blind method, because it requires the original image for sharp basis. In addition, the assumption of equivalent sparse coefficients for sharp and blurred image is not true in every case. The visual comparisons of results for Exp-4 is presented in figure 4.1. In figure 4.2, we plot the evolution of ISNR along different values of BSNR for Exp.2 setting with Lena image as input. A detailed performance of our algorithm, in terms of ISNR, for the online available blurred image data

\(^2\)http://www.cs.helsinki.fi/u/phoyer/software.html
\(^3\)http://www.cse.cuhk.edu.hk/~leojia/projects/motion_deblurring/
Blind image deblurring (BID) is a challenging task with applications in astronomy, medical, remote sensing, photography and many more. This chapter describes a new blind image deblurring algorithm using a non-negative sparse approximation. Sparsity based modelling has emerged as a powerful tool in signal processing. A unifying non-negative sparse approximation is presented to estimate the blurring kernel and

\[^4\text{www.wisdom.weizmann.ac.il/}\~\text{levina/papers/LevinEtalCVPR09Data.zip}\]
sharp image. Unlike most of the BID methods, the proposed algorithm is parameter free and requires no prior statistics. The sparse non-negative approximation is learned from the observed image directly, where sparsity is constrained on the mixing coefficients. Our experimental results illustrate that the proposed method offers improved performance against the state of the art.
Images are the primary source of information in many signal processing applications, for example in biomedical, engineering and astronomy. In most cases the observed images, obtained through various imaging devices, represent a degraded version of the original scene, due to imperfection in imaging device, environment or in both. This imperfection prevents the images from conveying the true or real information regarding the observed scene or object. The most salient sources of degradation in digital images are blurring and noise. Blurring is a form of low-pass filter or bandwidth reduction of the image, mainly due to the image formation process.

In this chapter a new Wavelet- Fourier based hybrid blind image deblurring algorithm is described, which can be seen as the extension of the non-blind deblurring method presented in chapter-3. The algorithm is developed by defining an intermediate variable (or image) to characterize the original image in the wavelet domain. Both the original image and the additive noise are modeled by multivariate Gaussian process. The blurring process is specified by its point spread function, which is unknown. The original image and the blur are estimated by successive alternating minimization of the Kullback-Leibler divergence, between a model family of probability distribution defined using a linear image model and a desired family of probability distributions constrained to be concentrated on the observed data. This algorithm can be viewed as a description of successive expectation maximization (EM) in information geometric terms and leads to a hybrid Fourier deconvolution, wavelet regularization algorithm. The regularization is introduced in the algorithm using the intermediate variable by promoting the sparsity of the wavelet coefficients of real image using the Gaussian scale mixture (GSM) model. Extensive simulation experiments illustrating the effectiveness of the proposed algorithm in comparison to other existing blind image deconvolution algorithms are presented to validate the
proposed method.

This chapter is organized as: section 5.1 presents a brief introduction of Kullback-
liebler divergence (KLD), respectively. In section 5.2 some related BID methods are
reviewed. The proposed method is explained in section 5.3. Section 5.4 presents
the experimental evaluation of the proposed method and its comparisons to other
state-of-the-art BID methods. Finally the conclusion is presented in section 5.5.

5.1 Kullback-Leibler Divergence

Information geometry is a strong tool for statistical distribution analysis and mod-
eling. The Kullback-Leibler divergence (KLD), which is central to information ge-
ometry Cover and Thomas [1991], is used to quantify the divergence between a true
probability distribution and a target or model probability distribution Amari and
Nagaoka [2000] Kullback and Leibler [1951]. The KLD, also known as the relative
entropy, is a widely used tool in statistics and pattern recognition MacKay [2003]. In
information geometric terms KLD compute the proximity of two probability distrib-
utions. The KLD between two probability distributions, p(x) and q(x), on a random
variable, X, is the measure of divergence between them, formally defined as,

$$D_{KL}(P||Q) = \int \log \frac{Q(x)}{P(x)} Q(x) dx$$

It is an estimate to measure the closeness of a probability distribution and a model
or target distribution. KLD is closely related to likelihood theory where it is used
to measure the probability of an observed data distribution, given a particular data
generating model is true Duda et al. [2001]. The KLD is an asymmetric function
cannot be used as distance function, so the argument order is important. Also it is
always positive and is zero only if the two distributions are equal Kullback [1959].
Furthermore, it is invariant with respect to monotonic nonlinear transformations of
distribution variables.

The KLD plays a key role in many approximation and estimation problems, for ex-
ample in Bayesian framework, it is used to estimate the intractable density models
Minka [2001]. In variational methods the KLD is used to find the parametric ap-
proximation Bishop [2006]. As a divergence metric it is also useful in classification
Moreno et al. [2004], recognition Printz and Olsen [2002], Silva and Narayanan [2006],
and matching Goldberger et al. [2003] tasks. Recently, KLD also finds its roots in
neural coding where it is used to estimate conditional dependency between neurons
5.1.1 KLD and Likelihood Estimation

The proposed BID algorithm presented in this chapter, is based on maximum likelihood (ML) estimation. The original unknown image and the additive noise is modelled as multivariate Gaussian processes, with unknown covariance matrices. This ML based BID is approached as alternating minimization of KLD, between a model family of probability distributions, using the linear image degradation model and a desired family of probability distributions.

The connection between ML and KLD is established by using the classical statistic formulation Amari and Nagaoka [2000]. Let assume we have an observation with identical and independent distribution (iid),

\[ Y_i \sim P_\theta(Y_i), \quad \forall i = 1, 2, ..., n \]

Where \( \theta \in \Theta \) (\( \Theta \) is some parameter space), determining density of \( Y_i \). The likelihood function in this case is given by

\[
L(\theta|Y_1, Y_2, ..., Y_n) = \log(P(Y_1, Y_2, ..., Y_n|\theta))
\]

\[
= \log(\prod_{i=1}^{n} P_\theta(Y_i))
\]

\[
= \sum_{i=1}^{n} \log P_\theta(Y_i)
\]

The ML estimation of \( \theta_{MLE} \) is define as

\[
\theta_{MLE} = \arg \max_{\theta} \sum_{i=1}^{n} \log P_\theta(Y_i)
\]

\[
= \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} - \log P_\theta(Y_i)
\]

Using the strong law of large number this converges to the expectation Kay [1993]

\[
\frac{1}{n} \sum_{i=1}^{n} - \log P_\theta(Y_i) \to E[- \log P_\theta(Y_i)]
\]

This last equation is very useful when considering the divergence of distribution
parameters $\theta$ from its actual true value $\theta^*$, given as

$$E[\log P_{\theta^*}(Y_i) - \log P_\theta(Y_i)] = E[\log \frac{P_{\theta^*}(Y)}{P_\theta(Y)}]$$

$$= \int \log \frac{P_{\theta^*}(y)}{P_\theta(y)} P_\theta(y) dy$$

$$= D_{KL}(P_\theta, P_{\theta^*})$$

This shows ML estimate as a special case of KLD. Therefore, minimizing $D_{KL}$ with respect to $\theta$ will get us closer to the true value $\theta^*$.

### 5.2 Related literature

In applications such as medical imaging, astronomy and seismology images are subject to degradation due to artifacts arising from the imaging devices and physical limitations of the sensors. The degradation is modeled as blurring affecting the spatial resolution and noise added on top in the recorded image. Blind image deconvolution is a longstanding linear inverse problem where the goal is to recover a plausible estimation of the original image from the distorted and noisy recorded image. The model commonly used to characterize the recorded image is

$$y = Hx + n$$

where $y$, $x$ and $n \in \mathbb{R}^p$ are the lexicographically ordered noisy blurred image, the original image and the Gaussian white noise respectively. $p$ is the number of pixels in one image. The matrix $H$ represents the block-circulant matrix constructed from the point-spread function (PSF) or blur kernel $h$ Banham and Katsaggelos [1997]. It represents the linear degradation process. Classical image restoration aims at recovering an estimate of $x$ from $y$ assuming the blur $H$ completely known Banham and Katsaggelos [1997]. However, this assumption is not always satisfied in real world applications, as it is sometimes improbable to obtain the actual model of the blur. This motivates and directs us to blind image restoration where the primary goal is to recover an estimate of $x$ given little or no prior knowledge about the blur $H$. This is a difficult and ill-posed problem as the uniqueness and stability of the solution is not guaranteed Kundur and Hatzinakos [1996].

A number of approaches have been proposed in the literature to solve the blind image restoration problem. These approaches can be divided into two groups. First, approaches which separate blur identification as a disjoint estimation task from im-
age restoration, such as zero sheet separation [Chang et al. 1991], general cross validation [Reeves and Mersereau 1992], expectation-maximization (EM) using ARMA modeling [Lagendijk et al. 1990], maximum likelihood (ML) using ARMA modeling [Pavlovic and Teklap 1992], a priori blur identification [Kundur and Hatzinakos 1996] and [Fergus et al. 2006] where variational approximation with a Gaussian prior on the original image gradients and exponential prior on the blur kernel is used to estimate the blur kernel, and then a standard deconvolution algorithm [Lucy 1974a] is used for restoration. Second, approaches which combine blur identification and restoration in one procedure, such as support constraints recursive inverse filtering (NAS-RIF) [Kundur and Hatzinakos 1998], ML using conjugate gradient minimization, EM [Katsaggelos and Lay 1991], variational approximation with Gaussian prior on the blur kernel, the original image and the noise [Likas and Katsaggelos 2004] [Molina et al. 2006], variational approximation with a total variation function as the image prior and a simultaneous autoregressive (SAR) model as the blur prior [Babacan et al. 2009], variational approximation with a superposition of kernel functions for the blur kernel and student’s-t for the local image differences [Tzikas et al. 2009], maximum a posteriori (MAP) estimation with Gaussian prior on noise derivatives, exponential prior on the blur kernel, a defined prior on the original image to satisfy two constraints and iterative optimization [Shan et al. 2008], double minimization with a new image prior which includes a new edge detector [Almeida and Almeida 2009], alternative minimization of the Kullback-Leibler (KL) divergence [Seghouane 2011], a MAP estimation of blur to restore the original image in the EM framework [Levin et al. 2011a], edge prediction based blur and image estimation for motion blur [Cho and Lee 2009]. The method proposed in this chapter belongs to the second group.

In this chapter, an extension of the recently proposed blind image restoration approach in [Seghouane 2011] is derived. It is obtained by the introduction of an intermediate variable to characterize the original image in the wavelet domain [Figueiredo and Nowak 2003]. This has the effect of leading to a successive alternating minimization of the KL divergence between a model family of probability distributions defined using a linear image model and a desired family of probability distributions constrained to be concentrated on the observed data [Csizár and Tusnády 1984] [Amari 1995]. This algorithm can be viewed as a description of successive EM’s [Segal and Weinstein 1988] in information geometric terms and leads to a hybrid Fourier deconvolution, wavelet regularization algorithm. The regularization is introduced in the algorithm using the intermediate variable by promoting the sparsity of the wavelet coefficients of real image using the Gaussian scale mixture (GSM) model [Portilla et al. 2003].
The resulting approach has the advantage of exploiting the strength of the Fourier transform for image deconvolution and the wavelet transform for image regularization Neelamani et al. [2004].

5.3 Proposed Approach

We adopt the general convention of representing an image as a vector obtained by stacking the pixels in a lexicographical ordered form. In blind image deconvolution/restoration problems, the goal is to estimate an original image $x$ from an observed image $y$ assumed to be the output of a blurring system with a PSF or impulse response $h$ which is corrupted by additive zero-mean Gaussian noise with covariance matrix $C_n$. In matrix/vector form, $y$ is expressed as

$$y = Hx + n$$  \hspace{1cm} (5.1)

The matrix $H$ represents a 2-D convolution and is assumed block circulant or block Toeplitz with Toeplitz blocks Kundur and Hatzinakos [1996] Banham and Katsaggelos [1997]. Multiplying any vector by $H$ (or $H^H$)\(^1\) can thus be done using the 2-D fast Fourier transform (FFT) at a cost of $O(p \log p)$.

As in Katsaggelos and Lay [1991] Likas and Katsaggelos [2004], an estimate of $x$ can be generated by assuming that the original image denoted by the $p \times 1$ vector $x$ is sampled from a zero-mean Gaussian process with circulant covariance $C_x$ and is independent of $n$ which is also zero mean Gaussian with circulant covariance $C_n$. This estimate is obtained in the Fourier domain because convolution simplifies to scalar Fourier operations and it corresponds to the Wiener estimation based on the available estimates of $H$, $C_x$ and $C_n$ Seghouane [2011]. However, the estimate of $x$ obtained this way is generally unsatisfactory since it is either noisy or distorted Neelamani et al. [2004]. Therefore some regularization procedure is needed to improve the quality of the estimate of $x$. To introduce regularization to the problem, the model (5.1) is enhanced by using $z$ as an intermediate variable in the following model

$$\begin{cases} 
  z = x + n_1 \\
  y = Hz + n_2 
\end{cases}$$  \hspace{1cm} (5.2)

where $Hn_1 + n_2 = n$ with $n_1 \sim \mathcal{N}(0, C_1)$ and $n_2 \sim \mathcal{N}(0, C_2)$ where $C_2 = C_n - HC_1 H^\top$ Figueiredo and Nowak [2003]. Model (5.2) has split the initial model (5.1) into deconvolution and denoising problems. The wavelet transform which economically

\(^1\)where $(.)^H$ denote the Hermitian (conjugate transpose) of a matrix or a vector
Proposed Approach

represents real world images facilitates an effective solution to the problem of estimating $x$ from $z$. It is obtained by regularization which is derived by assuming a prior on the wavelet coefficients of the original image. If $W^\top$ represents the inverse orthonormal wavelet transform such that $x = W^\top \omega$. The vector $\omega$ corresponding to a neighborhood of $N$ wavelet coefficients of the original image $x$ is modeled using the GSM model Portilla et al. [2003] Bioucas-Dias [2006]. Then the wavelet coefficients can be modeled as

$$\omega = \sqrt{S}G$$

where $G \sim \mathcal{N}(0,C_G)$ and $\sqrt{S}$ is an independent positive random scalar variable that plays the role of a scale factor multiplying the Gaussian random vector $G$. The density of $\omega$, $p(\omega)$ is then a mixture of Gaussian defined by

$$p(\omega) = \int p(\omega|S)p(S)ds = \mathbb{E}_S\{p(\omega|S)\}$$

where $p(S)$ is the mixing density and $p(\omega|G)$ is $\mathcal{N}(0,SC_G)$. By taking expectations over $S$, $C_\omega = \mathbb{E}\{S\}C_G$.

Based on (5.2), the complete data negative penalized log-likelihood is given by

$$-\log p(y,z,x) = -\log p(y|z) - \log p(z|x) - \log p(x)$$

$$\propto ||y - Hz||_C^{-1} + ||z - x||_C^{-1} - \log p(x)$$

(5.3)

By studying the penalized log-likelihood function (5.3), it is clear that direct minimization and estimation of $x$ is difficult to be achieved. Considering the problem as an estimation problem in presence of missing data an approach based on successive alternate minimization of the KL divergence when an intermediate variable can be defined is adopted in this chapter for the estimation of $x$.

However, rather than working directly with the three variables vector $(y,z,x)$ as the complete data, $(z,x)$ the augmented incomplete data and $(y)$ the observed data; the fact that given $z$, $x$ is independent of $y$ is used to split the estimation problem into two lower dimensional tasks with $(y,z)$ and $y$ as the complete data and the observed data respectively in the first task and $(z,x)$ and $z$ as the complete data and observed data respectively in the second task. The alternative minimization of the KL divergence in the first task will lead to a deconvolution method similar to Seghouane [2011] whereas the alternative minimization of the KL divergence in the second task will lead to a denoising method close to Portilla et al. [2003] Bioucas-Dias [2006]. In comparison to Seghouane [2011], the proposed method includes a
denoising step following the deblurring step whereas in comparison to Bioucas-Dias [2006], the proposed method doesn’t assume the blur \( H \) to be known but estimates it in the first step of the algorithm. Combining both tasks of the proposed method leads to a hybrid deblurring-denoising algorithm that relies on estimation in both the Fourier and wavelet domains.

5.3.1 Estimation by successive minimization of KL divergence

The method of estimation in the presence of missing data by double minimization of the KL divergence described in Seghouane [2011] is extended to the case when an intermediate variable can be defined.

As shown in Csiszár and Tusnády [1984], finding the ML estimate of a parameter vector \( \theta \in \Theta \) of an approximating probability density using only information on the marginal probability density \( p(y) \) instead of the of the complete data probability density \( p(y, z) \) can be written in the form of a double projections onto two sets of probability densities \( P \) and \( Q \), where \( Q \) is the approximating parametric family of probability densities obtained when \( \theta \) ranges over the parameter space \( \Theta \)

\[
Q = \{ q(y, z; \theta) | \theta \in \Theta \}
\]

and \( P \) is the generating family of probability densities that assigns the empirical density \( p(\hat{y}) \) given the observations \( y_1, ..., y_M \) (concentrated on the observed data)

\[
P = \{ p(y, z) | p(\hat{y}) = \frac{1}{M} \sum_{i=1}^{M} \delta(y - y_i) \}
\]

to the marginal distribution \( p(y) \) of \( p(y, z) \).

We further assume here that an intermediate complete data random variable \( z \) exists such that two hierarchies of complete data specification

\[
p(y) = \int_{D_1} p(y, z) dz
\]

and

\[
p(z) = \int_{D_2} p(z, x) dx
\]

can be defined.

From Csiszár and Tusnády [1984], finding the ML estimate of \( \theta \in \Theta \) that maximizes
the observed data log-likelihood is equivalent to finding a pair

\[ p^*, \theta^* = \arg \min_{p \in P, \theta \in \Theta} KL(p(y, z) \| q(y, z; \theta)) \]

This is achieved using an iterative alternating minimization scheme involving two steps. The first step consists in holding \( \theta \) at \( \theta(i) \) fixed and minimizing with respect to \( p \)

\[ p^{(i+1)}(y, z) = \arg \min_{p \in P} KL(p(y, z) \| q(y, z; \theta(i))) \quad (5.4) \]

This minimum exists and is achieved for a unique \( p \in P \). Csiszár and Tusnády [1984]. Using

\[
KL(p(y, z) \| q(y, z; \theta(i))) \\
= \int p(z|y) \hat{p}(y) \log \left( \frac{p(z|y) \hat{p}(y)}{q(z|y; \theta(i)) q(y; \theta(i))} \right) \, dz \, dy \\
= KL(\hat{p}(y) \| q(y; \theta(i))) + \mathbb{E}_{\hat{p}(y)} \left\{ KL(p(z|y) \| q(z|y; \theta(i))) \right\} 
\]

(5.5)

where \( \hat{p}(y) \) is the empirical marginal density estimate obtained from the observations. Since \( \hat{p}(y) \) and \( q(y; \theta(i)) \) are given, the first term of (5.5) is unchanged by changing \( p \). This term is the lower bound on the KL divergence between \( p(y, z) \) and \( q(y, z; \theta(i)) \).

Therefore, the minimum of (5.5) is realized for

\[ p^{(i+1)}(z|y) = q(z|y; \theta(i)) \quad (5.6) \]

because it makes the second term of the right hand side of (5.5) vanish and the minimum of (5.5) corresponds in this case to \( KL(\hat{p}(y) \| q(y; \theta(i))) \).

Using the empirical estimate \( \hat{p}(y) \) obtained from the observed data, an estimate \( p^{(i+1)}(y, z) \) of \( p(y, z) \) can then be obtained as follows

\[ p^{(i+1)}(y, z) = q(z|y; \theta(i)) \hat{p}(y) \quad (5.7) \]

The second step consists then in holding \( p^{(i+1)}(y, z) \) fixed and minimizing with respect to \( \theta \)

\[ \theta(i+1) \in \arg \min_{\theta \in \Theta} KL(p^{(i+1)}(y, z) \| q(y, z; \theta)) \quad (5.8) \]
From (5.4) and (5.8) we have

\[ KL(p^{(i+1)}(y, z) \parallel q(y, z; \theta^{(i+1)})) \leq KL(p^{(i+1)}(y, z) \parallel q(y, z; \theta^{(i)})) \]

\[ \leq KL(p^{(i)}(y, z) \parallel q(y, z; \theta^{(i)})) \] (5.9)

Therefore, the sequence of parameters generated by the algorithm decreases the KL divergence between \( p(y, z) \) and \( q(y, z; \theta) \) and then increases the log-likelihood of the observed data. Since \( KL(p(y, z) \parallel q(y, z; \theta)) \) is bounded below by zero, it converges to a local minimum when \( n \to \infty \). The relation of (5.8) with the EM algorithm [Dempster et al. 1977] can be obtained by substituting \( p^{(i+1)}(z|y) = q(z|y; \theta^{(i)}) \) into (5.8) and finding the parameters that minimizes the divergence evaluated using the observations \( \hat{y} \)

\[ \arg\min_{\theta \in \Theta} KL(p^{(i+1)}(y, z) \parallel q(y, z; \theta)) \]

\[ = \arg\min_{\theta \in \Theta} \int q(z|y; \theta^{(i)}) \hat{p}(y) \log \left( \frac{q(z|y; \theta^{(i)}) \hat{p}(y)}{q(y, z; \theta)} \right) dy dz \]

Since the numerator does not depend on \( \beta \), this simplifies to

\[ \theta^{(i+1)} \in \arg\min_{\theta \in \Theta} \int q(z|y; \theta^{(i)}) \hat{p}(y) \log q(z, y; \theta) dy dz \]

and writing the integral as an expectation gives

\[ \theta^{(i+1)} \in \arg\max_{\theta \in \Theta} E_{q(z|y)} \left[ \log q(y, z; \theta) | \hat{y}, \theta^{(i)} \right] \]

\[ \in \arg\max_{\theta \in \Theta} \Phi(\theta|\theta^{(i)}) \] (5.10)

In a similar way the relation of (5.8) with the proximal point representation of the EM algorithm [Celeux et al. 2001] can also be obtained by by substituting \( p^{(i+1)}(z|y) = q(z|y; \theta^{(i)}) \) into (5.8).

An estimate of the intermediate complete data \( \hat{z} \) can be obtained from the estimate \( p(y, z; \theta^{(i+1)}) \) and the observed data \( y \). The conditional mean of \( z \) based on \( y \) is used since the densities are Gaussian.

Let \( p(z; x) \) represents the generating density associated to the second hierarchy and \( q(z; x; \alpha) \) the approximating parametric density. Then the ML estimate of \( \alpha \in \Xi \) that maximizes the intermediate complete data log-likelihood can be obtained by
iteratively applying the following two steps

\[ p^{(i+1)}(z, x) = \arg \min_{p \in P} KL(p(z, x) \parallel q(z, x; \alpha^{(i)})) \]  

which is realized for (similar to (5.7))

\[ p^{(i+1)}(x|z) = q(x|z; \alpha^{(i)}) \]  

and

\[ \alpha^{(i+1)} \in \arg \min_{\alpha \in \Xi} KL(p^{(i+1)}(z, x) \parallel q(z, x; \alpha)) \]

Using the empirical estimate \( \hat{p}(z) \) obtained from (5.7) an estimate of \( p^{(i+1)}(z, x) \) is obtained as follows

\[ p^{(i+1)}(z, x) = q(x|z; \alpha^{(i)}) \hat{p}(z) \]  

Equations (5.7) and (5.13) with the Gaussian assumption form the basis for the proposed estimation method.

Similar to (5.9), from (5.11) and (5.13), we have

\[ KL(p^{(i+1)}(z, x) \parallel q(z, x; \alpha^{(i+1)})) \leq KL(p^{(i+1)}(z, x) \parallel q(z, x; \alpha^{(i)})) \]

The sequence of parameters generated by this algorithm decreases the KL divergence between \( p(z, x) \) and \( q(z, x; \alpha) \) and then increases the intermediate complete data log-likelihood. Since \( KL(p(z, x) \parallel q(z, x; \alpha)) \) is bounded below by zero, it converges to a local minimum when \( i \to \infty \). Similar to (5.8), the relation of (5.13) with the EM can be obtained in a similar fashion to (5.10). The equations (5.4), (5.8), (5.11) and (5.13) forms the basis of the proposed successive projection algorithm. Denote by \( \theta_{(i)} \) the current estimate of \( \theta \). The next parameter estimate is derived as

- **Step 1**: Generate an estimate of \( p(y, z) \):
  - Compute
  - Step 1-a: \( p^{(i+1)}(y, z) \) using (5.7)
  - Step 1-b: Generate \( \theta_{i+1} \) using (5.8)
  - Step 1-c: After convergence generate an estimate \( \hat{z} \)

- **Step 2**: Generate an estimate of \( p(z, x) \):
  - Compute
  - Step 2-a: \( p^{(i+1)}(z, x) \) using (5.13)
  - Step 2-b: Generate \( \alpha^{(i+1)} \) using (5.13).
Step 2-c: After convergence generate an estimate of $x$ and return to Step 1 and use $x$ as the initial $z$. 

In the definition (5.4) and (5.4) two couples of complete variables $(y, z)$ and $(z, x)$ were used and the estimation is done in cascade. The minimization steps (5.6) and (5.8) correspond to the alternating minimization algorithm using $(y, z)$ as the complete data whereas the minimization steps (5.12) and (5.13) correspond to another alternating minimization algorithm where $(z, x)$ is the complete data. As described in Csiszár and Tusnády [1984] Amari [1995] using the relation with the EM, under the stated regularity conditions Gunawardana and Byrne [2005] both procedures converge to a stationary point of the likelihood. The alternative approach would consist in using $(y, z, x)$, $(z, x)$ and $(y)$ as the complete data, the augmented incomplete data and the observed data respectively. In this case the estimation pair $(p, \theta, \alpha)$ is obtained by iterative minimization of the KL divergence

$$
KL(p(y, z, x) \parallel q(y, z, x; \theta)) = \int p(x|z, y)p(y, z)\log \left( \frac{p(x|z, y)p(y, z)}{q(x|z, y; \alpha(\theta))q(y, z; \theta)} \right) dx dz dy
$$

where the second term of the right hand side of (5.15) is obtained using the fact that given $z, x$ is independent of $y$. The two terms of the right hand side of (5.15) are minimized using (5.6), (5.8), (5.12) and (5.13).

### 5.3.2 Blind image restoration by KL divergence minimization

Using the Gaussian assumption on the original image and the additive noise and the second equation of (5.2) closed form expressions are obtained for the estimates of the additive noise variance $\sigma_2^2$, the blur kernel $H$ and the intermediate image $z$. This later estimate is then used to derive the restored image in the wavelet domain using the first equation of (5.2).

From the second equation of (5.2), each member $q(y, z; \theta)$ where $\theta = \{H, C_z, C_2\}$ of $Q$ is a Gaussian distribution $\mathcal{N}(0, \Sigma)$, where

$$
\Sigma = \begin{bmatrix}
HC_z H^\top + C_2 & HC_z \\
C_z H^\top & C_z
\end{bmatrix} \in \mathbb{R}^{2p \times 2p}
$$

(5.16)

where $C_z$ is the circulant covariance of the zeros mean Gaussian process $z$. The members $p(y, z)$ of the set of generating distributions $P$ are Gaussian with marginal
distribution \( p(y) = \mathcal{N}(0, C_y) \).

The step 1-a, corresponds to the generation of \( p^{i+1}(y, z) \) given by (5.7). This is a zero mean Gaussian distribution with covariance matrix

\[
\Phi_{i+1} = \begin{bmatrix} C_y & C_y C_y^{-1} H_i C_z \\ C_z H_i C_y^{-1} C_y & \phi_{22} \end{bmatrix}
\]

where \( C_y = H_i C_z H_i^T + C_2 \), and

\[
\phi_{22} = \Lambda_{z_i} - C_y H_i C_y^{-1} H_i C_z + C_2 H_i C_y^{-1} H_i C_z.
\]

The derivation of the covariance matrix (5.17) can be found in the Appendix A. Having generated \( p^{i+1}(y, z) \), the step 1-b consists in generating \( \theta_{i+1} \) by minimizing (5.8) using the updated complete data distribution. Using (5.16) the members \( q(y, s, \theta_{(i+1)}) \) of \( Q \) are Gaussian distributed \( \mathcal{N}(0, \Sigma_{i+1}) \) with

\[
\Sigma_{i+1} = \begin{bmatrix} H_{n+1} C_{z_{i+1}} H_{y_{i+1}}^T + C_{2_{i+1}} & H_{i+1} \Lambda_{z_{i+1}} \\ C_{z_{i+1}} H_{y_{i+1}}^T & C_{z_{i+1}} \end{bmatrix}
\]

The minimum of the KL divergence (5.8) is obtained when the two covariance matrices (5.17) and (5.18) are equal, which gives

\[
C_{z_{i+1}} = C_{z_i} - C_{z_i} H_{y_i} C_{y_i}^{-1} H_{n} C_{z_n} + C_{x_i} H_{n} C_{y_i}^{-1} C_y C_{y_i}^{-1} H_{n} C_{z_n}
\]

\[
H_{n+1} = C_y C_y^{-1} H_i C_z C_{z_{i+1}}^{-1}
\]

\[
C_{2_{i+1}} = C_y - H_{i+1} C_{z_{i+1}} H_{y_{i+1}}^T
\]

\( C_{2_{i+1}} \) is forced to be positive semidefinite by performing an eigenvector decomposition and setting any possible negative eigenvalues (nonexisting or negligible in most cases) to zero.

These expressions allow us to generate iterative estimates for the parameters \( H, C_z \) and \( C_2 \) which can then be used to derive the minimum mean square error estimate of the intermediate image Seghouane [2011] for the step 1-c

\[
\tilde{z}_{i+1} = C_{z_{i+1}} H_{y_{i+1}}^T C_{y_{i+1}}^{-1} y.
\]

Using the block circulant property, \( \tilde{z}_{i+1} \) is processed in the frequency domain Katsaggelos and Lay [1991].
Having an estimate of the intermediate variable $z$, the next step is to obtain the underlying original image $x$. This step of the algorithm is processed in the wavelet domain. Let $\omega_2 = Wz$, represents $z$ in wavelet domain, then from the first equation of (5.2), the Bayes least square estimate of each wavelet coefficients of the original image $x$ centrally located in a neighborhood of $N$ wavelet coefficients of $z$ is given by

$$
\hat{\omega} = \int \omega p(\omega | \omega_2) d\omega
= \int \int \omega p(\omega, \omega_2 | S)p(S | \omega_2) ds d\omega
= \int p(S | \omega_2) E\{\omega | S, \omega_2\} ds
$$

(5.23)

Modeling $\omega$ as GSM with the assumption of additive Gaussian noise the expected value, $E\{\omega | S, \omega_2\}$, for each window is a local Wiener estimate given by

$$
E\{\omega | S, \omega_2\} = SC_GC_{\omega_2}^{-1}\omega_2
$$

(5.24)

where $C_{\omega_2} = SC_G + C_1$. To estimate the unknowns $C_1$ and $C_G$ in equation (5.24), an iterative minimization of the KLD between a generative family $P_2$ and a model family $Q_2$ is used. Each members $q(\omega_2, \omega | S; \alpha)$, where $\alpha = \{C_G, C_1\}$ and $\omega_2$ corresponds to a neighborhood of $N$ wavelet coefficients of the restored image $\hat{z}_{i+1}$, of $Q$ is a Gaussian distribution $N(0, \Omega)$ where

$$
\Omega = \begin{bmatrix}
SC_G + C_1 & SC_G \\
SC_G & SC_G
\end{bmatrix} \in \mathbb{R}^{2N \times 2N}
$$

(5.25)

The members $p(\omega_2, \omega | S)$ of the set of generating distributions $P$ are also Gaussian with marginal distribution $p(y) = N(0, C_{\omega_2})$.

The data distribution $p^{(i+1)}(\omega_2, \omega | S)$ that satisfies (5.12) and (5.13) generated in step 2 is also zero mean Gaussian distribution with covariance matrix

$$
\Phi^{(i+1)} = \begin{bmatrix}
C_{\omega_2} & C_{\omega_2}^{-1}SC_G \\
SC_GC_{\omega_2}^{-1} & C_{\omega_2}^{-1}SC_GC_{\omega_2}^{-1}SC_G
\end{bmatrix}
$$

(5.26)

where $C_{\omega_2} = SC_G + C_1$ and $\tilde{\xi}_k = SC_G - SC_GC_{\omega_2}^{-1}SC_G + SC_GC_{\omega_2}^{-1}C_{\omega_2}C_{\omega_2}^{-1}SC_G$. The derivation of (5.26) is similar to the derivation of (5.17) with $H = I$ and $C_S = SC_G$.

Having $p^{(i+1)}(\omega_2, \omega | S)$, the second partial minimization consists in generating the estimation of $C^{(i+1)}$ using the updated complete data distribution and (5.13).
Using (5.25) the members $q(\omega^2, \omega | S, \alpha^{(i+1)})$ of $Q$ are Gaussian distributed $\mathcal{N}(0, \Omega^{i+1}_{+1})$ with

$$
\Omega^{i+1}_{+1} = \begin{bmatrix}
SC_{G,i+1} + C_{i+1} & SC_{G,i+1} \\
SC_{G,i+1} & SC_{G,i+1}
\end{bmatrix}
$$

(5.27)

The minimum of the KL divergence (5.13) is obtained when the two covariance matrices (5.26) and (5.27) are equal, which gives for the step 2-b

$$
C_{z_{i+1}} = C_{z_i} - SC_{z_i}C_{\omega^2_i}C_{z_i} + SC_{G_i}C_{\omega^2_i}C_{\omega^2_i}C_{z_i}
$$

(5.28)

$$
C_{1_{i+1}} = C_{\omega^2_i} - SC_{G_i}
$$

(5.29)

Similarly to $C_{z_{i+1}}$, $C_{1_{i+1}}$ is forced to be positive semi-definite by performing an eigenvector decomposition and setting any possible negative eigenvalues (nonexisting or negligible in most cases) to zero.

Equations (5.28) and (5.29) iteratively generate estimates for $C_G$ and $\sigma^2_1$, which are then used to estimate $C_{\omega^2}$ and $E\{\omega | S, \omega^2\}$ defined by (5.24). The weighted term, $p(S | \omega^2)$, in (5.23) are calculated using Bayes rule, given as

$$
p(S | \omega^2) = \frac{p(\omega^2 | S)p(S)}{\int_0^{\infty} p(\omega^2 | \alpha)p(\alpha) d\alpha}
$$

(5.30)

Where for each window $p(\omega^2 | S)$ is given by

$$
p(\omega^2 | S) = \frac{1}{\sqrt{(2\pi)^N |C_{\omega^2}|}} \exp\left(\frac{1}{2}(\omega^2 C_{\omega^2}^{-1} \omega^2)\right)
$$

For the prior density of multiplier, $p(S)$, in (5.30) we used the non-informative prior. Specifically Jeffrey’s prior is used which leads to a better image denoising compared to other priors [Portilla et al. 2003]. In case of GSM assumption for $p(\omega^2)$ in each window, this prior is given by,

$$
p(S) \propto \frac{1}{S}
$$

It is an improper prior, but this issue can be solved by setting the prior to zero on the interval $[0, S_{\min})$ [Portilla et al. 2003], where $S_{\min}$ is a small positive constant.

Using (5.24) and (5.30) an estimate of the original image, $x$ for step 2-c can be calculated by solving (5.23) numerically, which is then used to initialize $z$ of Step 1 in the next iteration of the algorithm.
Table 5.1: Experimental Setup.

<table>
<thead>
<tr>
<th>Exp</th>
<th>Blur</th>
<th>$\sigma^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gaussian, std=1.6</td>
<td>0.05</td>
</tr>
<tr>
<td>2</td>
<td>9x9 Uniform</td>
<td>0.04</td>
</tr>
<tr>
<td>3</td>
<td>9 Pixel Motion blur</td>
<td>0.07</td>
</tr>
<tr>
<td>4</td>
<td>$1/(1 + x_1^2 + x_2^2)$, $x_1, x_2 = -7,\ldots,7$</td>
<td>0.08</td>
</tr>
<tr>
<td>5</td>
<td>$1/(1 + x_1^2 + x_2^2)$, $x_1, x_2 = -7,\ldots,7$</td>
<td>0.03</td>
</tr>
</tbody>
</table>

5.4 Experimental Results

In this section, we present a set of experimental results illustrating the performance of our proposed technique and its comparisons with other methods of the same domain. Five different image degradation scenarios are examined, where in each case the original image is degraded by convolving a blur function, $h$, followed by additive random noise with variance $\sigma^2_n$, summarized in table 5.1. The comparisons are done using simulated blurred and noisy version of some well known test images. All the images are 8-bit grayscale and of size $256 \times 256$ and $512 \times 512$ pixels. In all the experiments, we employed Haar (Danbechies-2) wavelets; other wavelets always lead to very similar results, and the blur operator is applied via the Fast Fourier Transform (FFT). In all the experiments we used two iterations of the proposed algorithm to generate the results. For the GSM multiplier prior density, $p(S)$, we choose non-informative prior or objective prior, this kind of prior does not require the fitting of any parameters to the noisy observation and is efficient to implement [Portilla et al. 2003].

We compared our algorithm with some state of the art methods: including, KL-BIR Seghouane [2011], WaveGSM Bioucas-Dias [2006], EM Figueiredo and Nowak [2003], HQD Shan et al. [2008], VA-BD Molina et al. [2006], BM3D Danielyan et al. [2012], FMD Cho and Lee [2009], TV1 and TV2 Babacan et al. [2009], BSBD Mariana and Luis [2010], EM Figueiredo and Nowak [2003] and MLO Levin et al. [2011a]. The experiments are divided in two categories, non-blind and blind decovolution. In the case of non-blind deconvolution, the size and type of the blurring PSF is assumed known. Whereas in blind deconvolution the PSF is assumed unknown, however the size of the estimated blur is limited to a square of size $21 \times 21$ pixels, which is slightly larger than the size of actual blur. No constraints are imposed on the estimated blurring PSF. In the case of non-blind deconvolution, we are interested in noise suppression and blur removal in our final restored image, where as in blind deconvolution, in addition to final image comparison, we are also comparing the estimated PSF kernel.
The Improvement-in-Signal-to-Noise-Ratio (ISNR) is used as an objective measure of the quality for the restored images, defined as

\[ \text{ISNR} = 10 \log_{10} \frac{\| x - y \|^2}{\| x - \hat{x} \|^2} \]

where \( x, y, \) and \( \hat{x} \) are respectively the original, observed and estimated images.

The initial value of \( \sigma_n^2 \) is sampled from a normal distribution and \( C_y \) is used as \( C_{x_0} \), the initial value estimate of \( C_z \) \cite{Seghouane2011}. The 2-D impulse is used as the initial estimate of the blur. The ISNR is computed for different values of Blurred Signal to Noise Ratio (BSNR), defined as

\[ \text{BSNR} = 10 \log_{10} \frac{\| Hx \|^2}{\| n \|^2} \]

Where \( H, x, \) and \( n \) are blurring operator, original image and noise, respectively. For each value of BSNR we generated 20 results and plotted the average of ISNR values.

Similar to \cite{Portilla2003}, the GSM multipliers, \( z \), are obtained by sampling with logarithmically uniform spacing, which required fewer samples than linear sampling. In the case of logarithmic representation, \( \log(S) \), the Jeffrey’s improper prior corresponds to a constant. In our tests we used the interval \( [\log(S_{\text{min}}), \log(S_{\text{max}})] \), to generate 17 samples of \( \log(S) \), with a step size of 2. The values of \( S_{\text{min}} \) and \( S_{\text{max}} \) were chosen in consideration that left and right tails of posterior are properly covered by integration interval. We used the interval of \( [-23.9] \), however, we noticed very slight performance difference by shifting the interval values in range of ±20. We observed that for the same BSNR value our method always produces better restoration results and is computationally faster. As a guide, on a Window 7 workstation with 2.66 GHz Intel Core2 Quad CPU, our method takes only 1.074 seconds in our MATLAB implementation for 256 × 256 image.

\subsection*{5.4.1 Non-Blind Deconvolution}

In this section the model and the size of the kernel are assumed known when choosing the initial blurring kernel for the proposed algorithm. This reduces the unknowns to \( \sigma_n^2 \) and \( x \). With known PSF, the estimation is generally known as image restoration in the literature. The comparisons are done with WaveGSM, EM, BSBD, HQD, and BM3D, which are image restoration techniques. For these methods the original blurring PSF is provided as input. For WaveGSM, similar to \cite{Bioucas-Dias2006}, a Garrote prior is used as GSM density, which leads to the wavelet domain threshold-
Table 5.2: Non-Blind Deconvolution ISNR Comparison.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>WaveGSM</td>
<td>4.02</td>
<td>4.95</td>
<td>5.80</td>
<td>6.00</td>
<td>5.84</td>
<td>2.50</td>
<td>5.45</td>
</tr>
<tr>
<td>EM</td>
<td>-</td>
<td>7.59</td>
<td>-</td>
<td>6.93</td>
<td>4.88</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BSBD</td>
<td>1.99</td>
<td>3.61</td>
<td>4.19</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>HQD</td>
<td>-1.18</td>
<td>-0.19</td>
<td>2.18</td>
<td>-0.31</td>
<td>0.51</td>
<td>-1.17</td>
<td>0.11</td>
</tr>
<tr>
<td>Proposed</td>
<td>5.34</td>
<td>11.66</td>
<td>15.81</td>
<td>14.43</td>
<td>15.98</td>
<td>7.15</td>
<td>13.20</td>
</tr>
</tbody>
</table>

This threshold value is then used to calculate the mean value of GSM density. For BM3D and HQD we used the available online codes with the default parameters. In the BSBD and EM case, we obtained the ISNR values from Mariana and Luis [2010] and Figueiredo and Nowak [2003], respectively, with the similar setup. The obtained ISNR values are compared in Table 5.2.

Comparatively the proposed method outperforms the other image restoration methods. Modelling the neighboring wavelet coefficients as GSM, effectively utilized the intra and inter-scale dependencies. In the proposed method, the joint distribution model with the GSM prior for the neighboring wavelet coefficients is used in the denoising step. Associated with the double alternating minimization of the KL divergence, it results in the improved image restoration results presented in Table 5.2.

The ringing artifacts, which is a problem in many image deconvolution methods are also addressed. WaveGSM addresses this problem by using heavy-tailed priors, but the M step of this method involves the solution of a linear system to avoid the direct manipulation of blurring PSF and wavelet transformation matrix $(HW^\top)$, which in addition to computational complexity, also introduces blocky artifacts. The HQD method copes with both blocky and ringing artifacts, but the local prior assumption leaves unchanged the nearly smooth regions of the blurred images and therefore fails in the restoration of images with very fine edges or dense textures. Furthermore, HQD is mainly focused on motion deblurring and fails when applied to other

$^2$where $(x)_+$ represent “the positive part of”, i.e., $(x)_+ = x$, if $x > 0$, and $(x)_+ = 0$, if $x \leq 0$.

$^3$http://www.cs.tut.fi/foi/GCF-BM3D

$^4$http://www.cse.cuhk.edu.hk/leojia/projects/motion_deblurring/
5.4 Experimental Results

![Figure 5.1](image.png)

Figure 5.1: Non-blind deconvolution visual comparisons for the Exp. 2.

kind of blurs. BM3D produces promising results, but it is mainly a non-blind image restoration method, also it is computationally more expensive than our proposed method. BSBD is more effective for images with sparse edges, but this assumption may not be true in many natural images.

The visual comparisons of results for Exp.3 setting is presented in figure 5.1, again we can say that the proposed method provides the best visual results. An evolution of ISNR, along different values of BSNR for Exp.2 setting with the Lena image as an input is presented in figure 5.2.

5.4.2 Blind Deconvolution

In this set of experiments the blurring PSF is unknown to the algorithms. In addition to image estimation, in this case we are also interested in estimating the original PSF, thus the unknown parameters are $h, \sigma^2_v, \text{and} \ x$. The blind deconvolution case corresponds to many practical situations, where it is often costly or physically infeasible to get an exact estimate of the blurring process, i.e., astronomy, medical imaging, real time detection, x-rays imaging etc. We compared the proposed method with KLD-BIR, TV-1, TV-2, FMD, BSBD, MLO and VA-BD, which are blind image deconvolution algorithms. Similar to Molina et al. [2006], the VA-BD algorithm is
executed with the assumption that both original image (x) and blurring PSF are degenerated. The best mentioned initial values of hyperparameters were used, i.e., $[\gamma_\beta = 0.4, \gamma_{\alpha_{im}} = 0, \gamma_{\alpha_{bl}} = 0.7]$, see Molina et al. [2006]. For MLO $^5$ and FMD $^6$ we used the online available codes, where for MLO, we used the best mentioned algorithm. In case of BSBD and TV1, TV2 we used the values mentioned in Mariana and Luis [2010] and Babacan et al. [2009], respectively, for similar experimental setups.

The KL-BIR is mainly an image deconvolution method, which addressed the deblurring of an image in the Fourier domain, however this kind of estimation tends to suffer from unpleasant ringing artifacts near the edges and image boundaries. VA-BD provides acceptable results both for image and PSF estimation, but finding a good estimate of the hyper-parameters is problematic and computationally expensive. Also we noticed that VA-BD results are affected with an increase in noise level. TV1 and TV2 provide good estimations, but required more iterations to converge. FMD is mainly a motion deblurring algorithm which fails to deal with other kind of blurs. MLO suggests an optimized estimation of blur kernel which is then used in a second step to recover the latent image, the results are very much acceptable. However, we noticed that MLO requires the exact size of blurring PSF for a good estimate, also the execution time is very high compared to the proposed method. On the same machine our algorithm takes 1.07, sec where as MLO takes around 80 to 230 sec for the same setup. The presented results for MLO are with the exact size of PSF as input, where our proposed method requires the exact size only in case of uniform blurs. The other issue with MLO is that it assumes the noise variance is

\[^5\text{www.wisdom.weizmann.ac.il/~levina/papers/LevinEtalCVPR2011Code.zip}\]

\[^6\text{http://cg.postech.ac.kr/research/fast_motion_deblurring/}\]
known, which is not true in the case of blind deconvolution.

The comparisons were made with the same degradation setup as table 5.1. We summarized the obtained ISNR values in table 5.3. We can observe that the proposed method provides the best estimation for the original image. Visual comparisons of estimated images for Exp.2 setting are shown in figure 5.3. The ISNR values for different BSNR levels is presented in figure 5.4, where Exp.4 setup is used with the Lena image as input. The detailed performance of the proposed algorithm compared with KL-BIR and MLO, in terms the sum of the squared errors (SSE), for the available online image Levin data are presented in table 5.4.

For an additional comparison, the proposed algorithm is compared with the method presented in Tzikas et al. [2009] (VBSK) and the results are shown in table 5.5. In the case of VBSK, due to the unavailability of the MATLAB code and the executable, the best mentioned values in section 4 of Tzikas et al. [2009] are selected for comparison. We ran our proposed method on the same experimental conditions as in Tzikas et al. [2009]. The comparison of the estimated PSF generated by the proposed method with FMD, RCS Fergus et al. [2006] and MLO, in terms of PSNR, is presented in table 5.6, for the same Levin data. The visual comparisons of the estimated PSF along with the restored image is presented in figure 5.5 for two different degradation scenarios. The PSF is displayed in the top left corner of each image. Comparatively the better estimate of PSF verifies the efficient recovery of the original image by the proposed method.

7www.wisdom.weizmann.ac.il/levina/papers/LevinEtalCVPR09Data.zip

### Table 5.3: Blind Deconvolution ISNR Comparison.

<table>
<thead>
<tr>
<th>Method</th>
<th>Exp. Lena</th>
<th>Exp. Boat</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV1</td>
<td>3.19</td>
<td>-</td>
</tr>
<tr>
<td>TV2</td>
<td>3.29</td>
<td>-</td>
</tr>
<tr>
<td>KL-BIR</td>
<td>4.28</td>
<td>5.20</td>
</tr>
<tr>
<td>FMD</td>
<td>-1.52</td>
<td>-3.78</td>
</tr>
<tr>
<td>VA-BD</td>
<td>2.45</td>
<td>4.65</td>
</tr>
<tr>
<td>BSBD</td>
<td>1.99</td>
<td>3.61</td>
</tr>
<tr>
<td>MLO</td>
<td>3.37</td>
<td>4.17</td>
</tr>
<tr>
<td>Proposed</td>
<td>5.66</td>
<td>6.19</td>
</tr>
</tbody>
</table>

Exp. 1: Lena
Exp. 2: Boat
Exp. 3: Lena
Exp. 4: Boat
Exp. 5: Lena
Figure 5.3: Blind deconvolution visual comparison for Exp. 1

Figure 5.4: Blind deconvolution ISNR for different BSNR levels for Exp. 3
Table 5.4: Comparison for Levin data in terms of SSE.

<table>
<thead>
<tr>
<th>Method</th>
<th>Filter</th>
<th>Image 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>MLO</td>
<td>44.41</td>
<td>48.88</td>
<td>35.23</td>
<td>97.10</td>
<td>26.67</td>
<td>28.66</td>
<td>46.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KL-BIR</td>
<td>38.67</td>
<td>40.48</td>
<td>24.59</td>
<td>72.94</td>
<td>17.84</td>
<td>13.98</td>
<td>34.68</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed</td>
<td>35.19</td>
<td>35.37</td>
<td>19.60</td>
<td>58.09</td>
<td>15.22</td>
<td>10.20</td>
<td>31.47</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLO</td>
<td>64.40</td>
<td>74.53</td>
<td>64.60</td>
<td>105.91</td>
<td>36.15</td>
<td>36.95</td>
<td>64.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KL-BIR</td>
<td>60.21</td>
<td>58.64</td>
<td>41.39</td>
<td>90.49</td>
<td>27.79</td>
<td>24.93</td>
<td>46.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed</td>
<td>58.77</td>
<td>55.20</td>
<td>35.10</td>
<td>85.37</td>
<td>20.11</td>
<td>18.97</td>
<td>41.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLO</td>
<td>39.05</td>
<td>47.13</td>
<td>19.18</td>
<td>68.80</td>
<td>17.59</td>
<td>21.15</td>
<td>27.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KL-BIR</td>
<td>31.37</td>
<td>37.81</td>
<td>15.61</td>
<td>54.03</td>
<td>10.25</td>
<td>17.79</td>
<td>19.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed</td>
<td>27.22</td>
<td>30.12</td>
<td>8.67</td>
<td>43.87</td>
<td>8.07</td>
<td>14.40</td>
<td>14.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MLO</td>
<td>78.38</td>
<td>129.01</td>
<td>50.08</td>
<td>98.02</td>
<td>34.07</td>
<td>62.52</td>
<td>98.19</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KL-BIR</td>
<td>60.80</td>
<td>93.73</td>
<td>38.56</td>
<td>84.91</td>
<td>26.30</td>
<td>42.20</td>
<td>72.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Proposed</td>
<td>51.78</td>
<td>81.21</td>
<td>29.19</td>
<td>77.46</td>
<td>18.05</td>
<td>34.27</td>
<td>60.72</td>
</tr>
</tbody>
</table>

Figure 5.5: Estimated image and PSF’s
### Table 5.5: Comparison with VBSK

<table>
<thead>
<tr>
<th>Methods</th>
<th>Gaussian, $\sigma^2 = 5$</th>
<th>Uniform 7x7</th>
</tr>
</thead>
<tbody>
<tr>
<td>VBSK</td>
<td>4</td>
<td>5.87</td>
</tr>
<tr>
<td>Proposed method</td>
<td>5.36</td>
<td>6.27</td>
</tr>
</tbody>
</table>

### Table 5.6: Comparison of estimated PSF in terms of PSNR.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Image 1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FMD</td>
<td>40.24</td>
<td>39.45</td>
<td>41.87</td>
<td>41.66</td>
<td>37.84</td>
<td>39.06</td>
<td>39.95</td>
</tr>
<tr>
<td></td>
<td>RCS</td>
<td>41.33</td>
<td>38.11</td>
<td>40.23</td>
<td>42.28</td>
<td>40.83</td>
<td>37.98</td>
<td>37.46</td>
</tr>
<tr>
<td></td>
<td>MLO</td>
<td>37.59</td>
<td>37.40</td>
<td>39.77</td>
<td>42.06</td>
<td>36.13</td>
<td>38.46</td>
<td>39.76</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>48.89</td>
<td>46.18</td>
<td>43.58</td>
<td>51.10</td>
<td>41.09</td>
<td>60.73</td>
<td>42.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FMD</td>
<td>37.63</td>
<td>42.69</td>
<td>39.18</td>
<td>42.27</td>
<td>39.54</td>
<td>38.39</td>
<td>39.38</td>
</tr>
<tr>
<td></td>
<td>RCS</td>
<td>41.08</td>
<td>41.05</td>
<td>39.18</td>
<td>41.73</td>
<td>37.46</td>
<td>37.83</td>
<td>37.91</td>
</tr>
<tr>
<td></td>
<td>MLO</td>
<td>36.15</td>
<td>39.66</td>
<td>35.09</td>
<td>41.63</td>
<td>36.28</td>
<td>37.54</td>
<td>38.82</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>47.61</td>
<td>46.20</td>
<td>41.31</td>
<td>53.14</td>
<td>41.08</td>
<td>57.09</td>
<td>46.36</td>
</tr>
</tbody>
</table>

### 5.5 Conclusion

A new method for BID is presented in this chapter. This method is an extension and a refinement of the method proposed in Seghouane [2011]. It is based on successive alternating minimization of the Kullback-Leibler divergence and it leads to a hybrid Fourier deconvolution similar to Seghouane [2011], wavelet regularization algorithm. In both steps of the algorithm the results correspond to the output of a Wiener filter. The resulting approach is computationally efficient and has the advantage of exploiting the strength of the Fourier transform for image deconvolution and the wavelet transform for image regularization. The effectiveness of the method in comparison to the methods of Seghouane [2011], Molina et al. [2006] and Babacan et al. [2009] and others, is illustrated using simulation on five different image degradation scenarios.
In past two decades, with the rapid developments in sensor, computing and storage technology, the volume of observed data increased exponentially in every field of science, from a trickle to torrent. Every day masses of data is generated in different forms, such as still images, radar images, voice data, CT scans, remote sensing, and more. A proper way to manage and process this deluge of data is to find an efficient representation or model that can exploit the internal structure present in the data. Generally the quest is for a low-rank representation where a small set of basic data blocks, called basis or atoms, are sufficient enough to reveal the relevant or key data information. This kind of data representation is termed as sparse representation, which enables compact storage and rapid transmission of large size data.

The selection of optimization model and the prior knowledge about the problem in hand are the two key factors in the solution feasibility of any data processing problem. In image processing applications different optimization models are suggested with specific regularization on data, such as markov random fields, principal component analysis (PCA), total variation (TV), besov spaces and more. The usefulness of any given model depends on how well it represents the data and its applicability in practice. In last two decades models with sparse and redundant representation constraint proved very effective in solving linear inverse problems, such as image denoising, deblurring, inpainting. Sparsity is an inherent characteristic of natural signals and can be obtained by decomposing the signal into its elementary basis components, selected from a predefined basis collection or trained over the data. The collection of basis is generally termed as dictionary. The search for an adequate sparse representation is a well-studied research problem, with the wavelet orthogonal bases and local time-frequency dictionaries as pioneers in this direction [Tosic and Frossard 2011]. In sparse representation the quest is to seek a compact representation of data.
Part 2: Sparse Representation based Image Restoration

Figure 6.1: Sparse representation of a data set $X$ in terms of $D$ and $A$.

interms of linear combination of basis vectors from a dictionary, using some a priori information about the problem in hand. The underlying key observation in sparse data modelling is the fact that many natural signals and images contain highly redundant information and live on low dimension manifolds or subspace that involves much fewer basis compare to the original number of samples [Tosic and Frossard 2011]. Mathematically, sparse representation of a signal $x \in \mathbb{R}^{n \times 1}$ can be expressed as $x = Da$, where $D \in \mathbb{R}^{n \times K}$ is the dictionary (with $K > n$ in overcomplete case) and $a \in \mathbb{R}^{K \times 1}$ is a coefficient vector with minimal number of non-zero elements, exhibited is Fig. 6.1.

The sparse representation journey start in 1910 with the development of wavelets, a piecewise constant functions introduced by Haar [Haar 1910]. In wavelet the signal is decomposed over a set of oscillatory waveforms, called wavelets, which are well localized and contain the key signal properties [Mallat 1998, Crouse et al. 1998]. The unitary wavelet basis were first used to examine sparse representation [Donoho 1995], leading to the class of shrinkage algorithms [Chambolle et al. 1998]. Several multi-resolution and directional transforms, such as curvelets [Candes and Donoho 2002], bandlets [Mallat and LePennec 2005], contourlets [Do and Vetterli 2005], and steerable wavelets [Freeman and Adelson 1991] are suggested with better mathematical properties to account for different image singularities and smoothness [Mallat 1998, Portilla et al. 2003]. These early developments and evolution of wavelets, from a unitary transform to overcomplete transform, bring up the redundancy factor in the data representation. These redundant and non-redundant sparse signal representations led to efficient data processing, compression and feature extraction algorithms [Donoho 1995, Bruckstein et al. 2009]. In addition to suitable transform, the signal representation in terms of these transform is also an important task, since the uniqueness of such representation is not guaranteed [Kreutz-Delgado et al. 2003b]. Different
methods are then suggested to find the sparsest possible solution based on the redundant representation, such as basis pursuits (BP) [Chen et al., 1998] and orthogonal matching pursuits (OMP) [Pati and Krishnaprasad, 1993]. Another interesting direction for learning sparse representation based on a learned dictionary from the data was presented in Aharon et al. [2006] and Kreutz-Delgado et al. [2003b]. Sparse representation is now considered as heart of recent developments in signal processing and machine learning [Dong et al., 2013] and Elad et al. [2010]. The key word "sparse representation" come up with 4,531 publications alone in the IEEE database in the period 2005 to 2015, which reflects its importance in the research community.

In recent years the research in sparse representation mainly focused on its three main aspects [Tosic and Frossard, 2011] and Kreutz-Delgado et al. [2003b]: solving the optimization problem using pursuit algorithms, such as basis pursuits [Chen et al., 1998], matching pursuits [Mallat and Zhang, 1993], orthogonal matching pursuits [Pati and Krishnaprasad, 1993]; dictionary learning, such as MOD [Engan et al., 1999], K-SVD [Aharon et al., 2006], ODL [Mairal et al., 2010]; and the application of such methods in different data processing tasks, such as image denoising [Elad and Aharon, 2006] etc.

Sparse representation based methods are rapidly developing on blend of signal processing, machine learning and statistics. Sparsity extracts the hidden structure in the data and reduces the computational and storage burden [Elad et al., 2010], Bradley and Bagnell [2009]. In past the decades extensive progress has been made in developing new methods for sparse signal processing, where the used dictionaries can be pre-defined like wavelets, bandlets, curvelets, or can be learned from the data like MOD [Engan et al., 1999], KSVD [Aharon et al., 2006]. Dictionary learning methods, which are central of sparse representation [Tosic and Frossard, 2011], have been successfully used in a number of signal and image processing applications among them image denoising [Elad and Aharon, 2006], Mairal et al. [2008b], Giryes and Elad [2014], face recognition [Wright et al., 2009], image inpainting [Fadili et al., 2009], classification [Starck et al., 2005], compression [Zepeda et al., 2011], speech denoising [Jafari and Plumbley, 2009] and more [Mairal et al., 2008a], Ravishankar and Bresler [2011], Soltani-Farani et al. [2015]. These methods seek to uncover a linear multivariate latent structure in the observed data by imposing the constraint that the estimates of the observed variables are sparse linear functions of the regressors (dictionary atoms). The other particularity of these methods is that the set of regressors or the dictionary is also tuned iteratively to find the optimal linear multivariable latent model.

1Based on the online search on http://www.ieee.org/index.html, dated 21 May, 2015.
Section 6.1 presents a brief introduction to the dictionary learning. The applications of dictionary based methods in image processing are given in section 6.2. Specifically, sparsity based image denoising and image in-painting task is explained.

6.1 Dictionary learning

In sparse data modeling the main ingredient is the basis representation matrix, referred as dictionary in sparse theory [Elad 2010]. This dictionary can be either pre-defined or can be learned from the data using some training mechanism. A good dictionary is the one which can properly sparsifies the data. The pre-defined dictionary, also referred as transforms or implicit dictionaries, are tailored specifically to images and signals which are assumed to be piece wise smooth [Mallat and LePennec 2005], examples includes steerable wavelets, bandlets, contourlets, curvelets etc. These dictionaries are computationally faster but offer limited sparsity and mostly restricted to a certain type of images and signals [Elad et al. 2010]. On other hand, to enhance the data adaptivity, the dictionary can be learned based on some training data set. The training set is either constructed from the signal itself or from similar class of signals. In such dictionaries the atoms are estimated using the empirical data, rather than some theoretical model.

The learned dictionaries are better suited to the data and leading to better performance in many data related applications, for example [Aharon et al. 2006, Mairal et al. 2012, Salzo et al. 2014]. Generally the dictionary learning algorithms mainly iterate between two stages: a sparse coding stage and a dictionary update stage. Mathematically, posed as [Olshausen and Field 1997]

$$\hat{D}, \hat{A} = \arg \min_{D,A} \| X - DA \|_F^2 \text{ s.t. } \| a_i \|_0 \leq s, \forall 1 \leq i \leq N$$

(6.1)

where \( X \in \mathbb{R}^{n \times N} \) is the data set (\( N \) is the total number of training signals/examples), \( D \in \mathbb{R}^{n \times K}, N \gg K > n \) represent the overcomplete dictionary, and \( A \in \mathbb{R}^{K \times N} \) is a sparse coefficients matrix, with \( a_i \)'s represent the column vectors. The operator \( \| \cdot \|_0 \) is the \( l_0 \) quasi-norm, which counts the number of nonzero coefficients.

In the first stage the \( D \) is kept constant and the sparsity assumption is used to produce linear approximations for each observed data vector \( x_i \), given as

$$\hat{a}_i = \arg \min_{a_i} \| a_i \|_0 \text{ s.t. } \| x_i - Da_i \|_F^2 \leq \tau, \forall 1 \leq i \leq N$$

(6.2)
§6.1 Dictionary learning

Where \( \tau \) is a small error parameter. This is an NP-hard problem \cite{Tropp2010}, where the exact solution is not possible. Different approximation methods, known as pursuit algorithms, are used to get an estimate solution for eq. (6.2). The two main approaches are greedy algorithms and relaxation methods \cite{Elad2010}. In greedy algorithms the non-zero elements of \( a \) are estimated one at a time \cite{Pati1993Mallat1993}. On other hand the relaxation methods find the solution by smoothing the \( \ell_0 \) norm and using the continuous optimization techniques \cite{Elad2010}.

The matching pursuits (MP) algorithm \cite{Mallat1993} is one of the basic greedy algorithm, which iteratively finds the atom that best defines the input signal. An improved version of MP method, called orthogonal matching pursuit (OMP) \cite{Pati1993} is mostly used in the recent years. In OMP the whole coefficient set is re-computed for each atom addition, which speeds up the convergence and insures no-repetition of atoms. In the relaxation methods, basis pursuit (BP) \cite{Chen1998} is the basic approach which replaces the \( \ell_0 \) norm estimation with \( \ell_1 \) norm approximation. Recently, an alternative approach called iterative reweight least square (IRLS) or FOCUSS \cite{Gorodnitsky1997} is also used for the solution of eq.(6.2).

In the second stage the dictionary is updated, using the learned coefficients matrix \( A \), to minimize a cost function to achieve a certain objective. Given as,

\[
\hat{D} = \arg\min_D \| x_i - Da_i \|_F^2 \quad \text{s.t.} \quad \tau \leq \| a_i \|_0 \leq s
\]  

(6.3)

Different cost functions have been used for the dictionary update to achieve different objectives. The dictionary learning methods iterate between these two stages until convergence or a specific approximation bound is reach. The performance of the dictionary methods strongly depend on the dictionary update stage since most of these methods share a similar sparse coding stage.

Besides the difference in the strategy used in the dictionary update stage, the dictionary update can also be be made sequential: where each atom is updated separately; or in parallel: the whole dictionary is updated at once. For a specific application (characterized by the cost function used to update the dictionary), the parallel approach will generally have lower computational complexity than the sequential approach. While the parallel update approach may be preferred for its advantage in computational complexity, the sequential approach generally offers better results. The two
basic and most used dictionary learning methods are MOD (Method of optimal directions) [Engan et al. 1999] and K-SVD [Aharon et al. 2006]. A basic outline of MOD and K-SVD is presented below.

### 6.1.1 MOD

In MOD the optimization problem in eq. (6.3) is addressed as nested minimization problem; with respect to coefficient matrix $A$ and dictionary $D$. Starting from an initial guess, with fix $X$ and $A$, the dictionary is estimated in parallel; given as,

$$
\hat{D} = \arg\min_D \| X - DA \|_F^2 = X A^T (A A^T)^{-1}
$$

The columns of estimated dictionary needs to be re-normalized.

### 6.1.2 K-SVD

In dictionary learning literature K-SVD [Aharon et al. 2006] enjoyed the most appreciation and is considered as the benchmark. In K-SVD a different dictionary update rule is adopted to train the dictionary in sequential fashion. Each column $d_k$ of $D$ and its corresponding coefficients row $a_k^{row}$ are updated based on rank-1 matrix approximation using the following optimization.

$$
\{d_k, a_k^{row}\} = \arg\min_{d_k, a_k^{row}} \| X - DA \|_F^2
\begin{align*}
&= \arg\min_{d_k, a_k^{row}} \left\| \left( X - \sum_{i=1, i \neq k}^K d_i a_i^{row} \right) - d_k a_k^{row} \right\|_F^2 \\
&= \arg\min_{d_k, a_k^{row}} \| E_k - d_k a_k^{row} \|_F^2 \\
&= \arg\min_{d_k, a_k^{row}} \| E_k^{R} - d_k a_k^{rowR} \|_F^2
\end{align*}
$$

where $E_k = E_k W_k$; $a_k^{rowR} = a_k^{row} W_k$, whit $w_k = \{i|1 \leq i \leq N; a_k^{row}(i) \neq 0\}$ and $I_{w_k}$ the $N \times |w_k|$ sub matrix of the $N \times N$ identity matrix obtained by retaining only those columns whose index numbers are in $w_k$. For more details readers are directed to [Aharon et al. 2006]

The singular value decomposition (SVD) of $E_k = U \Delta V^T$ is used to find the closest rank-1 matrix approximation of $E_k^{R}$ [Eckart and Young 1936]. In this case, $d_k$ is up-
dated by taking the first column of $U$ and the $a_k^{row}$ by taking the first column of $V$ multiplied by the first diagonal element of $\Delta$. This form of dictionary update retains the sparsity pattern information by updating only the non-zeros entries of $a_k^{row}$.

Aharon et al. [2006].

6.2 Dictionary learning in image processing

Recently image processing flourished with different dictionary learning based methods and its applications in restoration, enhancement, compression, inpainting, super resolution and many more. Images by nature admit a sparse decomposition over a redundant dictionary and sparse representation based methods effectively exploit this property, leading to efficient image processing algorithms. When addressing inverse problem under Bayesian framework with sparsity prior, dictionary is treated as the learning parameter set Elad [2010]. In image applications the dictionary based methods operate on patches rather than whole image, due to the data dimensionality factor Elad and Aharon [2006]. The main assumption is that every patch (e.g. $\sqrt{n} \times \sqrt{n}$) in an image has a sparse representation over a dictionary, such that it can be reconstructed as a linear combination of few atoms from the dictionary. These patches can be either collected randomly from a set of high quality images- extract patches and used them for dictionary training- or can be extracted from the observed image. The dictionary obtained using the patches from training image set is also called global or universal dictionary, which can be used for every image in the training set. In most cases the dictionary trained over the observed image performs much better than global/universal dictionary Giryes and Elad [2014].

In this thesis we used dictionary learning methods for image denoising and image inpainting tasks. In every case the dictionary is trained over the patches from the observed image. The OMP is used for the sparse coding stage, i.e. estimating the coefficient matrix $A$.

A brief outline of dictionary based image denoising and image inpainting scheme is presented in the following sections. The main observation model used in this part is

$$y = Hx + n = HDA + n$$

(6.4)

where $y$ is the observed image and $H$ is the degradation matrix. In image denoising case $H = I$ and for image inpainting $H$ represents the mask. An example of learned
dictionary and the respective image is shown in Fig. 6.2

6.2.1 Image denosing

In image denoising task the patches are extracted in full overlapping fashion from the observed image to create the training data set. A dictionary $D$ is trained over the patches set using a dictionary learning algorithm. Using the learned dictionary and sparsity constraint over the original image the main optimization equation is given as [Elad and Aharon 2006],

$$\hat{x} = \arg \min_{x,A} \frac{1}{2} \left[ \|y - x\|_F^2 + \sum_i \|R_i x - Da_i\| \right] \quad s.t. \quad \|a_i\|_0 \leq s$$ (6.5)

where $R_i$ represents the operator which extract the $i^{th}$ patch from the image and $a_i$ is the corresponding sparse coefficient vector. The first term is the proximity between the original image $x$ and its observed degraded version $y$, forced through the log-likelihood. The second term imposes the sparsity constraint, that is every patch in the estimated image has a sparse representation over some dictionary. The constraint $\|\cdot\|_0 \leq s$ assure each coefficient vector is sparse.
A coordinate decent approach is used to solve e.q (6.5), where in the first step, fixing \( x = y \), sparse vector for each image patch is estimated using

\[
\hat{a}_i = \arg \min_{a_i} \| R_i x - D a_i \|_F^2 \quad s.t. \quad \| a_i \|_0 \leq s
\]

This is basically sparse coding problem which can be solve using pursuit algorithm. In this thesis we have used OMP [Tropp 2004] for its computational efficiency. The next step is to estimate the original image using Elad and Aharon [2006]

\[
\hat{x} = \left( \gamma y + \sum_i R_i^T D \hat{a}_i \right) \left( \gamma I + \sum_i R_i^T R_i \right)^{-1}
\]

where \( \gamma \) is a positive constant and \( I \) is an identity matrix.

### 6.2.2 Image in-painting

In image in-painting the main task is to fill-in the missing pixel values that are mainly occluded by texts, subtitles, stamps, and publicity or generated by randomly removing some image pixels. Mathematically the image in-painting problem is to reconstruct the underlying complete image \( x \) from its observed incomplete version \( y \). It is assumed that the location information of the missing pixels is available. The incomplete image \( y \) and its complete version \( x \) are related through \( y = M x \), where \( M \) is the mask matrix which represent the layout of missing or occluded pixels. Thus, the optimization problem for image in-painting can be formulated as

\[
\hat{x} = \arg \min_{x,a} \frac{1}{2} \left[ \| y - M x \|_F^2 + \sum_i \| R_i x - D a_i \|_F^2 \right] \quad s.t. \quad \| a_i \|_0 \leq s
\]

Similar to denoising task, the coordinate decent method is used to get an estimate of the latent complete image.

The training data set is constructed by randomly selecting patches of size \( \sqrt{n} \times \sqrt{n} \) from the set of training images (given in chapter I, Fig. 1.1). A dictionary \( D \) is trained using the training data set. An image is then selected randomly from the training images and divided into non-overlapping patches. In each image patch a fraction of \( m \) random pixels are deleted. For each image patch with missing pixels, the sparse coefficients are estimated under the learned dictionaries, given as

\[
\hat{a}_i = \arg \min_{a_i} \| M_i (R_i x - D a_i) \|_F^2 \quad s.t. \quad \| a_i \|_0 \leq s
\]
Where $M_i$ is the respective mask for each patch.

The final step is to reconstruct the image using the estimated sparse coefficient vectors and the learned dictionary, given as

$$\hat{x} = \left( M^T y + \sum_i R_i^T D_i \right) \left( M^T M + \sum_i R_i^T R_i \right)^{-1}$$

In image inpainting experiments we used three different tasks - i) fill-in missing image pixels, ii) text removal, and iii) scratches removal.

In addition to the above image processing applications, the proposed dictionary learning algorithms are also tested with the dictionary recovery task. In this task a synthetic signal is generated using some generating dictionary $D_g$, a small noise is added to this signal. The dictionary learning algorithms are then used to estimate a dictionary using the noisy signal, the estimated dictionary is then compared with the original generating dictionary. This experiment illustrates the dictionary recovery strength of each method.
Maximum Likelihood Based Dictionary Learning for Image Restoration

Optimization models with sparsity constraint have received widespread acceptance in image processing applications. The sparsity constraint has been efficiently utilized as regularization in the linear inverse problem. In image processing applications, the optimization with sparsity constraint consists of a model dictionary and a collection of sparse coefficient vectors, which assign weights to each dictionary element in the data representation. The dictionary, also known as basis collection, plays a vital role in the overall performance. Especially in Bayesian framework, the dictionary is treated as the main parameter set to estimate. The dictionary learning algorithms mostly consist of two stages: the sparse coding stage and dictionary update stage. This later stage can be achieved sequentially or in parallel. In this chapter, the maximum likelihood approach is used to derive a novel dictionary learning algorithm. The proposed method differs from recent dictionary learning algorithms by updating all the dictionary atoms in parallel using only one eigen-decomposition. The effectiveness of the proposed method is tested on two different image processing applications: filling-in missing pixels and noise removal.

In following, a brief review of dictionary learning methods is given in (7.1). The mathematical formulation of the problem is outlined in (7.2), followed by the proposed algorithm in section (7.3). The simulation results and comparisons are presented in (7.4), and the chapter is concluded in (7.5).
7.1 Related Methods

The literature on dictionary learning (DL) algorithms is fast growing, probabilistic and non-probabilistic approaches, have been adopted for the derivation of DL algorithms. These methods seek to uncover a linear multivariate latent structure in the observed data by imposing the constraint that the estimates of the observed variables are sparse linear functions of the regressors (dictionary atoms). The other particularity of these methods is that the set of regressors or the dictionary is also tuned iteratively to find the optimal linear multivariable latent model. Mostly these methods consist of two stages: a sparse coding stage and a dictionary updates stage. In the first stage the dictionary is kept constant and the sparsity assumption is used to produce linear approximations of the observed data. In the second stage, using the learned coefficients the dictionary is updated to minimize a cost function to achieve a certain objective. This two stage optimization is followed iteratively in almost all the DL algorithms, until some convergence. Majority of DL methods share a similar sparse coding stage and the difference appearing mainly in the dictionary update stage.

Besides the difference in the optimization used in the dictionary update stage, the dictionary update can also be be made sequential (each atom is updated separately) or in parallel (all the atoms are updated at once). While the parallel update approach may be preferred for its advantage in computational complexity, the sequential approach generally offers better results. The maximum likelihood and a-posterior based approaches are used in [Kreutz-Delgado et al. 2003a, Lewicki and Sejnowski 2000], with fixed column norm to derive an estimate of dictionary in parallel. Similarly, maximum likelihood approach is used in [Kreutz-Delgado et al. 2003b, Engan et al. 1999] to derive an estimate of the dictionary in parallel. The square of Frobenius norm is used in [Aharon et al. 2006] for sequential dictionary update where in the dictionary update stage the support of the coefficient vectors is kept constant and the non-zero values are updated. An improved and computationally efficient version of [Aharon et al. 2006] is outlined in [Smith and Elad 2013], where the SVD step is replaced with rank-1 approximation. Under the Bayesian assumption, with sparse and redundant regularization on the image, in [Elad and Aharon 2006] dictionary was targeted as the learning parameter. The sparseness constraint over the small overlapping image blocks (patches) is used in conjunction with the learned dictionary for the whole image restoration, using overlapping patch averaging. A majorization method, with a surrogate function, is used in [Yaghoobi et al. 2009a] and a first order series expansion is suggested in [Sadeghi et al. 2013] for sequential dictionary up-
date. Other examples of parallel and sequential DL algorithms are Kasiviswanathan et al. [2012], Jia et al. [2013], and Sahoo and Makur [2013]. Jiang et al. [2013], Rubinstein et al. [2010]; respectively. In contrast to the generic batch-base Aharon et al. [2006], Engan et al. [1999] DL methods, for large data size recently online DL Mairal et al. [2010], Stretting and Engan [2010], Zhou et al. [2012] has also attract attention.

In addition to the generic sparseness constraint on the representative coefficients $X$, further task related constraints on the dictionary atoms $d_i$ or sparse coefficients $x_i$ can be added to further enhance the performance. Recently variants of K-SVD Aharon et al. [2006] with additional constraints in dictionary update and sparse coding step is given in Sadegni et al. [2014]. Similarly, a regularized DL method based on proximal algorithm with Bregman-type data fitting term is presented in Salzo et al. [2014]. In Yaghoobi et al. [2008] a parallel DL algorithm is presented with constraint on bounded Frobenius and column norms. A block structured DL technique with regularization on the mutual coherence within each block of learned dictionary is outlined in Chi et al. [2013].

In this work we adopt a maximum likelihood method to derive a new approach to dictionary learning. In comparison to existing maximum likelihood methods, the proposed approach uses the profile-likelihood to derive the dictionary estimate. This leads to a novel dictionary learning algorithm that update the complete dictionary in parallel using only a single eigen-decomposition. The performance of the proposed method is compared with some existing dictionary learning method.

### 7.2 Problem Formulation

Let $X = [x_1, x_2, ..., x_P]$ where $x_i \in \mathbb{R}^n$ be the set of signal to be represented. Given a dictionary $D \in \mathbb{R}^{n \times K}$ containing a set of $K$ regressors $d_k \in \mathbb{R}^n$ for $k = 1, ..., K$, overcomplete dictionary learning algorithms generate a representation of signal $x_i$ as a sparse linear combination of the atoms $d_k$

$$\hat{x}_i = Da_i$$

where $a_i \in \mathbb{R}^K$ represent the corresponding signal strength. This signal is a sparse representation vector such that $||a_i||_0 = s << K$ ($|| \cdot \||_0$ is the $\ell_0$ norm). Overcomplete dictionary learning algorithms distinguish themselves from traditional model-based method by the fact that, in addition to $a_i$, they also train the dictionary $D$ to better fit the data set $X$. Given $X$, $D$ is trained to minimize the error $||X - DA||^2$ where
\( \mathbf{A} \in \mathbb{R}^{K \times P} \) is the matrix of sparse representations determined in the sparse coding stage. Overcomplete dictionary learning algorithms generate a solution by iteratively alternating between the sparse coding stage

\[
\hat{a}_i = \arg \min_{a_i} \| \mathbf{x}_i - \mathbf{D}a_i \|_2 \quad s.t. \| a_i \|_0 \leq T_0 \quad (7.1)
\]

for \( i = 1, \ldots, P \) and the dictionary update stage for the obtained \( \mathbf{A} \) from the sparse coding stage

\[
\mathbf{D} = \arg \min_{\mathbf{D}} \| \mathbf{X} - \mathbf{D} \mathbf{A} \|_2^2 \quad (7.2)
\]

The update step can either be sequential as in [Aharon et al. 2006] or in parallel as in [Engan et al. 1999]. The latter is preferable because it is computationally more efficient since it updates all the dictionary atoms at once using

\[
\mathbf{D} = \mathbf{X} \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1} \quad (7.3)
\]

and keep the structure of the dictionary.

The model generally used to characterize the data for the dictionary update is given by

\[
\mathbf{X} = \mathbf{D} \mathbf{A} + \mathbf{N} \quad (7.4)
\]

where the columns of \( \mathbf{N} \) are assumed i.i.d \( \mathcal{N}(0, \Sigma_N) \). The solution proposed in [Engan et al. 1999] and given in (7.3) corresponds to the maximum likelihood estimation of \( \mathbf{D} \) with \( \Sigma_N = I_N \). For the derivation of the proposed dictionary update stage, we assume the observations are the row of \( \mathbf{X} \) and that the rows of \( \mathbf{N} \) are i.i.d \( \mathcal{N}(0, \Sigma_N) \).

The log-likelihood function for the model (7.4) in this case is

\[
\begin{align*}
l(\mathbf{D}, \mathbf{A}; \mathbf{X}) &= -\frac{nP}{2} \log(2\pi) - \frac{P}{2} \log |\Sigma_N| \\
&\quad - \frac{1}{2} \text{tr} \left\{ (\mathbf{X} - \mathbf{D} \mathbf{A}) \Sigma_N^{-1} (\mathbf{X} - \mathbf{D} \mathbf{A})^\top \right\}
\end{align*}
\]

and the relevant part of the negative log-likelihood function for estimating \( \mathbf{D} \) is given by

\[
l(\mathbf{D}, \mathbf{A}; \mathbf{X}) \propto \text{tr} \left\{ (\mathbf{X} - \mathbf{D} \mathbf{A}) \Sigma_N^{-1} (\mathbf{X} - \mathbf{D} \mathbf{A})^\top \right\}. \quad (7.5)
\]

Assuming \( \mathbf{D} \) is orthogonal, \( \mathbf{D}^\top \mathbf{D} = \mathbf{I}_K \), using (7.2) an estimate of \( \mathbf{A} \) for \( \mathbf{D} \) fixed is

\[
\hat{\mathbf{A}} = \mathbf{D}^\top \mathbf{X}. \quad (7.6)
\]

Considering \( l(\mathbf{D}, \mathbf{A}; \mathbf{X}) \) as the likelihood for the data \( \mathbf{X} \), a profile likelihood for \( \mathbf{D} \) is
given by \( l(D, \hat{A}; X) \) and may be used as the cost function for \( D \). Using (7.5)

\[
l(D, \hat{A}; X) \propto \text{tr} \left\{ (X - D \hat{D}^\top X)^\top \Sigma_N^{-1} (X - D \hat{D}^\top X) \right\}
\]

\[
= \text{tr} \left\{ (X \Sigma_N^{-1} X^\top + DD^\top \Sigma_N^{-1} X^\top DD^\top - 2X \Sigma_N^{-1} X^\top DD^\top \right\}
\]

\[
= \text{tr} \left\{ X \Sigma_N^{-1} X^\top - D^\top X \Sigma_N^{-1} X^\top D \right\}
\]

(7.7)
iminizing the profile likelihood (7.7) correspond to maximizing

\[
\text{tr} \left\{ D^\top X \Sigma_N^{-1} X^\top D \right\}
\]

which corresponds to an eigen decomposition of the matrix \( X \Sigma_N^{-1} X^\top \). Therefore \( D \) correspond to the eigenvectors of \( X \Sigma_N^{-1} X^\top \) and is obtained with a single eigen decomposition of computational cost \( O(n^3) \). In a number of cases \( K << P \) and therefore we may be interested in reducing the computational cost of the dictionary update stage since only \( K \) eigenvectors are needed.

### 7.3 Proposed Method

Assume an initial \( D_0 \) such that \( D_0^\top D_0 = I \), we are interested in finding a matrix \( C \) of size \( K \times K \) such that the dictionary is updated as \( D_j = D_{j-1} C \).

According to (7.2), the estimate of \( A \) is given by

\[
\hat{A} = C^\top D_{j-1}^\top X
\]

(7.8)

and the profile likelihood \( l(C, \hat{A}; X, D_{j-1}) \) for \( C \) is given by

\[
\text{tr} \left\{ (X - D_{j-1} CC^\top D_{j-1}^\top X)^\top \Sigma_N^{-1} (X - D_{j-1} CC^\top D_{j-1}^\top X) \right\}
\]

and after some matrix manipulations, becomes

\[
\text{tr} \left\{ X \Sigma_N^{-1} X^\top - C^\top D_{j-1}^\top X \Sigma_N^{-1} X^\top D_{j-1} C \right\}.
\]

(7.9)

Minimizing the profile likelihood (7.9) is equivalent to maximizing

\[
\text{tr} \left\{ C^\top D_{j-1}^\top X \Sigma_N^{-1} X^\top D_{j-1} C \right\}.
\]

(7.10)

This corresponds to an eigen decomposition of the matrix \( D_{j-1}^\top X \Sigma_N^{-1} X^\top D_{j-1} \) of size \( K \times K \). The matrix \( C \) corresponds to the eigenvectors of \( D_{j-1}^\top X \Sigma_N^{-1} X^\top D_{j-1} \) of size
Table 7.1: Maximum Likelihood Orthogonal Dictionary Learning algorithm

<table>
<thead>
<tr>
<th>Algo.: Maximum Likelihood orthogonal dictionary learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong> Training data $X \in \mathbb{R}^{n \times P}$, no. of iterations $T$.</td>
</tr>
<tr>
<td><strong>Initializations:</strong> $D_0 = $ the first $K$ eigenvectors of $R^T$,</td>
</tr>
<tr>
<td><strong>For</strong> $i=1$ to $T$</td>
</tr>
<tr>
<td>1: <strong>Sparse Coding:</strong></td>
</tr>
<tr>
<td>\hspace{0.5cm} Find sparse coefficients $A$, using eq.(7.8)</td>
</tr>
<tr>
<td>2: $\Sigma_N = (X - DA)^T(X - DA)$.</td>
</tr>
<tr>
<td>3: <strong>Apply EVD:</strong> $(D^T X \Sigma_N X^T D) = U \Sigma U^T$,</td>
</tr>
<tr>
<td>\hspace{0.5cm} Assign first $K$ eigenvectors to $C$.</td>
</tr>
<tr>
<td>4: <strong>Dictionary Update:</strong></td>
</tr>
<tr>
<td>\hspace{0.5cm} $D_{\text{new}} = DC$.</td>
</tr>
<tr>
<td>5: $D = D_{\text{new}}$.</td>
</tr>
<tr>
<td><strong>end.</strong></td>
</tr>
<tr>
<td><strong>Output:</strong> $D$</td>
</tr>
</tbody>
</table>

$K \times K$ and is obtained with a single eigen decomposition of computational cost $O(K^3)$. The dictionary update is given in this case by $D_j = D_{j-1}C$ and therefore involves an additional matrix multiplication of computational complexity $O(nK^2)$. The main steps of the proposed algorithm are outlined in table 7.1.

The performance of this approach depends on an initial dictionary $D_0$. For faster convergence, the algorithm is started by setting the initial $D$ to the orthonormal matrix formed by the first $K$ right singular vectors from the SVD of $X$. That is, let $X = LRT^\top$, where $L$ and $R$ are orthonormal matrices, and set the initial $D_0$ as the first $K$ columns of $R^\top$.

7.4 Application Results

To demonstrate the strength of the proposed method we compare its performance with the state of the art dictionary learning methods K-SVD [Aharon et al. 2006]¹, SGK [Sahoo and Makur 2013]², and MOD [Engan et al. 1999]. The comparisons are done for two different image applications: filling-in missing pixels and noise removal. In the former case, some image pixels at random locations are deleted, and then filled their values using the learned dictionaries. In the second comparisons a noisy image restoration example is presented to demonstrate the estimation strength of each method.

¹http://www.cs.technion.ac.il/~ronrubin/software.html
²https://sites.google.com/site/sujitkusahoo/codes
7.4.1 Filling-in Missing Pixels

In this experiment a training data set is constructed using the $10 \times 10$ overlap patches from the input image. We applied K-SVD, MOD, SGK, and the proposed method to train dictionaries of size $100 \times 70$. The input complete image is divided into $P$ non-overlapping patches, in each patch fraction $m$ random pixels are deleted, set to zero, where $m \in [0.2, 0.9]$. For each patch with missing pixels, the sparse coefficients are estimated under the learned dictionaries using OMP. The resulting coefficients vector of each patch is denoted by $a_i$, where $i$ indicates the number of patch. The reconstructed patch is then obtained as,

$$\hat{x}_i = Da_i$$  \hspace{1cm} (7.11)

In table 7.2 we present the comparisons in terms of sum of squared difference (SSD), calculated using the reconstructed image and original input image, for $m = 0.4 \& 0.7$. 

Figure 7.1: Missing Pixel: Visual comparison.
Table 7.2: Filling-in missing pixels comparison in terms of SSD.

<table>
<thead>
<tr>
<th>Method</th>
<th>m = 0.4</th>
<th>m = 0.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-SVD</td>
<td>13.41</td>
<td>20.88</td>
</tr>
<tr>
<td>MOD</td>
<td>20.67</td>
<td>25.48</td>
</tr>
<tr>
<td>SGK</td>
<td>18.07</td>
<td>22.70</td>
</tr>
<tr>
<td>Proposed</td>
<td>14.09</td>
<td>21.17</td>
</tr>
</tbody>
</table>

Table 7.3: Noisy image restoration ISNR Comparisons.

<table>
<thead>
<tr>
<th>Method</th>
<th>Input PSNR</th>
<th>Input PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Barbara</td>
<td>Pepers</td>
</tr>
<tr>
<td></td>
<td>16.02</td>
<td>16.02</td>
</tr>
<tr>
<td></td>
<td>18.55</td>
<td>18.55</td>
</tr>
<tr>
<td></td>
<td>20.20</td>
<td>20.20</td>
</tr>
<tr>
<td></td>
<td>22.15</td>
<td>22.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Input PSNR</th>
<th>Input PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10.31</td>
<td>10.21</td>
</tr>
<tr>
<td></td>
<td>9.14</td>
<td>10.03</td>
</tr>
<tr>
<td></td>
<td>9.01</td>
<td>9.48</td>
</tr>
<tr>
<td></td>
<td>8.56</td>
<td>8.40</td>
</tr>
<tr>
<td>K-SVD</td>
<td>9.82</td>
<td>9.63</td>
</tr>
<tr>
<td>MOD</td>
<td>9.56</td>
<td>9.23</td>
</tr>
<tr>
<td>SGK</td>
<td>9.05</td>
<td>9.08</td>
</tr>
<tr>
<td>Proposed</td>
<td>8.16</td>
<td>8.07</td>
</tr>
<tr>
<td></td>
<td>9.04</td>
<td>9.70</td>
</tr>
<tr>
<td></td>
<td>9.37</td>
<td>9.32</td>
</tr>
<tr>
<td></td>
<td>9.06</td>
<td>9.17</td>
</tr>
<tr>
<td></td>
<td>8.25</td>
<td>8.12</td>
</tr>
</tbody>
</table>

As can be seen, the proposed method produce comparatively higher quality estimation than MOD and SGK. In case of K-SVD the results are comparatively same but the proposed method is much faster than K-SVD, as the dictionary is learned parallel where K-SVD used sequential update, requires $K$ numbers of iterations. A visual comparison for $m = 0.5$ is presented in Fig. 7.1, where a corrupted image is reconstructed using the learned dictionaries.

### 7.4.2 Noisy Image Restoration

In this section the learned dictionaries are compared in restoring a noisy image. Four different noise levels are examined, where in each case the original image is corrupted with additive white Gaussian noise of variance, $\sigma^2_N$, corresponds to PSNR decibel values [20.30, 26.56, 31.30, 40.68]. The comparisons are done using simulated noisy version of some well known grayscale test images. The Improvement-in-Signal-to-Noise-Ratio (ISNR) is used as an objective measure of the quality for the restored images, calculated as

$$\text{ISNR} = 10 \log_{10} \frac{\| x - y \|^2}{\| x - \hat{x} \|^2}$$

(7.12)

where $x$, $y$, and $\hat{x}$ represent the original, noisy and estimated images, respectively.

For each PSNR value we generated 20 results and plot the average of ISNR values. In each case a dictionary of size $100 \times 80$ is learned using the $10 \times 10$ overlap patches.
from the noisy image. The learned dictionary is then used to estimate the original image similar to [Aharon et al. 2006]. The ISNR values are summarized in table 7.3 for images of Barbara and Pipers with different PSNR levels. The proposed method produce very close results to K-SVD with more simpler and fast computations, where as better estimations are obtained compared to MOD and SGK, which are computationally similar to the proposed method. The visual comparison for the input noisy image with PSNR value of 26 dB is presented in figure 7.2.

7.5 Conclusion

In this chapter a new dictionary learning algorithm is derived. The proposed algorithm has the particularity being computationally efficient since it updates the complete dictionary in parallel using only one eigen decomposition of a matrix of
size $K \times K$ and a matrix multiplication. The performance of the proposed method was compared with existing parallel and sequential dictionary learning methods on two different image processing applications: filling-in missing pixels and noise removal. The proposed method showed higher performance than the MOD and SGK approaches on both applications. It also presented comparative results with the K-SVD which is a sequential dictionary learning algorithm.
Penalized Rank-1 Matrix Approximation for Dictionary Learning

The algorithms for learning overcomplete dictionaries in sparse signal representation are mostly iterative minimization methods, that alternate between a sparse coding stage and a dictionary update stage. In all these dictionary learning methods, the sparsity constraint has been confined to the sparse coding stage, while the difference between existing dictionary learning algorithms has mainly been maintained in the dictionary update stage. The singular value decomposition (SVD) has been successfully used in sequential dictionary update, producing finer tuned dictionary atoms.

In this chapter, we look at the impact on performance of dictionary learning approaches when the sparsity constraint is further enforced in the dictionary update stage. Using the connection between alternate conditional least squares and rank one matrix approximation, an effective approach is proposed to include a sparsity constraint in the dictionary update stage. Within this framework, we not only propose dictionary update stages that can reinforce sparsity, but can also enforce other forms of constraint such as smoothness of the dictionary atoms effectively. Furthermore the proposed dictionary updates are computationally more efficient, compared to the use of the SVD. Experiments on synthetic tests and applications on image data, illustrating the considerable performance improvement obtained with the proposed approaches are provided.

In section 8.1 a brief description of some related methods are given, followed by the mathematical formulation in section 8.2. The general idea of sequential dictionary update with enforced sparsity is presented in section 8.3 and the proposed method is explained in section 8.4. The simulation results are given in section 8.5. The chapter
is concluded in section 8.6.

8.1 Related Work

Whether it is for restoration, filtering, compression, or higher level tasks such as extracting meaningful features or uncovering structural relationships, transforming signals and images to other domains and characterizations with desirable properties constitutes the heart of signal and image processing. The last few years have seen sparse signal representations and overcomplete dictionary learning methods [Tosic and Frossard 2011] take an important place in signal and image processing offering solutions that outperform classical approaches in most cases. Dictionary learning methods have proven successful in solving inverse problems arising in a variety of signal and image processing problems such as Gaussian image denoising [Elad and Aharon 2006], multiplicative noise removal [Huang et al. 2012], Poisson image denoising [Giry and Elad 2014], color image restoration [Mairal et al. 2008], magnetic resonance image reconstruction [Ravishankar and Bresler 2011], low-dose X-ray CT reconstruction [Xu et al. 2012], image formation in wide-angle synthetic aperture radar (SAR) [Varshney et al. 2008], hyperspectral image classification [Soltani-Farani et al. 2015], image compression [Zepeda et al. 2011, Bryt and Elad 2008], synthetic aperture radar (SAR) image compression [Zhan et al. 2013], face recognition [Wright et al. 2009, Jiang et al. 2013], object recognition [Mairal et al. 2012], speech denoising [Jafari and Plumbley 2009], in-painting [Mairal et al. 2010], and fMRI data analysis [Lee et al. 2011, Eavani et al. 2012]. While the underlying key constraint of all dictionary learning methods is that the observed signal is sparse, which explicitly means that it can be adequately represented using a small number of dictionary atoms, their particularity is that the dictionary is also learned to find the optimal dictionary that best describes the observed signal.

Given a data set $X \in \mathbb{R}^{n \times P}$, overcomplete dictionary learning methods find a dictionary matrix $D \in \mathbb{R}^{n \times K}$, $P > K > n$, with unit column norms and a sparse coefficient matrix also known as the sparse codes $A \in \mathbb{R}^{K \times P}$ such that they solve

$$\min_{D, A} \|X - DA\|_F^2 \text{ s.t. } \|a_i\|_0 \leq s, \forall \ 1 \leq i \leq P.$$ 

where the $a_i$’s are the column vectors of $A$, $\| \cdot \|_0$ is the $\ell_0$ quasi-norm, which counts the number of nonzero coefficients. Most of these methods consists of a two stage optimization scheme: a sparse coding stage and a dictionary update stage. In the first stage the dictionary is kept constant and the sparsity constraint is used to produce a
sparse linear approximation of the observed data. A number of sparse coding algorithms have been proposed to perform this stage, among them orthogonal matching pursuit \cite{Pati:1993}, subspace pursuit \cite{Dai:2009} and the \ell_1 minimization \cite{Osborne:2000}. In the second stage, based on the current sparse codes, the dictionary is updated to minimize a cost function to achieve a certain objective. Different cost functions have been used for the dictionary update to achieve different objectives. For Example, the Frobenius norm with column normalization has been widely used. The dictionary learning methods iterate between the sparse coding stage and a dictionary update stage until convergence. The performance of the overcomplete dictionary learning methods strongly depend on the dictionary update stage since most of these methods share a similar sparse coding stage.

Besides the difference in the approach used to update the dictionary, the dictionary update can be made sequential where each dictionary atom (column \(\mathbf{d}_i, i = 1, ..., K\) of \(\mathbf{D}\)) is updated separately, for example as in \cite{Aharon:2006} or in parallel where the dictionary atoms are updated all at once as in \cite{Engan:1999} for example. For a specific approach (characterized by the cost function used to update the dictionary), the parallel update will generally require a lower computational complexity cost than the sequential update that uses the same approach for example the Frobenius norm to reduce the representation error. While the parallel update approach may be preferred for its advantage in the computational cost, the sequential approach generally offers better performance because it produces finer-tuned dictionaries compared to the parallel approach.

Most proposed algorithms have kept the two stages optimization procedure, the difference appearing mainly in the dictionary update stage with some exceptions having a difference in the sparse coding stage \cite{Tosic:2011}. The maximum likelihood and aposteriori approaches have been used in \cite{Kreutz-Delgado:2003b,Lewicki:2000,Engan:1999} to derive an estimate of the dictionary in parallel. The square of the Frobenius norm was used in \cite{Aharon:2006} to derive a sequential dictionary update stage whereas the parallel dictionary update stage derived in \cite{Engan:1999} can also be considered as derived from this principle. Improved variants of these two algorithms were proposed in \cite{Smith:2013}. Based on a new optimization framework, a generalization of \cite{Aharon:2006} was proposed in \cite{Dai:2012}, where as in \cite{Yaghoobi:2009a} an algorithm based on a majorization method was derived. In \cite{Sadeghi:2013} a first order series expansion for the factorization function was used with the Frobe-
Penalized Rank-1 Matrix Approximation for Dictionary Learning

nius norm. Other examples of parallel dictionary learning algorithms can be found in [Stretting and Engan, 2010; Kasiviswanathan et al., 2012; Jia et al., 2013] whereas examples of other sequential dictionary learning algorithms can be found in [Sahoo and Makur, 2013; Jiang et al., 2013; Rubinstein et al., 2010]. Furthermore, dictionary learning algorithms can also be online as in [Mairal et al., 2010] and [Stretting and Engan, 2010]. While most proposed algorithms have kept the two stages optimization procedure, some dictionary learning algorithms that do not rely on alternating between these two stages have also been proposed as in [Hawe et al., 2013; Rakotomamonjy, 2013].

All dictionary learning algorithms cited above have used the sparsity constraint in the sparse coding stage. From this stage, besides the sparse codes, a sparsity pattern information is also derived, i.e. the associated non-zeros entries positions. Some of the above algorithms have either used this sparsity pattern information in the dictionary update stage as in [Aharon et al., 2006; Dai et al., 2012; Smith and Elad, 2013] or ignored it as in [Engan et al., 1999; Stretting and Engan, 2010; Sahoo and Makur, 2013]. The algorithms proposed in this paper avoid following one of these paths. Particularly, we revisit one of the most popular dictionary learning method [Aharon et al., 2006] where we include a sparsity constraint also in the dictionary update stage such that with each dictionary atoms and its associated sparse code, the sparsity pattern is also updated. As revealed by the results obtained from different experiments, including the sparsity constraint also in the second stage of the algorithm appears to be a good idea. With many data sets, regularizing the dictionary elements or atoms to encourage smoothness of the data set in the columns direction may be of interest. This is for example the case in functional magnetic resonance imaging (fMRI) studies and specially in resting-state studies where a primary goal is to find major brain activation patterns. In this case the data matrix $X$ is formed by vectorizing each time point creating a matrix $n \times P$ where $n$ is the number of time points and $P$ the number of voxels ($\approx 10,000 - 100,000$). While we expect to have only a limited number of voxel active at each time point, it is also expected to have continuous activity along the time. The signal at a fixed voxel over time is believed to be smooth and of low frequency. We therefore first develop a dictionary learning algorithm that is adapted to such data set by enforcing smoothness of the dictionary atoms. Motivated by the results obtained with the proposed dictionary learning algorithm that include a sparsity constraint also in the dictionary update stage, we then develop a dictionary update stage such that it includes both a smoothness constraint on the dictionary atoms and reinforce the sparsity constraint. All proposed dictionary update stages are derived within the penalized matrix decomposition framework [Witten et al., 2009].
8.2 Mathematical formulation

Given a set of signals \( X = [x_1, x_2, ..., x_P] \), a learned dictionary is a collection of vectors or atoms \( d_k, k = 1, ..., K \) that can be used for optimal linear representation. Usually the objective is to find a linear representation for the set of signals \( X \)

\[
\{D, A\} = \arg \min_{D,A} \| X - DA \|_F^2
\]

where \( D = [d_1, d_2, ..., d_K] \), that makes the total representation error as small as possible. This optimization problem is ill-posed unless extra constraints are imposed on the dictionary \( D \) and the sparse codes \( A \). The common constraint on \( A \) is the sparseness. Let the sparse coefficient vectors \( a_i, i = 1, ..., P \) constitute the columns of the matrix \( A \), with this constraint on \( A \), the above objective can be re-stated as the minimization problem

\[
\{D, A\} = \arg \min_{D,A} \| X - DA \|_F^2
\]

s.t. \( \| a_i \|_0 \leq s, \forall 1 \leq i \leq P \)

where \( s \ll K \). To prevent \( D \) from being arbitrarily large and therefore have arbitrarily small values of \( a_i \), it is common to constrain its columns to have unit norm. Finding the optimal \( s \) corresponds to a problem of model order selection that can be resolved using a univariate linear model section criterion Seghouane and Bekara [2004]. The generally used optimization strategy, not necessarily leading to a global minimum consists in splitting the problem into two stages which are alternately solved within an iterative loop. These two stages are, first, the sparse coding stage, where \( D \) is fixed and the sparse coefficient vectors are found by solving

\[
\hat{a}_i = \arg \min_{\hat{a}_i} \| x_i - Da_i \|^2 \quad \text{s.t.} \quad \| a_i \|_0 \leq s, \forall i = 1, ..., P
\]

(8.1)

While there are a number of alternative penalties to the \( \ell_0 \) quasi-norm that encourage sparsity and approximately solve the sparse coding stage, the simple and popular alternative: the \( \ell_1 - \text{norm} \) penalty a direct convex relaxation of the \( \ell_0 \) quasi-norm, Tibshirani [1994] Osborne et al. [2000] is widely adopted and used here. The dictionary update stage where \( A \) is fixed and \( D \) is derived by solving

\[
D = \arg \min_{D} \| X - DA \|_F^2
\]

(8.2)
followed by normalizing its columns constitutes the second stage.

This is where sequential and parallel update methods differ. In parallel update methods all dictionary atoms are updated in parallel using least squares [Engan et al. 1999; Stretting and Engan 2010] or maximum likelihood [Kreutz-Delgado et al. 2003b; Lewicki and Sejnowski 2000] whereas sequential update methods [Aharon et al. 2006; Sahoo and Makur 2013] break the global minimization (8.2) into $K$ sequential minimization problems. In the method proposed in [Aharon et al. 2006], which has become a benchmark in dictionary learning, each column $d_k$ of $D$ and its corresponding row of coefficients $a^{row}_k$ are updated based on a rank-1 matrix approximation of the error for all the signals when $d_k a^{row}_k$ is removed.

$$\{d_k, a^{row}_k\} = \arg\min_{d_k, a^{row}_k} \|X - DA\|_F^2$$

$$= \arg\min_{d_k, a^{row}_k} \left\| \left(X - \sum_{i=1, i \neq k}^K d_i a^{row}_i\right) - d_k a^{row}_k \right\|_F^2$$

$$= \arg\min_{d_k, a^{row}_k} \|E_k - d_k a^{row}_k\|_F^2. \quad (8.3)$$

The singular value decomposition (SVD) of $E_k = U \Delta V^\top$ is used to find the closest rank-1 matrix approximation of $E_k$ [Eckart and Young 1936]. In this case, $d_k$ could be updated by taking the first column of $U$ and the $a^{row}_k$ could be updated by taking the first column of $V$ multiplied by the first diagonal element of $\Delta$. This form of update corresponds to a dictionary update stage that ignores the sparsity pattern information derived in the sparse coding. A dictionary update stage that uses the sparsity pattern information with (8.3) can be obtained by avoiding the loss of sparsity in $a^{row}_k$ that will be created by the direct application of the SVD on $E_k$. This solution was adopted in [Aharon et al. 2006] where it was proposed to modify only the nonzero entries of $a^{row}_k$ by taking into account only the signals $x_i$ that use the atom $d_k$ in $d_k a^{row}_k$ or by taking the SVD of $E_k^w = E_k I_{|w_k|}$ where $w_k = \{i|1 \leq i \leq P; a^{row}_k(i) \neq 0\}$ and $I_{|w_k|}$ the $N \times |w_k|$ submatrix of the $P \times P$ identity matrix obtained by retaining only those columns whose index numbers are in $w_k$, instead of the SVD of $E_k$.

The motivation of the proposed approach comes from the observation that the rank-1 approximation obtained using the SVD and written as $d_k a^{row}_k$ can also be approximated by applying few iterations of the power method for computing the SVD [Golub and f. Van Loan 1996]. Recall that the power method, or alternating least square method, sequentially estimates $d_k$ with $a^{row}_k$ fixed and vice versa by solving the least
square problem

\[ \| E_k^R - d_k a_k^{row} \|_F^2 = \text{tr} \left\{ (E_k^R - d_k a_k^{row}) (E_k^R - d_k a_k^{row})^T \right\} \]

\[ = \| E_k^R \|_F^2 - 2d_k^T E_k^R a_k^{row}^T + \| d_k \|_2^2 \| a_k^{row} \|_2^2 \]

(8.4)

and then rescaling the estimates to give

\[ d_k = \frac{E_k^R a_k^{row}^T}{\| E_k^R a_k^{row} \|_2} \]

(8.5)

\[ a_k^{row} = d_k E_k^R \]

(8.6)

These equations can be used to justify the power algorithm, which, if initialized randomly, converges almost surely to a least square rank one fit. Using this observation a dictionary update stage can be obtained by iterating (8.5) and (8.6) until convergence or by applying only few iterations of these equations instead of the computationally expensive SVD of \( E_k^R \). Therefore, all the techniques used to improve least squares estimation can be used in the dictionary update stage to improve the performance of the existing dictionary learning method based on the SVD.

### 8.3 Sequential dictionary update with enforced sparsity

As indicated in paragraph five of the introduction all dictionary learning algorithms have either used or ignored the sparsity pattern information in the dictionary update stage. As mentioned above after (8.3), in [Aharon et al. 2006], the sparsity pattern information was included in the dictionary update stage by taking the SVD of \( E_k^R \) instead of the SVD of \( E_k \) since the former will only modify the nonzero entries of \( a_k^{row} \). This approach has lead to an improved performance dictionary learning algorithm as it was observed in a number of experiments provided in [Aharon et al. 2006].

The algorithms proposed in this paper avoid following one of these paths. Indeed, an alternative approach to using the sparsity pattern information is: instead of only updating the nonzero entries of \( a_k^{row} \) in the dictionary update stage we propose to re-update all the entries of \( a_k^{row} \) and the sparsity pattern information. The resulting problem is a regularized rank one matrix approximation where the penalty is introduced in the minimization problem to promote sparsity of \( a_k^{row} \). [Shen and Huang]
Penalized Rank-1 Matrix Approximation for Dictionary Learning

[2008] Witten et al. [2009]. Form the connection of principal component analysis (PCA) with SVD this can also be seen as a problem of sparse PCA [Jolliffe et al. 2003] where the \( \ell_1 \) penalty is used to promote sparsity of the loading vectors and improve the interpretability of PCA.

Given the above description, we seek an optimization framework to achieve a rank one approximation \( \mathbf{d}_k \mathbf{a}^{\text{row}}_k \) with a sparse vector \( \mathbf{a}^{\text{row}}_k \) where \( \mathbf{a}^{\text{row}}_k = \lambda \tilde{\mathbf{a}}^{\text{row}}_k \), \( \lambda \) being the largest singular value of \( \mathbf{E}_k \) and \( \tilde{\mathbf{a}}^{\text{row}}_k \) the normalized sparse right singular vector of \( \mathbf{E}_k \). From the numerous proposals for sparse PCA we adopt the popular penalized regression approach [Shen and Huang 2008]. With this approach, the updates of \( \mathbf{d}_k \) and \( \mathbf{a}^{\text{row}}_k \) are obtained by alternating minimization of

\[
\{ \mathbf{d}_k, \mathbf{a}^{\text{row}}_k \} = \arg \min_{\mathbf{d}_k, \mathbf{a}^{\text{row}}_k} \| \mathbf{E}_k - \mathbf{d}_k \mathbf{a}^{\text{row}}_k \|_F^2 + \alpha \| \mathbf{a}^{\text{row}}_k \|_1
\]

subject to \( \| \mathbf{d}_k \|_2 = 1 \)  

where \( \alpha \) is a non-negative penalty parameter controlling the amount of sparsity in \( \mathbf{a}^{\text{row}}_k \) (increasing \( \alpha \) increases the amount of sparsity in \( \mathbf{a}^{\text{row}}_k \)) and the penalty encouraging sparsity is taken as the \( \ell_1 \)-norm. The use of \( \mathbf{a}^{\text{row}}_k \) instead of \( \tilde{\mathbf{a}}^{\text{row}}_k \) in the \( \ell_1 \) penalty is justified by the fact that \( \tilde{\mathbf{a}}^{\text{row}}_k \) is a unit vector and thus subject to scale constraint. This in turn would invalidate its use with the \( \ell_1 \) penalty. For fixed \( \mathbf{d}_k \) and \( \| \mathbf{d}_k \|_2 = 1 \), the \( \mathbf{a}^{\text{row}}_k \) that minimizes (8.7) is derived from

\[
\mathbf{a}^{\text{row}}_k = \arg \min_{\mathbf{a}^{\text{row}}_k} \| \mathbf{a}^{\text{row}}_k \|_2^2 + \alpha \| \mathbf{a}^{\text{row}}_k \|_1 - 2 \mathbf{d}_k \mathbf{E}_k \mathbf{a}^{\text{row}}_k \mathbf{a}^{\text{row}}_k \mathbf{E}_k^\top \quad (8.8)
\]

which gives

\[
\mathbf{a}^{\text{row}}_k = \text{sgn}(\mathbf{d}_k \mathbf{E}_k) \left( |\mathbf{d}_k \mathbf{E}_k| - \frac{\alpha}{2} \mathbf{1}_p \right)_+ \quad (8.9)
\]

whereas, for fixed \( \mathbf{a}^{\text{row}}_k \), the \( \mathbf{d}_k \) that minimizes (8.7) is derived from

\[
\mathbf{d}_k = \arg \min_{\mathbf{d}_k} -2 \mathbf{d}_k \mathbf{E}_k \mathbf{a}^{\text{row}}_k \mathbf{a}^{\text{row}}_k \mathbf{E}_k^\top + \| \mathbf{d}_k \|^2 \| \mathbf{a}^{\text{row}}_k \|^2 \quad (8.10)
\]

\[1\text{where } (x)_+ \text{ represent "the positive part of", i.e., } (x)_+ = x \text{, if } x > 0 \text{, and } (x)_+ = 0 \text{, if } x \leq 0 \text{ and } \text{sgn}() \text{ is the sign operator.} \]
which with the constraint $\|d_k\|_2 = 1$ gives

$$d_k = \frac{E_k a_k^\text{row}^T}{||E_k a_k^\text{row}||_2}$$

(8.11)

where $1_{(P)}$ is a vector of ones of size $P$. The resulting dictionary learning algorithm is depicted in table 8.1. Instead of the SVD of $E_k$ [Aharon et al. 2006], the updates of $d_k$ and $a_k^\text{row}$ are found by iterating (8.9) and (8.11) until convergence. This updating strategy is similar to using an alternating minimization procedure to calculate the first singular values and first left and right singular vectors obtained in the SVD [Golub and f. Van Loan 1996] and given by (8.5) and (8.6). The sparsity constraint on $a_k^\text{row}$ in (8.7) lead to the loss of the orthogonality property between $\tilde{a}_k^\text{row}$ and the subsequent singular vectors. This is the price to pay to obtain a sparse $\tilde{a}_k^\text{row}$. However in comparison to [Shen and Huang 2008], since we are only interested in rank one approximation this is not an issue since we are not generating the subsequent vectors. The selection of the penalty parameter $\alpha$ can be obtained using a model selection criterion or cross validation. The simplified dictionary update stage used in this paper is obtained by applying a single iteration of (8.9) and (8.11) rather than alternating until convergence. Little advantage was found in the subsequent experimental results in repeating the iteration of (8.9) and (8.11) more than one time. The computational cost of this iteration is $O(nP)$ compared to the $O(ln^2|w_k| + l'|w_k|^3)$ computational cost of an SVD [Golub and f. Van Loan 1996] used in [Aharon et al. 2006].

### 8.4 Sequential dictionary update with enforced constraints

#### 8.4.1 Update with smooth dictionary atoms

With a number of data sets $X$ we may be interested in obtaining smooth dictionary atoms to encourage smoothness in the column direction of $X$. Among the options for regularized penalties that can be used in the cost function used in the dictionary update stage to encourage smoothness of the dictionary atoms, we focus here on the widely used $\ell_2$ penalty defined by

$$d_k^\text{row}^T \Omega d_k = d_k^2(1) + d_k^2(n) + \sum_{i=2}^{n-1} (d_k(i+1) - 2d_k(i) + d_k(i-1))^2$$

where $\Omega$ is a non-negative definite roughness penalty matrix used to penalize the second differences [Rice and Silverman 1991] [Ramsay and Silverman 2005] [Huang et al. 2008]. From the previous descriptions, the updates of $d_k$ and $a_k^\text{row}$ that gen-
Table 8.1: Sequential dictionary learning with enforced sparsity

<table>
<thead>
<tr>
<th>Algorithm S1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong> X, D_{ini}, s, α, and J.</td>
</tr>
<tr>
<td><strong>Set</strong> D = D_{ini}</td>
</tr>
<tr>
<td><strong>For</strong> i = 1 to J</td>
</tr>
<tr>
<td><strong>1:</strong> Sparse Coding Stage:</td>
</tr>
<tr>
<td>Find sparse coefficients A, by approximately solving</td>
</tr>
<tr>
<td>( \hat{a}<em>i = \arg \min</em>{a_i} | x_i - D \hat{a}_i |_2^2; )</td>
</tr>
<tr>
<td>subject to ( | a_i |_0 \leq s \ \ \ i = 1, ..., P )</td>
</tr>
<tr>
<td><strong>2:</strong> Dictionary Update Stage:</td>
</tr>
<tr>
<td>For each column ( k = 1, 2, ..., K ) in D,</td>
</tr>
<tr>
<td><strong>2.a:</strong> Compute the error matrix using</td>
</tr>
<tr>
<td>( E_k = X - \sum_{i=1, i \neq k}^K d_i a_i )</td>
</tr>
<tr>
<td><strong>2.b:</strong> Update the row ( a_{row}^k ) and its sparsity using</td>
</tr>
<tr>
<td>( a_{row}^k = \text{sgn}(d_k^T E_k) \left(</td>
</tr>
<tr>
<td><strong>2.c:</strong> Update the dictionary atom ( d_k ) using</td>
</tr>
<tr>
<td>( d_k = \frac{E_k a_{row}^k}{</td>
</tr>
<tr>
<td><strong>end.</strong></td>
</tr>
<tr>
<td><strong>Output:</strong> D, A</td>
</tr>
</tbody>
</table>

Erates smooth dictionary atoms in the dictionary update stage can be obtained by alternating minimization of

\[
\{d_k, a_{row}^k\} = \arg \min_{d_k, a_{row}^k} \|E_k^R - d_k a_{row}^k\|_F^2 + \alpha d_k^T \Omega d_k
\]

\[
= \arg \min_{d_k, a_{row}^k} \|E_k^R\|_F^2 - 2d_k^T E_k^R a_{row}^k \\
+ \|d_k\|_2^2 \|a_{row}^k\|_2^2 + \alpha d_k^T \Omega d_k
\]

subject to \( \|d_k\|_2 = 1 \) (8.12)

For fixed \( a_{row}^k \), the \( d_k \) that minimizes (8.12) is derived from

\[
d_k = \arg \min_{d_k} -2d_k^T E_k^R a_{row}^k + \|d_k\|_2^2 \|a_{row}^k\|_2^2 + \alpha d_k^T \Omega d_k
\]

which gives

\[
d_k = \left( \|a_{row}^k\|_2^2 I_n + \alpha \Omega \right)^{-1} E_k^R a_{row}^k
\]

with \( d_k = \frac{d_k}{||d_k||_2} \) (8.14)
Table 8.2: Sequential dictionary learning with regularized dictionary atoms

<table>
<thead>
<tr>
<th>Algorithm S2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong> X, D_{ini}, s, α and J.</td>
</tr>
<tr>
<td><strong>Set</strong> D = D_{ini}</td>
</tr>
<tr>
<td><strong>For</strong> i=1 to J</td>
</tr>
<tr>
<td><strong>1:</strong> Sparse Coding Stage:</td>
</tr>
<tr>
<td>Find sparse coefficients X, by approximately solving</td>
</tr>
<tr>
<td>( \hat{a}<em>i = \arg \min</em>{a} | x_i - D a_i |_2^2 );</td>
</tr>
<tr>
<td>subject to ( | a_i |_0 \leq s )</td>
</tr>
<tr>
<td><strong>2:</strong> Dictionary Update Stage:</td>
</tr>
<tr>
<td>For each column k = 1, 2, ..., K in D,</td>
</tr>
<tr>
<td>2.a: Compute the error matrix using</td>
</tr>
<tr>
<td>( E_k = X - \sum_{i=1, i \neq k}^{K} d_i a_i )</td>
</tr>
<tr>
<td>2.b: Using ( w_k ) the set of indices in A that uses ( d_k ) generate</td>
</tr>
<tr>
<td>( E^R_k = E_k I_{w_k} )</td>
</tr>
<tr>
<td>2.c: Update the dictionary atom ( d_k ) using</td>
</tr>
<tr>
<td>( d_k = (|a_i^{row}|_2^2 I_n + \alpha \Omega)^{-1} E_k^R a_k^{row\top} )</td>
</tr>
<tr>
<td>2.d: Normalize the dictionary atom ( d_k ) using</td>
</tr>
<tr>
<td>( d_k = \frac{d_k}{|d_k|_2} )</td>
</tr>
<tr>
<td>2.e: Update the ( w_k ) none zero entries of the row ( a_k^{row} ) using</td>
</tr>
<tr>
<td>( a_k^{row} = d_k^\top E_k^R )</td>
</tr>
<tr>
<td><strong>end.</strong></td>
</tr>
<tr>
<td><strong>Output:</strong> D, A</td>
</tr>
</tbody>
</table>

where the scaling corresponds to the constraint \( \|d_k\|_2 = 1 \). For fixed \( d_k \) and \( \|d_k\|_2 = 1 \), the \( a_k^{row} \) that minimizes (8.12) is derived from

\[
 a_k^{row} = \arg \min_{a} \|a_k^{row}\|_2^2 - 2d_k^\top E_k^R a_k^{row\top} \tag{8.15}
\]

which gives

\[
 a_k^{row} = d_k^\top E_k^R \tag{8.16}
\]

Note that (8.16) is used to update only the \( w_k \) non zeros elements of \( a_k^{row} = a_k^{row} I_{w_k} \) obtained from the sparse coding stage stage (8.1). The algorithm presented in table 8.2 is similar to the one depicted in table 8.1 where the steps 2.c and 2.b associated with the updates of \( d_k \) and \( a_k^{row} \) are replaced by (8.14) and (8.16). In this case it is the orthogonality property between \( d_k \) and the subsequent singular vectors that is lost. Assuming \( P < n^2 \), the computational cost of a simplified dictionary update using a single iteration of (8.14) and (8.16) is dominated by the computational cost of the matrix inversion which is \( O(n^3) \).
8.4.2 Update with enforced sparsity and smooth dictionary atoms

A dictionary update stage that combines both sparsity update as well as smooth dictionary atoms can be obtained by updating $d_k$ and $a_{k}^{row}$ by alternating minimization of

$$\{d_k, a_{k}^{row}\} = \arg \min_{d_k, a_{k}^{row}} \|E_k - d_k a_{k}^{row}\|_F^2 + \alpha_1 \|a_{k}^{row}\|_1$$

$$+ \alpha_2 d_k^\top \Omega d_k$$

subject to $\|d_k\|_2 = 1$

(8.17)

As for the minimization of (8.7) and (8.12), the updates of $d_k$ and $a_{k}^{row}$ can be obtained by iterative alternating minimization, i.e., first fixing $d_k$, the $a_{k}^{row}$ that minimizes (8.17) is derived from

$$a_{k}^{row} = \arg \min_{a_{k}^{row}} \|a_{k}^{row}\|_2^2 + \alpha \|a_{k}^{row}\|_1 - 2 d_k^\top E_k a_{k}^{row\top}.$$  

and gives

$$a_{k}^{row} = \text{sgn}(d_k^\top E_k) \left( d_k^\top E_k - \frac{\alpha_1}{2} \mathbf{1}_{(P)} \right) +$$

(8.18)

then fix $a_{k}^{row}$ to derive $d_k$ as

$$d_k = \arg \min_{d_k} -2 d_k^\top E_k a_{k}^{row\top} + \|d_k\|^2 \|a_{k}^{row}\|^2 + \alpha d_k^\top \Omega d_k$$

which gives

$$d_k = \left( \|a_{k}^{row}\|^2 I_n + \alpha_2 \Omega \right)^{-1} E_k a_{k}^{row\top}$$

with

$$d_k = \frac{d_k}{\|d_k\|_2}$$

(8.19)

The estimates of $d_k$ and $a_{k}^{row}$ are obtained by iterating between these two steps until convergence. The derived dictionary learning algorithm illustrated in table 8.3 is also similar to the one depicted in table 8.1 where the steps 2.b and 2.c associated with the update of $a_{k}^{row}$ and $d_k$ are replaced by (8.18) and (8.19).

In the case of combining both sparsity update and regularized dictionary atoms, the orthogonality property is lost among $d_k$ and the subsequent singular vectors as well as among $a_{k}^{row}$ and the subsequent singular vectors. The simplified regularized and sparse dictionary update stage used in this paper is obtained by applying a single iteration of (8.18) and (8.19) rather alternating until convergence. Assuming $P < n^2$, the computational cost of this iteration is dominated by the computational cost of the matrix inversion which is $O(n^3)$. At the completion of our methods, recent
Experimental results

Table 8.3: Dictionary learning with regularized atoms and enforced sparsity

<table>
<thead>
<tr>
<th>Algorithm S3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong> ( X, D_{ini}, s, \alpha_1, \alpha_2, ) and ( J. )</td>
</tr>
<tr>
<td><strong>Set:</strong> ( D = D_{ini}. )</td>
</tr>
<tr>
<td><strong>For</strong> ( i=1 ) to ( J )</td>
</tr>
<tr>
<td>1: <strong>Sparse Coding Stage:</strong></td>
</tr>
<tr>
<td>Find sparse coefficients ( A_i ) by approximately solving ( \hat{a}<em>i = \arg \min</em>{a_i} | x_i - Da_i |_2^2; )</td>
</tr>
<tr>
<td>subject to ( | a_i |_0 \leq s ) ( i = 1, ..., P. )</td>
</tr>
<tr>
<td>2: <strong>Dictionary Update Stage:</strong></td>
</tr>
<tr>
<td>For each column ( k = 1, 2, ..., K ) in ( D, )</td>
</tr>
<tr>
<td>2.a: Compute the error matrix using ( E_k = X - \sum_{i=1,i\neq k}^{K} d_i a_i ) row ( )</td>
</tr>
<tr>
<td>2.b: Update the row ( a_k ) row using ( a_k ) row = \text{sgn}(d_k \top E_k) \left(</td>
</tr>
<tr>
<td>2.c: Update the dictionary atom ( d_k ) using ( d_k = \left( | a_k |_2^2 I_n + \alpha_2 \Omega \right)^{-1} E_k a_k ) row ( \top )</td>
</tr>
<tr>
<td>2.d: Normalize the dictionary atom ( d_k ) using ( d_k = \frac{d_k}{| d_k |_2} )</td>
</tr>
<tr>
<td><strong>end.</strong></td>
</tr>
<tr>
<td><strong>Output:</strong> ( D, A. )</td>
</tr>
</tbody>
</table>

work on a slightly similar approach for rank one matrix approximation came out to our attention Sadegni et al. [2014].

8.5 Experimental results

In this section we evaluate the performance of the controlled dictionary learning algorithms, derived using the proposed dictionary update stages, with synthetic and image data. An extensive comparisons of results with some well known dictionary learning algorithms, including K-SVD Aharon et al. [2006], MOD Engan et al. [1999], MMDL Yaghoobi et al. [2009b], SGK Sahoo and Makur [2013], SepDL Hawe et al. [2013], RLS Skretting and Engan [2010], ImprDL Smith and Elad [2013] and ODL ? are presented in three different applications. In all experiments the initial overcomplete dictionary is generated by the DCT transform and \( \ell_1 \) minimization Tibshirani [1994], Osborne et al. [2000] is used for the sparse coefficients estimation.

In the synthetic signal test, the dictionary learning algorithms are evaluated on their performance to recreate the ground truth dictionary, which was used to generate the
Table 8.4: Average number of restored dictionary atoms.

<table>
<thead>
<tr>
<th>Method</th>
<th>Sparsity level (s)</th>
<th>Sparsity level (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNR 20 dB</td>
<td>SNR 35 dB</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>K-SVD</td>
<td>61.17</td>
<td>84.32</td>
</tr>
<tr>
<td>MOD</td>
<td>58.21</td>
<td>82.60</td>
</tr>
<tr>
<td>SGK</td>
<td>57.77</td>
<td>83.11</td>
</tr>
<tr>
<td>MMDL</td>
<td>51.67</td>
<td>81.40</td>
</tr>
<tr>
<td>RLS</td>
<td>26.17</td>
<td>24.10</td>
</tr>
<tr>
<td>ImprDL</td>
<td>60.75</td>
<td>84.01</td>
</tr>
<tr>
<td>S1</td>
<td>88.10</td>
<td>93.31</td>
</tr>
<tr>
<td>S2</td>
<td>80.11</td>
<td>92.06</td>
</tr>
<tr>
<td>S3</td>
<td>75.60</td>
<td>87.45</td>
</tr>
</tbody>
</table>

In case of image data, the comparisons are presented for two well known image processing tasks: image denoising and in-painting. Images are known to admit a sparse representation over a dictionary and present a good test case for sparsity based dictionary learning algorithms. For the image data tests the simulations are performed using a training set of some well known gray scale images: Lena, Barbara, Cameraman, Boat, House, Fingerprints, Peppers, Flinstones, and Kids of size $256 \times 256$ and $512 \times 512$.

We have used the respective online available codes for each algorithm: ODL\(^2\); SepDL\(^3\); SGK\(^4\); K-SVD and ImprDL\(^5\); MOD, RLS and MMDL\(^6\). All the comparison results are generated under the same software and hardware environment.

### 8.5.1 Synthetic data

Similar to the reported works [Aharon et al. (2006), Kreutz-Delgado et al. (2003b) and Engan et al. (1999)] the dictionary learning algorithms are evaluated with a synthetically generated signal where the learned dictionaries are compared with the original dictionary that generates the signal. This test will demonstrate the dictionary learning accuracy for each method. A generating dictionary, $D_g$, of size $20 \times 100$ is generated with i.i.d. uniformly distributed random entries. Each column (atom) of $D_g$ is normalized to unit $\ell_2$ norm. We generated a training signals set $X$ of 2000

\(^2\)http://spams-devel.gforge.inria.fr/downloads.html
\(^3\)http://www.gol.ei.tum.de/index.php
\(^4\)https://sites.google.com/site/sujitkusahoo/codes
\(^5\)http://www.cs.technion.ac.il/elad/software/
\(^6\)http://www.small-project.eu/keyresults/learning-structures-in-dictionaries
signals of dimension 20 \(,\) with the absolute value of non-zero entries \(\in [0.1, 1]\). Each training signal is created by a linear combination of randomly located \(s\) atoms from \(D_g\) \(,\) where \(s \in \{2, 3, 4, 5\}\). Finally a white Gaussian noise is added to each training signal to maintain a uniform signal to noise ratio (SNR) \(\in [15, 20, 25, 30, 35, 40]\) dB.

In each algorithm the learned dictionary, \(D_l\), is initialized with a (same) set of training signals. The experiments are run for different sparsity levels, \(s\), and for each sparsity level the algorithms are iterated \(j = 250 \& 350\) times, where \(j\) indicates the number after which the output is almost stable. For each \(s\) and SNR combination we run 20 trails and record the mean values.

The learned dictionaries by each algorithm were compared against the generating dictionary in the similar way as in [Aharon et al. 2006, Kreutz-Delgado et al. 2003b], i.e., within an acceptable squared norm. The average numbers of recovered dictionary atoms for SNR 20 and 35 dB are compared in table 8.4 which show the performance edge of the proposed methods over the other dictionary learning algorithms. Although, RLS produced good results in image data tests but in case of dictionary recovery its performance is very low. One possible reason might be the sparse approximation and the whole dictionary update based on individual training signals that limits the learning improvement for random dictionaries after some initial iterations. Also we notice that ODL and SepDL algorithms were not producing any good results for synthetic data. The improved performance of S1 shows the advantage of enforced sparsity in addition to coefficients value update in the dictionary update stage. In S2, and S3 smoothness is added and produced comparative lower performance than S1 as the generating atoms may not be very smooth. But com-
Comparing to other dictionary learning algorithms the enforced sparsity and smoothness constraints showed the advantage in learning a better basis set. In Fig. 8.1 we plot the average number of recovered dictionary atoms at each iteration of the algorithms for $s = 3$ and 20 dB SNR, similarly Fig. 8.2 shows the plot for $s = 2$ and 30 dB SNR. These results show the improved performance of our proposed methods for dictionary learning applications. A comparison in term of data fitting with learned dictionaries, calculated as $\|X - D_l A\|_F / \|X\|_F$, for SNR = 20 dB and different sparsity level is presented in Fig. 8.3.
### 8.5.2 Image denoising

Image denoising is a well known inverse problem in sparsity base signal processing, where the sparse nature of an image is exploited through dictionary training algorithms. In this test the dictionary learning algorithms are evaluated in terms of noise suppression performance. For this test the training signals set, $X \in \mathbb{R}^{64 \times P}$, where $P = 62001$ (all possible $8 \times 8$ overlapping patches for $256 \times 256$ image or every second patch from every second row for $512 \times 512$ size image) is created from the input noisy image. The image patches are uniformly selected from the input noisy image and each patch is normalized to have zero mean.

The algorithms are compared for four different simulated noise levels, where in each case the original image, $I$ is corrupted with additive white Gaussian noise $N$ of variance $\sigma^2$ corresponds to PSNR decibel values $[16.02, 26.56, 31.30, 40.18]$ dB. The degraded image $O$ is then given by,

$$O = I + N$$

where $O, I, N \in \mathbb{R}^{m \times n}$.

All the compared dictionary learning algorithms were used to learn dictionaries of size $64 \times 100$ from $X$, with $s = 3$ for the sparse coding step. The initial dictionary is generated from the overcomplete DCT transform. The learned dictionaries are then used to find the sparse representation, $x_i$ with $s = 3$, for each patch and the de-noised patches are estimated as $\hat{I}_i = D \cdot x_i$, similar to Elad and Aharon [2006]. The whole
noisy image is then restored using the patch average hypothesis with a weighted sum of noisy patches [Elad and Aharon 2006], given as

\[ \hat{I} = \frac{(\gamma O + \sum_i R_i^T \hat{I}_i)}{(\gamma I + \sum_i R_i^T R_i)} \]

where \( \gamma = 30/\sigma_N \) (\( \sigma_N \) is the standard deviation of noise) and \( R_i \) is a matrix that extracts the \( i^{th} \) patch from the image [Elad and Aharon 2006]. The Improvement-in-Signal-to-Noise-Ratio (ISNR) is used as an objective measure of the quality for the restored images, calculated as

\[ \text{ISNR} = 10 \log_{10} \frac{\| I - O \|^2}{\| I - \hat{I} \|^2} \]

For each PSNR value we have performed 20 trails and computed the average ISNR value over all trails. In table 8.5 we summarized the ISNR values for images of Cameraman and Boat with different PSNR levels. The proposed methods S1, S2 and S3 results in high restoration performance as compared to other methods, with S1 producing the best results. The RLS produce much better results in this experiment as compared to the synthetic data test, this improvement might be due to the inner structure in the learned dictionary and the high sparsity in the data. Also we found that the results of other algorithms falls very near to that of the K-SVD algorithm, except ODL. The ODL which is basically designed for very large data sets, is not a good choice for small or medium training sets. The better performance of S1 reveals the edge of sparsity constraint over the smoothing, as in S2 and S3. The enforced sparsity enhance the sparse coefficients estimation which help with the atom update in the following step. The visual comparison for the input noisy image with PSNR value of 21 dB is presented in Fig. 8.4.

8.5.3 Image in-painting

In the image in-painting task the aim is to fill-in the missing pixel values that are commonly occluded by texts, subtitles, stamps, and publicity or generated by randomly removing some image pixels. Mathematically the image in-painting problem is to reconstruct the underlying complete image (vectorized) \( I \in \mathbb{R}^n \) from its observed incomplete version \( O \in \mathbb{R}^l \), where \( l < n \). We assumed a sparse representation of \( I \) over a dictionary \( D \in \mathbb{R}^{n \times K} \): \( I \approx DA \). The incomplete image \( O \) and the complete image \( I \) are related through \( O = MI \), where \( M \in \mathbb{R}^{l \times n} \) represent the layout of the
§8.5 Experimental results

Figure 8.4: Visual comparison for image restoration.

missing or occluded pixels. Thus, image in-painting problem can be formulated as:

\[ \mathbf{O} = \mathbf{M}\mathbf{I} \]
\[ \approx \mathbf{M}\mathbf{D}\mathbf{x} \approx \Phi \mathbf{x} \]

Assuming a well trained dictionary \( \mathbf{D} \) is available, then the problem boils down to the estimation of sparse coefficients \( \mathbf{x} \) such that the underlying complete image \( \hat{\mathbf{I}} \) is given by \( \hat{\mathbf{I}} = \mathbf{D}\hat{\mathbf{x}} \).

A training data set \( \mathbf{Y} \) is constructed by randomly selecting 32000 patches of size 8 \( \times \) 8 from the set of training images. All the compared methods were used to learn dictionaries of size 64 \( \times \) 100 from \( \mathbf{Y} \), with the sparsity level \( s = 3 \) and overcomplete DCT as initial dictionary. We then select an input image from the set of training images and divided into \( P \) non-overlapping patches of size 8 \( \times \) 8, \( \mathbf{I} \in \mathbb{R}^{64 \times P} \). In each image patch a fraction of \( \tau \) random pixels are deleted, set to zero, where \( \tau \in [0.2, 0.7] \). For each image patch with missing pixels, the sparse coefficients are estimated under the learned dictionaries. The estimated sparse coefficients vector of each patch is denoted by \( a_i \), where \( i \) indicates the number of patch. The reconstructed patch is
Table 8.6: SSD comparison for fill-in missing pixels example.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\tau$</th>
<th>0.3</th>
<th>0.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-SVD</td>
<td>16.27</td>
<td>25.62</td>
<td></td>
</tr>
<tr>
<td>MOD</td>
<td>16.11</td>
<td>30.34</td>
<td></td>
</tr>
<tr>
<td>SGK</td>
<td>17.12</td>
<td>33.02</td>
<td></td>
</tr>
<tr>
<td>MMDL</td>
<td>18.20</td>
<td>32.77</td>
<td></td>
</tr>
<tr>
<td>RLS</td>
<td>15.71</td>
<td>22.32</td>
<td></td>
</tr>
<tr>
<td>ImprDL</td>
<td>16.70</td>
<td>24.87</td>
<td></td>
</tr>
<tr>
<td>ODL</td>
<td>19.12</td>
<td>30.86</td>
<td></td>
</tr>
<tr>
<td>SepDL</td>
<td>31.20</td>
<td>54.37</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>13.50</td>
<td>21.82</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>14.28</td>
<td>22.77</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>15.17</td>
<td>23.80</td>
<td></td>
</tr>
</tbody>
</table>

then obtained as,

$$\hat{i}_i = Da_i$$

where $D$ is the learned dictionary.

In table 8.6 we present the result comparisons in terms of sum of squared difference (SSD), calculated using the reconstructed image and the original input Lena image, for $\tau = 0.3$ and 0.6. As can be seen, the proposed methods, $S_1, S_2$ and $S_3$ produce comparatively higher quality estimation, with $S_1$ having the best performance. A visual comparison for $\tau = 0.7$ is presented in Fig. 8.5, where a corrupted image is reconstructed using the learned dictionaries.

### 8.5.4 Image compression

In the last comparative test the learned dictionaries are compared in the image compression task. Again a training data set $X$ of size $100 \times 9600$ is created by randomly selecting 1200 patches of size $10 \times 10$ from each eight training images. Using KSVD, $S_1, S_2$ and $S_3$ we learned dictionaries of size $100 \times 400$ based on $X$, with OMP for sparse step and same number of iterations. To compress an image using the learned dictionary we divide the image into $P$ non-overlapping patches of size $10 \times 10$ forming a set, $I \in \mathbb{R}^{100 \times P}$. The sparse representation of $I$ using learned dictionary is estimated by OMP, denoted by $A \in \mathbb{R}^{400 \times P}$. The non-zero coefficients of $A$ are quantized using a uniform quantizer with a threshold value. The quantized $A$ is then entropy encoded; the non-zeros coefficients are gathered in one vector and their location information in another vector i.e. the number of zeros preceding each non-zero...
§8.6 Conclusion

The commonly used K-SVD algorithm proposed in [Aharon et al. 2006] for dictionary learning was partly developed by extending the algorithm proposed in [Lesage et al. 2005] where the procrustes rotation used to learn an orthonormal basis was replaced with an SVD used to learn a dictionary atom in the dictionary update stage. By taking a regression point of view on the SVD and observing that the SVD can be seen as an alternate conditional least squares problem all the techniques used to improve least squares estimation can be used in the dictionary update stage to im-

Figure 8.5: Missing Pixel: Visual comparison.

coefficient. Similar to [Skretting and Engan 2011] these two vectors are Huffman coded and formed into a bit sequence. We tested the learned dictionaries for different compression rates (bit rate) and calculated the PSNR in each case, with the peak signal value $p = 255$. The obtained results for Barbara and Lena images are summarized in Table 8.7. For the reference we also includes the PSNR obtained for JPEG-2000 in every case, using the MATLAB built-in functions imread and imwrite. A visual comparison of a small portion from Cameraman image for compression rate 0.60 is presented in Fig. 8.6. These results show comparatively better performance of the proposed methods against K-SVD with faster computations.
Table 8.7: Comparison in terms of PSNR for image compression example.

<table>
<thead>
<tr>
<th>Method</th>
<th>Compression rate</th>
<th>Compression rate</th>
<th>Lena</th>
<th>Barbara</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.50</td>
<td>0.75</td>
<td>1.00</td>
<td>1.25</td>
</tr>
<tr>
<td>JPEG-2000</td>
<td>39.15</td>
<td>41.20</td>
<td>42.21</td>
<td>44.10</td>
</tr>
<tr>
<td>K-SVD</td>
<td>37.10</td>
<td>38.19</td>
<td>40.13</td>
<td>42.45</td>
</tr>
<tr>
<td>S1</td>
<td>38.86</td>
<td>40.38</td>
<td>41.57</td>
<td>43.71</td>
</tr>
<tr>
<td>S2</td>
<td>38.30</td>
<td>39.64</td>
<td>41.03</td>
<td>43.05</td>
</tr>
<tr>
<td>S3</td>
<td>38.47</td>
<td>39.59</td>
<td>41.30</td>
<td>42.89</td>
</tr>
</tbody>
</table>

Figure 8.6: Visual comparison for image compression example
prove the performance of the existing dictionary learning method based on the SVD. Inspired from Shen and Huang [2008] Witten et al. [2009], we have first proposed in this paper a dictionary update stage that also updates the sparsity by $l_1$ penalization of the left singular vector in the matrix rank one approximation. Second based on Huang et al. [2008], we have proposed an approach for regularizing the dictionary atoms in the dictionary update stage and then proposed a dictionary update stage that combine both sparsity update as well as regularized or smooth dictionary atoms. All the experimental results presented showed that the proposed dictionary learning approaches provide better performance than when using a simple SVD for dictionary update. While in this paper we have only proposed three cost functions that could be used in the dictionary update stage, a number of other cost functions can easily be derived. However, the use of a priori information for dictionary learning is recommended in this process. For example, rather than considering only the regularization in the column direction by including a regularization constraint on the dictionary atoms, this regularization constraint could also be included in the matrix row direction Huang et al. [2009]. We could also imagine including a constraint of sparsity on the dictionary atoms to generate sparse dictionary atoms (by sparse dictionary atoms we mean vectors with zeros in their entries) Lee et al. [2010]. A general form of the penalized rank one matrix approximation cost function that can used in the dictionary update stage is given by

$$\{d_k, a^\text{row}_k\} = \arg \min_{d_k, a^\text{row}_k} \|E_k - d_k a^\text{row}_k\|_F^2 + \alpha_1 \|x^\text{row}_k\|_1 + \alpha_2 \|d_k\|_1 + \alpha_3 d_k^\top \Omega_1 d_k + \alpha_4 a^\text{row}_k \Omega_2 a^\text{row}_k^\top \tag{8.20}$$

subject to $\|d_k\|_2 = 1$.

For $\alpha_2 = \alpha_3 = \alpha_4 = 0$ and $\alpha_1 \neq 0$ we obtain the penalized rank one matrix approximation cost function (8.7) used in section 8.3 above to reinforce sparsity in the dictionary update stage. Equation (8.20) offers an efficient framework for introducing a priori information on the data set $X$ in the dictionary learning algorithm. This is for example the case of reshaped fMRI data sets into temporal-spatial matrices in which we are interested to encourage smoothness in the column direction of $X$. For $\alpha_1 = \alpha_2 = \alpha_4 = 0$ and $\alpha_3 \neq 0$, we obtain a cost function for the dictionary update stage adapted to this task by regularizing the dictionary atoms as described in (8.12). For $\alpha_2 = \alpha_4 = 0$ and $\alpha_1, \alpha_3 \neq 0$ we obtain the penalized rank one matrix approximation cost function (8.17) used in section 8.4 above to reinforce sparsity and generate smooth dictionary atoms. A detail derivation for case $\alpha_2 = \alpha_3 = 0$ and $\alpha_1, \alpha_4 \neq 0$ is
given in Appendix B. On the other hand for a data set $X$ for which we are interested in enforcing smoothness in the row direction of $Y$, the associated dictionary update stage can be obtained by alternating minimization of a cost function of the form

$$\{d_k, a_{k}^{\text{row}}\} = \arg \min_{d_k, a_{k}^{\text{row}}} \|E_k^R - d_k a_{k}^{\text{row}}\|_F^2 + \alpha_4 a_{k}^{\text{row}} \Omega_2 a_{k}^{\text{row}^T}$$

subject to $\|d_k\|^2 = 1$. It is derived from (8.12) by taking $\alpha_1 = \alpha_2 = \alpha_3 = 0$. The cost function used in the dictionary update stage to enforce smoothness in both the columns and rows directions of the data matrix $Y$ is written as

$$\{d_k, a_{k}^{\text{row}}\} = \arg \min_{d_k, a_{k}^{\text{row}}} \|E_k^R - d_k a_{k}^{\text{row}}\|_F^2 + \alpha_3 d_k^T \Omega_1 d_k + \alpha_4 a_{k}^{\text{row}} \Omega_2 a_{k}^{\text{row}^T}$$

subject to $\|d_k\|^2 = 1$. It is also derived from (8.20) by taking $\alpha_1 = \alpha_2 = 0$. The results could also be improved by using the adaptive lasso [Zou 2006] to penalize each entry of $a_{k}^{\text{row}}$ differently based on the prior information we have from the sparse coding stage for example. Instead of the penalties used in (8.20) to generate both sparse and smooth singular vector alternative rank one penalized matrix approximations based on the fused lasso [Tibshirani et al. 2005] could also be used. The obtained algorithms for the dictionary update stage can be seen as variants of the power algorithm based on constrained least squares used for computing the SVD. In this paper the results presented in the experimental section are only based on a single iteration, while iterating until convergence could be used, little advantage has been found in doing so the performance improvement in this case will be obtained at the cost of increasing the computational complexity. During the generation of the experimental results it has also been observed that using an SVD of $E_k$ in the dictionary update stage of two (S1, S3) of the three proposed algorithms to generate initial solutions for $d_k$ and $a_{k}^{\text{row}}$ instead of using the resulting vectors obtained from the sparse coding stage also improves the performance. However, this will come at the cost of adding an SVD for the update of each atom of the dictionary.
Classical dictionary learning algorithms ignore structures in data matrices. In [Rubi
binstein et al. 2010] an intermediate method between implicit and explicit dictio-
nary learning methods that can be used for data matrices whose column domain is
structurally smooth was proposed. Using the connection between principal compo-
nent analysis (PCA) and singular value decomposition (SVD) of the data matrix, a
new more efficient sequential dictionary learning method is proposed in this chap-
ter. The proposed technique is adapted for data matrices whose column domain is
structurally smooth. The algorithm generates regularized dictionary atoms via the
solution of a left regularized rank-one matrix approximation problem where reg-
ularization is introduced through sparse basis expansion. Within this framework,
we further extend and propose a new efficient dictionary learning algorithm that
is adapted for data matrices whose row and column domains are both structurally
smooth. This new algorithm is derived by introducing regularization with basis ex-
pansion in both the left and right singular vectors when solving the rank one matrix
approximation problem of the dictionary update stage. Both proposed sequential
algorithms for training dictionaries for structured data are alternating optimization
procedures derived using the connection between alternating least squares and rank
one matrix approximation. Applications on image data illustrating the performance
of the proposed algorithms are provided.

This chapter is organized as: a brief motivation behind the proposed algorithm is
presented in section 9.1 followed by the mathematical model in section 9.2. The pro-
posed methods are outlined in section 9.3 and the simulation evaluation is presented
in section 9.4. As usual the chapter is concluded in section 9.5.
Motivation

In the last two chapters we look at the impact of sparse representation in signal and image processing applications. The strength of sparse representation comes from the fact that it offers representation basis that exploits the sparsity of signals by concentrating most of the signal energy on only few basis coefficients. The notion of signal sparsity is however basis dependent, i.e., a signal may present a better sparseness description in one basis than another. The sparse representation can be achieved by representing the signal either using a pre-defined basis also named implicit dictionaries, such as wavelets [Mallat 1998], contourlets [Do and Vetterli 2005], or overcomplete basis learned from the data, such as KSVD [Aharon et al. 2006], MOD [Engan et al. 1999]. However, due to the limited offer in term of existing basis, the implicit basis may not provide the optimal representation of the signal. On the other hand the overcomplete learned basis, also named as explicit dictionary, can model the signal more sparsely [Tosic and Frossard 2011]. Given a data set $X \in \mathbb{R}^{n \times P}$, dictionary learning algorithms seek a dictionary matrix $D \in \mathbb{R}^{n \times K}$ and a coefficient matrix $A \in \mathbb{R}^{K \times P}$ such that the error $\|X - DA\|_F^2$ is small and $A$ is sparse.

In this chapter we focus on sequential dictionary learning and extend on the algorithm presented in [Aharon et al. 2006]. The proposed method can also be seen as the extended version of the algorithms presented in chapter-5 of this thesis. Particularly, we revisit the algorithm presented in [Rubinstein et al. 2010] but rather than taking the point of view of an intermediate method between implicit dictionaries and explicit dictionaries learned from the data. We look at it as a regularization method obtained through sparse basis expansion in the corresponding dictionary learning algorithm. The regularization is obtained by shrinking the dictionary atom to a certain subspace. This basis can be constructed using a priori information on the data set or on the experiment conducted to generate the data. It is a way to introduce some known structure in the data such as smoothness in the column direction of $X$. Big data sets are structured and classical dictionary learning algorithms ignore these structures in the data resulting in lower performance. This approach suits data set consists of multiple 3D images, for example fMRI data (measured at periodic time during designed experiments or at rest) [Lazar 2008]. The data set is generally reshaped as a single spatio-temporal $n \times P$ type matrix where $n$ is the number of time points and $P$ represent the number of voxels. In fMRI experiments, for example, the goal is to identify the active brain regions, therefore the signal at a fixed voxel (over time) is believed to be smooth and of low frequency [Lazar 2008]. Thus the column domain of the spatio-temporal matrix is structured and dictionary with smoothness
constraint over the atoms can be used to represent such data efficiently. Such controlled dictionary learning approach makes the resulting algorithm more efficient and meanwhile less sensitive to noise.

Given the notion described in Rubinstein et al. [2010] that explicit dictionaries relate better to structured data sets, we propose an alternative sequential dictionary learning algorithm to the one described in Rubinstein et al. [2010] adapted to data matrices whose column domain is structured. We further extend the approach used to derive the proposed alternative algorithm to derive a new sequential dictionary learning algorithm that is adapted for data matrices whose row and column domains are both structured. As it is illustrated in the experimental results, the proposed dictionary learning algorithm accounts for known structure and incorporates regularization in a manner that is more efficient than the algorithm proposed in Rubinstein et al. [2010]. The development of the proposed dictionary learning algorithms are derived based on a variation of the familiar power method or alternating least square method for calculating the SVD Golub and Van Loan [1996]. The steps of the dictionary update stage of the proposed algorithms are obtained through the solution of regularized rank-one matrix approximation problems where regularization is introduced through sparse basis expansion.

9.2 Mathematical Modeling

The sparse representation model presented in previous chapters is briefly presented here again for the smooth follow-up of the presented method, however the reader can skip this repetition and start from section 9.2.1. Given a set of signals \( X = [x_1, x_2, ..., x_P] \in \mathbb{R}^{n \times P} \), a learned dictionary is a collection of vectors or atoms \( d_k \), \( k = 1, ..., K \) that can be used for optimal linear representation. Usually the objective is to find a sparse linear representation for the set of signals \( X \)

\[
\hat{x}_i \simeq Da_i \quad i = 1, ..., P
\]

where \( D = [d_1, d_2, ..., d_K] \), making the total error as small as possible, i.e., minimizing the sum of squared errors. Let the sparse coefficient vectors \( a_i \)'s, constitute the columns of the matrix \( A \), this objective can be stated as the minimization problem

\[
\{D, A\} = \arg \min_{D,A} \|X - DA\|_F^2
\]

where it is imposed a sparsity constraint on \( A \), i.e., \( \sum_{i=1}^{P} ||a_i||_0 \leq Ps \) with \( s \ll K \). Finding the optimal \( s \) corresponds to a problem of model order selection that can be
resolved using a univariate linear model selection criterion [Seghouane and Bekara 2004]. The generally used optimization strategy, not necessarily leading to a global minimum consists in splitting the problem into two stages which are alternately solved within an iterative loop. These two stages are, first, the sparse coding stage, where \( D \) is fixed and the sparse coefficient vectors are found by solving

\[
\hat{a}_i = \arg \min_{a_i} \| x_i - Da_i \|_2^2; \quad \text{subject to} \quad \| a_i \|_0 \leq s_i = 1, \ldots, P
\]  

While there are a number of alternative penalties to the \( \ell_0 \)-norm that encourage sparsity, the simple and popular alternative: the \( \ell_1 \)-norm penalty a direct convex relaxation of the \( \ell_0 \)-norm, is widely adopted. The dictionary update stage where \( A \) is fixed and \( D \) is derived by solving

\[
D = \arg \min_D \| X - DA \|_F^2
\]

constitutes the second stage.

In sequential dictionary learning, which is the focus of this chapter, the minimization (9.2) is separated into \( K \) sequential minimization problems. In the algorithm described in [Aharon et al. 2006], each column \( d_k \) of \( D \) and its corresponding row of coefficients \( a_{k}^{row} \) are updated using a rank-1 matrix approximation of the error matrix obtained from all the signals as follows

\[
\{d_k, a_{k}^{row}\} = \arg \min_{d_k, a_{k}^{row}} \| X - DA \|_F^2
\]

\[
= \arg \min_{d_k, a_{k}^{row}} \left\| \left( X - \sum_{i=1, i \neq k}^K d_i a_{i}^{row} \right) - d_k a_{k}^{row} \right\|_F^2
\]

\[
= \arg \min_{d_k, a_{k}^{row}} \| E_k - d_k a_{k}^{row} \|_F^2.
\]

The SVD of \( E_k = UDV^\top \) can be used to update both \( d_k \) and \( a_{k}^{row} \) by taking \( d_k \) as the first column of \( U \) and \( a_{k}^{row} \) as the first column of \( V \) multiplied by the first diagonal element of \( \Delta \). To avoid the loss of sparsity in \( a_{k}^{row} \) that will be created by the direct application of the SVD on \( E_k \), in [Aharon et al. 2006] it was proposed to modify only the nonzero entries of \( a_{k}^{row} \) resulting from the sparse coding stage. This is achieved by taking into account only the signals \( x_i \) that use the atom \( d_k \) in (9.3) or by taking the SVD of \( E_k^R = E_k I_{w_k} \), where \( w_k = \{ i | 1 \leq i \leq P; a_{k}^{row}(i) \neq 0 \} \) and \( I_{w_k} \) the \( P \times |w_k| \) submatrix of the \( P \times P \) identity matrix obtained by retaining only those columns whose index numbers are in \( w_k \), instead of the SVD of \( E_k \).
In practice, to avoid the use of the computationally expensive SVD, especially when the number of signals is large, an approximate solution obtained by applying a single iteration of the power method for computing the SVD can be used to reduce the computational cost of the dictionary update stage. Recall that the power method, or alternating least square method, sequentially estimates \( d_k \) with \( a_k^{rowR} = a_k^{row}I_{w_k} \) fixed and vice versa by solving the least square problem

\[
\|E_k^R - d_k a_k^{rowR}\|_F^2 = \text{tr}\left\{ (E_k^R - d_k a_k^{rowR}) (E_k^R - d_k a_k^{rowR})^T \right\} = \|E_k^R\|_F^2 - 2d_k^T E_k^R a_k^{rowR}^T + \|d_k\|^2 \|a_k^{rowR}\|^2 \tag{9.4}
\]

and then rescaling the estimates to give

\[
d_k = \frac{E_k^R a_k^{rowR}^T}{\|E_k^R a_k^{rowR}^T\|_2} \tag{9.5}
\]

\[
a_k^{rowR} = d_k^T E_k^R \tag{9.6}
\]

A simplified dictionary update stage is then obtained by applying a single iteration of (9.5) and (9.6) instead of the SVD of \( E_k^R \).

### 9.2.1 Double Sparsity

Rather than learning the complete dictionary using a variant of (9.2) as in (9.3) or (9.5) and (9.6) (explicit dictionary) or using a completely known dictionary (implicit dictionary), a compromise between the two may be more appropriate. This is particularly of interest when a priori information about the dictionary to account for known structure in the data \( X \) is available. The sparse dictionary model expressed as

\[
D = D_pZ \tag{9.7}
\]

where \( D_p \in \mathbb{R}^{n \times p} \) is a pre-specified dictionary and \( Z \in \mathbb{R}^{p \times K} \) is the atom representation matrix, was proposed in Rubinstein et al. [2010] as a simple and effective approach to derive this compromise. Within this dictionary model each atom \( d_k \) of \( D \) is itself a sparse linear combination of a set of known vectors defined by \( D_p \). As in Rubinstein et al. [2010], we focus on matrices \( Z \) having a fixed number of nonzeros per column, i.e., \( \|z_k\|_0 \leq t, k = 1, ..., K \), for some \( t \). Learning \( D \) is obtained by
learning $Z$ in the dictionary update stage by solving
\[ Z = \arg \min_Z \| X - D_p Z A \|_F^2 \]  
\[ \text{s. t. } z_k \|_0 \leq t \text{ and } \| D_p z_k \|_2^2 = 1; \forall k = 1, ..., K \]  
(9.8)

Under this modification, the change incurred to (9.4) in the sequential atom update is given by
\[ \{ z_k, a_{k,R}^{\text{row}} \} = \arg \min_{z_k, a_{k,R}^{\text{row}}} \| E_k^R - D_p z_k a_{k,R}^{\text{row}} \|_F^2 \]  
\[ \text{s. t. } z_k \|_0 \leq t \text{ and } \| D_p z_k \|_2^2 = 1; \forall k = 1, ..., K \]  
(9.9)

We named this dictionary learning approach in the introduction as controlled dictionary learning because (9.9) control the dictionary update stage by restricting it to find the best rank one approximation within a certain subspace.

The problem (9.9) was solved in Rubinstein et al. [2010] using a variant of the power method described above in (9.5) and (9.6) by first optimizing over $a_{k,R}^{\text{row}}$ for fixed $z_k$
\[ a_k^{R,R^\top} = \frac{E_k^R D_p z_k}{\| D_p z_k \|_2^2} \]  
(9.10)

and then optimizing over $z_k$ for fixed $a_k^{\text{row}}$
\[ \{ z_k \} = \arg \min_{z_k} \| E_k^R - D_p z_k a_k^{\text{row}R} \|_F^2 \text{ s. t. } z_i \|_0 \leq t \]  
(9.11)

using the equivalent expression
\[ \{ z_k \} = \arg \min_{z_k} \| E_k^R a_k^{\text{row}R^\top} - D_p z_k \|_F^2 \text{ s. t. } z_k \|_0 \leq t. \]  
(9.12)

Similar to the sparse coding stage (9.1), the $\ell_1$-norm penalty is used as an alternative to the $\ell_0$-norm to solve (9.12). Based on the observation that the optimization (9.9) can be resolved by solving a constrained least square problem where the regularization penalty is introduced to the minimization problem to promote sparsity in $z_k$ a new alternative variant of the power method for training a sparse dictionaries is derived in the next section.
9.3 Proposed Algorithms

9.3.1 Sequential learning of sparse dictionaries

Instead of using the equivalent expression (9.12), our proposed alternative variant of the power method to be used for the sequential atom update in the case of training a sparse dictionary is obtained by directly minimizing the criterion (9.9) which can be rewritten as

\[
\|E_k^R - D_p z_k a_k^{rowR}\|_F^2 = \text{tr}\{ (E_k^R - D_p z_k a_k^{rowR})(E_k^R - D_p z_k a_k^{rowR})^\top \}
\]

\[
= \text{tr}\{ E_k^R E_k^R \} - 2 \text{tr}\{ E_k^R a_k^{rowR} z_k^\top D_p \}
+ \text{tr}\{ a_k^{rowR} z_k^\top D_p D_p z_k a_k^{rowR} \}
\]

\[
= \text{tr}\{ E_k^R E_k^R \} - 2 z_k^\top D_p E_k^R a_k^{rowR}
+ a_k^{rowR} z_k^\top D_p D_p z_k
\]

(9.13)

subject to \( \| z_k \|_0 \leq t \) and \( \| D_p z_k \|_2^2 = 1 \); with respect to \( z_k \) and \( a_k^{rowR} \). Minimizing this criterion with respect to \( a_k^{rowR} \) for fixed \( z_k \) gives

\[
a_k^{rowR} = \frac{z_k^\top D_p E_k^R}{\| D_p z_k \|_2^2}
\]

(9.14)

for updating the nonzero elements of \( a_k^{rowR} \) during the dictionary update stage. By substituting this back into (9.13) and after some algebra, we obtain

\[
\{ z_k \} = \arg \min_{z_k} \text{tr}\{ E_k^R E_k^R \} - z_k^\top D_p E_k^R E_k^R D_p z_k
\]

subject to \( \| D_p z_k \|_2^2 = 1 \) and \( \| z_k \|_0 \leq t \). Or computed using

\[
\{ z_k \} = \arg \max_{z_k} z_k^\top D_p E_k^R E_k^R D_p z_k
\]

subject to \( \| D_p z_k \|_2^2 = 1 \) and \( \| z_k \|_0 \leq t \).

Written this way problem (9.9) resemble the problem of sparse PCA Jolliffe et al. [2003] where the data matrix consists of the projected data \( E_k^R D_p \). The difference is in the form of the constraints imposed; i.e.; the normalization is not imposed on \( z_k \) but on \( D_p z_k \) and the orthogonality constraint with subsequent vectors is not needed since we are interested in the rank one approximation. Recall that the SVD problem
can also be written as finding the linear combination of variables maximizing the sample variance such that these linear combinations are orthonormal. The derivation of the proposed approach for sequential atom update in controlled dictionary learning is based on the connection of PCA with the SVD of $E_k^T D_p$ and $z_k$ is extracted through solving a penalized rank one matrix approximation problem [Witten et al. 2009; Shen and Huang 2008]. Regularization penalty is introduced in the corresponding minimization problem to promote sparsity in $z_k$. The key to this approach is the observation that the optimization problem

$$
\text{arg min}_{z_k, a_k^{row}} \| E_k^T D_p - a_k^{rowR^T} z_k \|_F^2
$$

is connected to least squares regressions. To achieve sparseness on $z_k$ we employ some regularization in (9.17) that promotes shrinkage and sparsity of the entries of $z_k$

$$
\text{arg min}_{z_k, a_k^{row}} \| E_k^T D_p - a_k^{rowR^T} z_k \|_F^2 + \lambda |z_k|
$$

where $|z_k| = \sum_{i=1}^p |z_{ki}|$. For fixed $a_k^{rowR}$, minimization of (9.18) with respect to $z_k$ is equivalent to the minimization of

$$
\left\| q - \left( I_p \otimes a_k^{rowR^T} \right) z_k \right\|_F^2 + \lambda |z_k|
$$

where $q = (a_1^T, ..., a_p^T)^T \in \mathbb{R}^{P_q \times p}$, with $a_i$ being the $i^{th}$ column of $E_k^T D_p$ and $P_q = \text{card}(w_k)$. The minimizer $z_k$ of (9.19) is given by $^1$

$$
z_k = \frac{1}{\|a_k^{rowR}\|_2^2} \text{sgn} \left( D_p E_k a_k^{rowR^T} \right) \left( |D_p E_k a_k^{rowR^T}| - \frac{\lambda}{2} I_p \right)_+ \tag{9.20}
$$

and imposing the scaling constraint $\| D_p z_k \|_2^2 = 1$ gives

$$
z_k = \frac{z_k}{\| D_p z_k \|_2}
$$

Using this constraint the minimizer to $a_k^{row}$ given in (9.14) becomes

$$
a_k^{row} = z_k^T D_p E_k \tag{9.21}
$$

Note that, in comparison to the algorithm proposed in [Rubinstein et al. 2010], $z_k$ is obtained by the closed form solution (9.20) while in [Rubinstein et al. 2010] it is obtained by the approximate solution of (9.12) implemented using a pursuit algorithm.

$^1$where $(x)_+$ represent "the positive part of", i.e., $(x)_+ = x$, if $x > 0$, and $(x)_+ = 0$, if $x \leq 0$ and $\text{sgn}()$ is the sign operator.
Based on the above set of equations, the sequential dictionary update stage consists of the following steps: 1) apply an SVD to $E_k^R D_p$ and use the first left singular vector as $a_k^{rowR}$, 2) generate $z_k$ using (9.20), 3) normalize $z_k$ such that $\| D_p z_k \|_2^2 = 1$ and 4) update $a_k^{rowR}$ using (9.21). These steps may be repeated until convergence, though we have found little advantage in repeating them more than one time. Instead of using the SVD in step 1), the $a_k^{rowR}$ obtained directly from the sparse coding stage (9.1) could also be used. Setting $\lambda = 0$ in (9.20), step 2) reduces to $z_k = \frac{D_p E_k^R a_k^{rowR}}{\| a_k^{rowR} \|_2^2}$ and corresponds to the alternating least-squares solution of (9.17) when calculating the SVD of $E_k^R D_p$. The complete proposed dictionary learning algorithm is detailed in table-9.1. The proposed controlled dictionary learning approach acts as a regularizer and generates smooth dictionary atoms. Besides the appearance of a pre-specified dictionary $D_p$ in the dictionary update cost function, the proposed approach for regularized dictionary learning is different from a regularization dictionary learning algorithm obtained with a dictionary update based on the cost function

$$\| E_k^R - d_k a_k^{rowR} \|_F^2 + \alpha d_k^T \Omega d_k$$

(9.22)

where $\Omega$ is a non-negative definite roughness penalty matrix used to penalize the second differences [Ramsay and Silverman 2005]

$$d_k^T \Omega d_k = d_k^2(1) + d_k^2(n) + \sum_{i=2}^{n-1} (d_k(i+1) - 2d_k(i) + d_k(i-1))^2$$

(9.23)

and $\alpha$ is the hyperparameter that control the amount of regularization. In the case of (9.22) the term $\alpha d_k^T \Omega d_k$ is used to impose some smoothness on the atoms $d_k$'s. However, it is well known that enforcing smoothness using penalties does not work well for all functions. It is the case for functions with spikes and sharp jumps. Wavelets regularization for example performs well for this kind of function by concentrating on sparness in the transformed domain rather than on regularization in the original domain. Furthermore, an efficient algorithm exits for computing the discrete wavelet transform [Mallat 1998]. Therefore, if in the transformed domain obtained by the basis $D_p$ in the case of controlled dictionary learning, the atoms are sparse, then we should take advantage of this information and work in this transformed domain [Johnstone and Lu 2009]. Direct regularization as offered by (9.22) should work well in situations when the atom function doesn’t contain abrupt or sudden variations or when the atom function is “a nice function”.
Table 9.1: Double sparsity controlled dictionary learning algorithm

<table>
<thead>
<tr>
<th>Algorithm 1</th>
</tr>
</thead>
</table>

**Given:** Training data $X \in \mathbb{R}^{n \times P}$, pre-specified dictionary $D_p$, initial dictionary representation $Z_{ini}$, sparsity parameters $\lambda$ and $s$, no. of iterations $j$.

**Dictionary initialization:**

$$D = D_pZ_{ini}$$

**For** $i=1$ to $j$

1. **Sparse Coding Stage:**

   Find sparse coefficients $A$, using
   
   $\hat{a}_i = \arg \min_{a_i} \| x_j - D a_i \|^2$;
   subject to $\| a_i \|_0 \leq s \quad i = 1, ..., P$

2. **Dictionary Update Stage:**

   For each column $k = 1, 2, ..., K$ in $Z$,
   
   a. Set the $k^{th}$ column of $Z$, $z_k = 0$;
   b. Compute the error matrix using $E_{k} = Y - D_p Z A$
   c. Using $w_{k}$ the set of indices in $A$ that uses $z_k$ generate $E_{k}^R = E_{k}I_{w_{k}}$
   d. Generate initials vectors $z_k$ and the associated row in $A$, $a_{k}^{rowR}$ using an SVD of $E_{k}^{R} D_p$
   e. Set $z_k$ as the first right singular vector multiplied by the associated singular value and $a_{k}^{rowR}$ as the first left singular vector
   f. Update the vector $z_k$ and its sparsity using
   
   $$z_k = \frac{1}{\| a_{k}^{rowR} \|_2} \text{sgn} \left( D_p^T E_k^R a_{k}^{rowR} \right) \left( D_p^T E_k^R a_{k}^{rowR} - \frac{\lambda}{2} I_p \right)$$
   g. Rescale $z_k$ using $z_k = \frac{z_k}{\| D_p z_k \|_2}$
   h. Update the corresponding row in $A$, $a_{k}^{rowR}$ using
   
   $$a_{k}^{rowR} = z_k^T D_p^T E_k^R$$

**Output:** $D, A$
9.3.2 Sequential dictionary learning with smooth constraint

With a number of data sets \( X \) we may not only be interested in obtaining smooth dictionary atoms to encourage smoothness in the column direction of the data set \( X \) but may also be interested in enforcing smoothness in the row direction of \( X \) as well. Instead of the widely used \( \ell_2 \) penalty (9.23) and since the focus of this chapter is regularization via basis expansion [Creltin 1996] this later approach is adopted in the proposed cost function to enforce smoothness in both the column and row directions of the data set \( X \). Using these prerequisites, the dictionary update stage problem (9.4) is written as

\[
\|E_k^R - d_k a_k^{row} \|_F^2 \quad \text{subject to } d_k = D_p z_k, \quad \| z_k \|_0 \leq t, \\
\| D_p z_k \|_2^2 = 1, \quad \text{and } a_k^{row} = Bc_k
\]

(9.24)

where \( B \) is another predefined basis and the sequential atom update cost function is given by

\[
\{ z_k, c_k \} = \arg \min_{z_k, c_k} \| E_k^R - D_p z_k c_k^T B^T \|_F^2 \\
\text{subject to } \| z_k \|_0 \leq t, \quad \text{and } \| D_p z_k \|_2^2 = 1, \quad k = 1, \ldots, K
\]

(9.25)

Regularization in the row direction is obtained by representing \( a_k^{row} \) to a fully known low-dimensional subspace \( S_{a_k}^{row} = \text{span}(b_1, \ldots, b_l) \) where \( b_i \) is the \( i \)th column of \( B \). Description in a lower dimensional subspace can be seen as a limiting case of regularization by using the squared \( \ell_2 \) norm between \( a_k^{row} \) and its orthogonal projection \( P_B a_k^{row} \), i.e.; \( \alpha \| (I - P_B) a_k^{row} \|_2^2 \) as a penalty. In this case \( P_B \) is the orthogonal projector on \( S_{a_k}^{row} \). The associated penalty matrix is the residual operator \( \Omega = \alpha (I - P_B) \) where \( \alpha \to \infty \) [Huang et al. 2009]. The shrinkage of the solution toward the lower dimensional subspace \( S_{a_k}^{row} \) can be obtained by using finite values for the parameter \( \alpha \).

The variant of the power method to be used for the sequential atom update in this case computes \( z_k \) and \( c_k \) by directly minimizing

\[
\| E_k^R - D_p z_k c_k^T B^T \|_F^2 = \text{tr}\{ (E_k^R - D_p z_k c_k^T B^T) \} \\
= \{ E_k^R - D_p z_k c_k^T B^T \}^T \\
= \text{tr}\{ E_k^R E_k^R^\top \} - 2 z_k^\top D_p E_k^R Bc_k \\
+ \| D_p z_k \|_2^2 \| Bc_k \|_2^2
\]

(9.26)
subject to $\| z_k \|_0 \leq t$ and $\| D_p z_k \|_2^2 = 1$; with respect to $z_k$ and $c_k$. Minimizing this criterion with respect to $c_k$ for fixed $z_k$ gives

$$c_k^\top = \frac{z_k^\top D_p^\top E_k^g B \left( B^\top B \right)^{-1} \left( B^\top B \right)}{\| D_p z_k \|_2^2}$$  \hspace{1cm} (9.27)

for updating $c_k$ during the dictionary update stage. By substituting this back into (9.26) and after some algebra, we obtain

$$\{ z_k \} = \arg \max_{z_k} z_k^\top D_p^\top E_k^g B \left( B^\top B \right)^{-1} \left( B^\top E_k^g D_p \right)$$  \hspace{1cm} (9.28)

subject to $\| D_p z_k \|_2^2 = 1$ and $\| z_k \|_0 \leq t$. Similar to (9.16), this is a problem of sparse PCA [Jolliffe et al. 2003] where the data matrix is defined by $Q = \left( R^{-1} \right)^\top B^\top E_k^g D_p$ with $B^\top B = R^\top R$ being the Cholesky decomposition. Using the connection between PCA and SVD [Witten et al. 2009; Shen and Huang 2008], $z_k$ can be computed through solving the penalized rank one matrix approximation problem

$$\arg \min_{z_k, c_k} \| Q - c_k z_k^\top \|_F^2 + \lambda \| z_k \|$$  \hspace{1cm} (9.29)

where $\| z_k \| = \sum_{i=1}^p |z_{ki}|$. For fixed $c_k$ the minimizer of (9.29) with respect to $z_k$ is given by

$$z_k = \frac{1}{\| c_k \|_2} \sgn \left( Q^\top c_k \right) \left( \left| Q^\top c_k \right| - \frac{\lambda}{2} I_p \right)_+$$  \hspace{1cm} (9.30)

with the constraint $\| D_p z_k \|_2^2 = 1$, we have

$$z_k = \frac{z_k}{\| D_p z_k \|_2}.$$  

Using this constraint the minimizer to $c_k$ given in (9.27) becomes

$$c_k^\top = z_k^\top D_p^\top E_k^g B \left( B^\top B \right)^{-1}$$  \hspace{1cm} (9.31)

and both $z_k$ and $c_k$ are obtained using closed form solutions.

Based on the above set of equations, the sequential dictionary update stage consists of the following steps: 1) apply an SVD to $Q$ and use the first singular vector as $c_k$, 2) generate $z_k$ using (9.30), 3) normalize $z_k$ such that $\| D_p z_k \|_2^2 = 1$ and 4) update $c_k$ using (9.31). Setting $\lambda = 0$ in (9.30), step 2) reduces to $z_k = \frac{Q c_k}{\| c_k \|_2}$ and corresponds to the alternating least-squares solution when calculating the SVD of $Q$. The complete proposed dictionary learning algorithm is described in table-9.2. For an orthonormal
basis $B$, so that the matrix satisfies $B^\top B = I$, the problem (9.26) boils down to (9.13) on the projected data $E_k^R B$. This is because

$$
\|E_k^R - D_p z_k c_k^\top B^\top \|_F^2 = \| E_k^R \|_F^2 - 2z_k^\top D_p E_k^R B c_k + \| D_p z_k \|_2^2 \| c_k \|_2^2
$$

which, modulo the irrelevant constant $\| E_k^R \|_F^2$, is the same as (9.13) with $E_k^R$ replaced by $E_k^R B$.

### 9.4 Experimental results

The performance of the resulting controlled dictionary learning algorithms derived using the proposed dictionary update stages are compared with the algorithms described in [Aharon et al. 2006] and [Rubinstein et al. 2010] on two different image processing experiments. The first experiment consists of image in-painting whereas the second image processing experiment corresponds to the problem image denoising. In all experiments the initial dictionary is generated by the DCT transform and orthogonal matching pursuit (OMP) [Davis et al. 1997] is employed to estimate the sparse coefficients with $s = 3$ coefficients approximation for each signal. The simulations are performed using a training set of some well known gray scale images\(^2\) of size $256 \times 256$ or $512 \times 512$. The training signals set is constructed using uniformly selected image patches of size $8 \times 8$ pixels either from the input image or from the set of training images. The patches are regularly sampled in an overlapping fashion and each patch is normalized to have zero mean. All the comparison results are generated under the same software and hardware environment. For the algorithms [Aharon et al. 2006] and [Rubinstein et al. 2010] and named below K-SVD and K-SVDS respectively, we used our own implementations.

#### 9.4.1 Dictionary training

In all our experiments the first step is to learn a dictionary using the proposed learning algorithms, algorithm 1 (Alog-1), algorithm 2 (Algo-2), K-SVD and the K-SVDS. A training signals set $Y \in \mathbb{R}^{64 \times 2000}$ is created by randomly selecting 2000 patches of size $8 \times 8$ from the set of training images used in the simulations. We then seek an over-complete dictionary, $D_t \in \mathbb{R}^{64 \times 100}$, using K-SVD, K-SVDS, Algo-1, and Algo-2, that approximates the columns of $X$ in terms of sparse coefficients vectors $A$ generated using OMP. Each algorithm is iterated 5 times to train the dictionary by alternating between sparse coding and dictionary update steps. The learned dictionary $D_t$ is

\(^2\)Lena, Barbara, Cameraman, Boat, House, Fingerprints, Peppers, Flinstones
Table 9.2: The double sparsity algorithm with a smooth constraint

**Algorithm 2**

**Given:** Training data $X \in \mathbb{R}^{n \times P}$, pre-specified dictionary $D_p$, initial dictionary representation $Z_{ini}$, sparsity parameters $\lambda$ and $s$, no. of iterations $j$ and an approach to build $B$.

**Dictionary initialization:**

$$D = D_p Z_{ini}$$

**For** $i=1$ to $j$

1. **Sparse Coding Stage:**

   Find sparse coefficients $A_i$ using
   $$\hat{a}_i = \arg \min_{a_i} \| x_i - D a_i \|_2^2;$$
   subject to $\| a_i \|_0 \leq s$ $i = 1, ..., P$

2. **Dictionary Update Stage:**

   For each column $k = 1, 2, ..., K$ in $Z$
   Set the $k^{th}$ column of $Z$, $z_k = 0$;
   2.a: Compute the error matrix using
   $$E_k = X - D_p Z A$$
   2.b: Using $w_k$ the set of indices in $A$ that uses $z_k$ generate
   $$E_k^R = E_k I_{w_k}$$
   2.c: Generate the basis $B$ of size $P_q \times l$,
   2.d: Apply a Cholesky decomposition on $B^\top B = R^\top R$,
   2.e: Generate the matrix $Q = (R^{-1})^\top B^\top E_k^R D_p$,
   2.f: Generate initials vectors $z_k$ and $c_k$ using an SVD of $Q$
   2.g: Set $z_k$ as the first right singular vector multiplied
       by the associated singular value and $c_k$ as the
       first left singular vector
   2.g: Update the vector $z_k$ and its sparsity using:
   $$z_k = \text{sgn} \left( Q^\top c_k \right) \left( |Q^\top c_k| - \frac{1}{2} \lambda \right)^+$$
   2.i: Rescale $z_k$ using:
   $$z_k = \frac{z_k}{\|D_p z_k\|_2}$$
   2.j: Update the corresponding row in $A$, $a_{k}^{row}$ using:
   $$a_{k}^{rowR} = a_{k}^\top D_p E_k^R B \left( B^\top B \right)^{-1} B^\top$$

**end.**

**Output:** $D, A$
then used in the image restoration and reconstruction steps in the similar fashion as outlined in [Elad and Aharon] [2006] Aharon et al. [2006].

In image denoising experiments when the dictionary is trained on the patches from the noisy image, the training signal set $X$ contains 62001 patches (all possible $8 \times 8$ overlapping patches for $256 \times 256$ image or every second patch from every second row for $512 \times 512$ size image).

### 9.4.2 Image in-painting

In image in-painting task the aim is to fill-in the missing pixels values that are mainly occluded by texts, subtitles, stamps, and publicity or generated by randomly removing some image pixels. Mathematically the image in-painting problem is to reconstruct the underlying complete image (vectorized) $X \in \mathbb{R}^n$ from its observed incomplete version $Y \in \mathbb{R}^l$, where $l < n$. We assumed a sparse representation of $X$ over a dictionary $D \in \mathbb{R}^{n \times K}$: $X \approx DA$, $\|a_i\|_0 = s$ with $s << n$ and $K \geq n$. The incomplete image $Y$ and the complete image $X$ are related through $Y = MX$, where $M \in \mathbb{R}^{l \times n}$ represent the layout of the missing or occluded pixels. Thus, image in-painting problem can be formulated as:

$$Y = MX$$
$$= MDA \approx FA$$

Assuming a well trained dictionary $D$ is available, then the problem boils down to the estimation of sparse coefficients $\hat{A}$ such that the estimate of underlying complete image $\hat{X}$ can be estimated as $\hat{X} = D\hat{A}$.

The first comparison is shown for the widely used image in-painting example of text removal. We select an input image from the training images set and impose a mask $M$ of some text, i.e write some text on the image. The corrupt image is then divided into $P$ non-overlapping patches of size $8 \times 8$; vectorized each patch to form the matrix $Y \in \mathbb{R}^{64 \times P}$. For each column vector in $Y$ we estimated the sparse coefficients under the learned dictionary $D_i$ using the OMP. The estimated sparse coefficients vector for each patch is denoted by $a_i$, where $i$ indicates the number of patch. The reconstructed image patch is then obtained as, $\hat{X}_i = D_i\hat{a}_i$.

A visual comparison of the reconstructed image is presented in Fig 9.1 along with the a zoom-in region in the top-right corner of the image to reveal the performance clearly. It can be observed that dictionaries of Algo-1 and Algo-2 produced better
Table 9.3: Scratches and text removal comparison in terms of SSD.

<table>
<thead>
<tr>
<th>Method</th>
<th>Text</th>
<th>Scratches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lena</td>
<td>Boat</td>
</tr>
<tr>
<td>K-SVD</td>
<td>35.08</td>
<td>32.43</td>
</tr>
<tr>
<td>K-SVDs</td>
<td>42.47</td>
<td>37.22</td>
</tr>
<tr>
<td>Algo.1</td>
<td>27.32</td>
<td>28.18</td>
</tr>
<tr>
<td>Algo.2</td>
<td>34.25</td>
<td>31.19</td>
</tr>
</tbody>
</table>

Figure 9.1: Visual comparison of text removal.

reconstruction results compared to K-SVDs and K-SVD, with Algo.1 having the best results. One possible reason for the better performance of Algo.1 & 2 is the small coherence among the learned dictionary atoms as compared to K-SVD and K-SVDs. Another visual comparison for a mask $\mathbf{M}$ of some random scratches is shown in Fig. 9.2, a zoom-in portion in the top-right corner is presented for a detailed inspection. In table-9.3 we present a numerical comparisons in terms of sum of squared difference (SSD), calculated using the reconstructed image and original input image for text and scratches (lines) occluded images. In the next example we show a more challenging example of fill-in the missing image pixels to evaluate the performance of dictionary learning. In this experiment we select an input image from the training set and divided it into $P$ non-overlapping patches of size $8 \times 8$ to form the matrix $\mathbf{Y} \in \mathbb{R}^{64 \times P}$. 
In each image patch a fraction of \( m \) random pixels are deleted, set to zero, where \( m \in [0.2, 0.7] \). For each image patch with missing pixels, the sparse coefficients are estimated under the learned dictionaries, \( D_i \) using OMP. The estimated sparse coefficients vector of each patch is denoted by \( a_i \), where \( i \) indicates the number of patch. The reconstructed patch is then obtained in similar way as, \( \hat{X}_i = Da_i \). A visual comparison for \( m = 0.5 \) is presented in Fig. 9.3, where a corrupted image is reconstructed using the learned dictionaries. The comparisons in terms of SSD, calculated using the reconstructed image and original input Lena image, for \( m = 0.3 \) and 0.6 are presented in table 9.4. As can be seen, the proposed methods, Algo.1 & 2 produce comparatively higher quality estimation than K-SVDs. In addition, the proposed algorithms are computationally less expensive than K-SVDs, as the dictionary in K-SVDs is learned using sequential update which requires \( K \) numbers of SVD iterations.

### 9.4.3 Image denoising

In our second test the learned dictionaries are compared in image denoising task. The comparisons are presented for four different simulated noise levels, where in each case the original image, \( X \) is corrupted with additive white Gaussian noise, \( N \) of variance, \( \sigma_N^2 \), corresponds to PSNR decibel values \([15.10, 26.56, 31.30, 36.68] \) dB.
Table 9.4: Double sparsity: Fill-in missing pixels comparison in terms of SSD.

| Method | m | SSD  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.3</td>
<td>0.6</td>
</tr>
<tr>
<td>K-SVD</td>
<td>26.77</td>
<td>40.70</td>
</tr>
<tr>
<td>K-SVDs</td>
<td>43.01</td>
<td>54.68</td>
</tr>
<tr>
<td>Algo-1</td>
<td>20.26</td>
<td>33.10</td>
</tr>
<tr>
<td>Algo-2</td>
<td>27.89</td>
<td>39.67</td>
</tr>
</tbody>
</table>

Figure 9.3: Visual comparisons for Fill-in missing image pixels.
The degraded image $Y$ is then given by,

$$Y = X + N$$

where $Y, X, N \in \mathbb{R}^{m \times n}$.

A data training set is created using all possible $8 \times 8$ overlapping patches from the noisy image $Y$. We used K-SVD, K-SVDS, Algo-1 and Algo-2 to learn dictionaries of size $64 \times 100$ from $X$, using OMP for the sparse step. The learned dictionaries are then used to find the sparse representation, $a_i$, for each patch and the de-noised patches are estimated as $\hat{X}_i = DA_i$. The whole noisy image is restored using the patch average hypothesis with a weighted sum of noisy patch [Aharon et al. 2006], given as

$$\hat{X} = \frac{(\alpha Y + \sum_i R_i^T \hat{X}_i)}{(\alpha 1 + \sum_i R_i^T R_i)}$$

where $\alpha$ is a positive number and $R_i$ is the matrix used to extract a patch at $i^{th}$ location. The Improvement-in-Signal-to-Noise-Ratio (ISNR) is used as an objective measure of the quality for the restored images, calculated as

$$\text{ISNR} = 10 \log_{10} \frac{\|X - Y\|^2}{\|X - \hat{X}\|^2}$$

For each PSNR value we generated 20 results and select the average of ISNR values. In Table 9.5 we summarized the ISNR values for images of Cameraman and Peppers with different PSNR levels. It can be observed that our proposed methods Algo-1 & 2 produced comparative better results than K-SVDS. Although our proposed method Algo-1 & 2 could not beat K-SVD in noise removal but produced very close results. The visual comparison for the input noisy image with PSNR value of 20 dB is presented in Fig. 9.4.
9.5 Conclusion

Big data sets arising for example from spatio-temporal measurements as in fMRI studies can be structurally smooth. In this case the data set reshaped as a spatio-temporal matrix is structured in the column domain and classical dictionary learning algorithms such as Engan et al. [1999] Aharon et al. [2006] ignoring this structure in the data matrix will result in lower performance. While we did focus on smoothness in this chapter other forms of structure such as sparsity or positivity could also be present in the data. In Rubinstein et al. [2010] a dictionary learning algorithm that can handle such data set was proposed but instead it was presented as a new dictionary structure bridging the gap between implicit and explicit dictionaries. Taking a regularized rank-one matrix approximation approach via sparse basis expansion in the dictionary update stage an alternative sequential dictionary learning method to the one presented in Rubinstein et al. [2010] adapted for data matrices whose column domain is structurally smooth was proposed in this chapter. The obtained algorithm for the dictionary update stage can be seen as a variant of the power method or alternating least square method for computing the SVD in which the steps are derived based on the connection between PCA and SVD of the data matrix. All the presented experimental results obtained using applications on image data showed that the proposed controlled dictionary learning algorithm outperform the method.
presented in Rubinstein et al. [2010]. This algorithm is thus a compelling alternative to Rubinstein et al. [2010]. The potential performance of the algorithm could be improved by introducing extra-tuning through the adaptive lasso Zou [2006] in (9.18) to penalize each entry of $a_k$ differently. Within this framework of regularization via basis expansion, we further proposed a new sequential dictionary learning algorithm that is adapted for data matrices whose column and row domains are both structured. The steps of the dictionary update stage of this algorithm were derived by introducing regularization via basis expansion in both directions when solving the rank one matrix approximation problem through alternating least square. All the experimental results obtained on applications with image data also showed that this algorithm provided performance improvement compared to Rubinstein et al. [2010].
Chapter 10

Conclusion

"This is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning...."
— W. Churchill

10.1 Summary

Image restoration has been a well studied linear inverse problem in image processing. In modern imaging systems the manufacturers continue to pack a large numbers of pixels per unit area, which increase the noise sensitivity in image data. The main focus of this thesis was to develop robust and optimized image restoration techniques, mainly degraded by blur and noise artifacts. We addressed this problem with image deconvolution and sparse representation based approaches. The thesis is divided into two parts: i) image deconvolution based techniques and ii) sparse representation based methods. The main motivation and objectives are outlined in Chapter 1, along with the mathematical formulation of general image restoration problem.

The first part starts from Chapter 2, where the general concept of image deconvolution and a brief review of its related literature is presented. This part consists of a total three chapters, from Chapter 3 to 5, and mainly concern with image deblurring. A hybrid Fourier-wavelet technique, based on expectation maximization (EM) is presented in Chapter 3 for non-blind image deblurring. In this method the main objective function is divided into a Fourier domain deconvolution and wavelet domain denoising, where the Gaussian scale mixture (GSM) is used to model the wavelet coefficients. This decomposition avoids the computational burden of handling blurring operator in wavelet domain and results in two step image restoration. The methods presented in Chapter 4 and 5 are designed for the more challenging problem of blind image deconvolution (BID), where either no or very little information about
original image and blurring operator is given. A unifying non-negative sparse matrix approximation (SNMA) for BID is outlined in Chapter 4, to iteratively estimate the blurring kernel and sharp image. Unlike many BID methods, this method is parameter free and does not requires any prior statistics. Compared to other SNMA methods, the proposed algorithm explicitly estimate the blur kernel which enhanced the image deblurring. In Chapter 5, a BID method based on alternative minimization of Kullback-Leibler divergence (KLD) between a model family of distribution based on a linear image model and a desired distribution based on observed image is presented. This method is an extension and refined version of Seghouane [2011], where, in addition to Fourier deconvolution, a wavelet based regularization is added for image and blur estimation. This method can also be seen as cascaded EM in information geometric terms, and thus can also be seen as extension of the method presented in Chapter 3. All the presented methods are compared and evaluated against state-of-the-art image deblurring methods and the results are presented in respective chapters.

The second part of thesis presents the dictionary learning methods for sparse image representation and its application in image denoising and in-painting. This part also consists of a total three chapters- from Chapter 6 to 9. The basic concept of sparse representation and its ingredients, is presented in Chapter 6, along with a brief literature review of related methods. In last two decades the sparse and redundant representation based image models show significant results in image denoising, in-painting, deblurring, compression and super resolution. Sparsity is an inherent characteristic of natural signal and can be obtained by decomposing the signal into its elementary basis. These basis can be either selected from a predefined basis collection or learned from the data. The later approach is termed as dictionary learning (DL), which has gained attention in the recent years. A parallel DL method, based on profile likelihood estimation is presented in Chapter 7. A single eign decomposition is used to update the whole dictionary which is comparatively more efficient. In most DL methods the sparsity constraint is restricted to the sparse coding stage, in chapter 8 we present a novel DL method where the sparsity constraint is further enforced in the dictionary update stage. Within this framework, we not only proposed dictionary update stage that enforced sparsity, but can also enforce other constraints effectively such as smoothness of dictionary atoms. The classical DL methods such as KSVD [Aharon et al. 2006] and MOD [Engan et al. 1999] ignore the structure in data matrices, in Chapter 9 we proposed an intermediate of the implicit and explicit DL method, which account for the structure in the learned dictionary. This method is based on the observation presented in [Rubinstein et al. 2010], and is derived us-
ing the connection between principal component analysis (PCA) and singular value decomposition (SVD) of the data matrix. This new DL method bridging the gap between implicit and explicit dictionaries that results in fast and data adaptive dictionaries.

\section{Future Work}

The importance of images and its usage in this technological era is well reflected from its applications range. Although a lot of effort and work has been done in creating flawless and efficient imaging systems, the contamination of image data by corruption of degradation such as blur, noise and aliasing, can not be ruled out and are always present in one form or in other. Image restoration is a classical ill posed linear inverse problem and many different avenues have been explored to address this problem, but never the less it is still an active research problem. The future directions we presented below are mainly related to image deconvolution and dictionary learning methods for sparse image representation.

Image deconvolution is basically the recovery of sharp image from its blurred version. Although, dramatic progress has been afforded by recent algorithms yet many respects of the problem is remain challenging. One main issue is that many recent algorithms are mostly designed for non-blind deconvolution which heavily depends on the information regarding blurring operator and the original image. In real applications this information is not available or very ill modeled, thus many methods fail badly in real image deblurring. Many blind image deconvolution (BID) methods are also reported in the literature, but they involve many parameters which required fine tuning or limited to certain types of blurs. The BID, by nature is a very ill-conditioned and ill-posed inverse problem where a small perturbation can result in a totally different result, so the parameter based approaches are not very encouraged. Recently, sparsity has been used as a regularization in many linear inverse problems with better results. These sparse models can be adopted in BID more efficiently, as the blurred and non-blurred image share the same sparse coding. So the problem boils down to the estimate of the dictionary. This dictionary can be seen as a shifted version of the original dictionary which represents the original sharp image, as illustrated below,

\[
y = Hx + n = HDA + n \approx \hat{D}A + n
\]
where $\hat{D}$ represents the shifted version of dictionary $D$. In addition, the double sparsity constraint can be also applied for further regularization, specifically for images such as smoothness or shape edges.

Regarding the dictionary learning algorithms, many possible directions are still open to explore. For example, the enforcement of sparse and other data related constraints in the dictionary update stage. Although, very recently some methods [Sadegni et al. 2014] are reported with such structure, but they are limited to the sparse constraint only, in both stages. Another very important direction may be the training of both dictionaries, i.e., implicit and explicit, in double sparsity case on the observed data, this means the adaptation of implicit dictionary to the training data. This can reduce the computational burden and will also expose the hidden structure in the data matrix.
Appendices
Maximum Likelihood estimation via Kullback-Leibler divergence

Consider an observed image $y$ represents a degraded version of original image $x$, given as

$$
y = Hx + n \sim \mathcal{N}(Hx, \sigma^2_n I_n), \quad (A.1)$$

where $H$ is a square block-Toeplitz matrix accounting for space invariant blur and $n$ is a sample of zero-mean white Gaussian noise. In blind image deconvolution (BID) the blur $H$, original image $x$, and noise $n$ are considered unknown. Further let assume,

$$
x = s + n_2
\quad s = \sqrt{zu}
$$

Suppose $'\theta'$ represent the unknown vector $\theta = \{H, C_x, C_n\}$.

The maximum likelihood estimation (MLE) of $p(x)$ and $\theta$ can be defined as

$$
(p(x), \theta)_{MLE} = \arg \min_{p,\theta} KL(p(y, x) \parallel q(y, x; \theta)) \quad (A.2)
$$

This minimization can be estimated by an iterative, alternative minimization scheme, composed of the following two steps.

- First fix $'\theta'$ and minimize with respect to $p(x)$, as $p(x)^{k+1} = \arg \min_{p,\theta} KL(p(y, x) \parallel q(y, x; \theta))$

- In the second step minimize with respect to $\theta$, keeping $p(x)$ fix, given as $\theta^{k+1} = \arg \min_{\theta} KL(p(y, x) \parallel q(y, x; \theta))$
A.1 Minimization w.r.t. \( p(x) \)

\[
p(x)^{k+1} = \arg\min_{p} KL(p(y, x) \mid\mid q(y, x; \theta)) 
\]  
\( \text{(A.3)} \)

This minimization can be realized for \( p(x)_{x/y} = q(x; \theta)_{x/y} \). As \( p(y, x) = p(x/y)p(y) \) from eq. (A.3) we can write \( p(x/y) = q(x/y; \theta) \), so

\[
p(y, x) = q(x/y; \theta) + \log p(y) 
\]  
\( = -\frac{1}{2}(x - \mu_{x/y})^{T}C_{x/y}^{-1}(x - \mu_{x/y}) - \frac{1}{2}y^{T}C_{y}^{-1}y
\]  
\( \text{(A.4)} \)

from the image model (A.1), the covariance matrix can be written as,

\[
C_{{x,y}} = \begin{bmatrix} 
H^{T}C_{x} + C_{n} & HC_{x} \\
C_{y}H^{T} & C_{y} 
\end{bmatrix}
\]  

The inverse of this is,

\[
C_{x,y}^{-1} = \begin{bmatrix} 
C_{n}^{-1} & -C_{n}^{-1}H \\
-H^{T}C_{n}^{-1} & C_{x}^{-1} + H^{T}C_{n}^{-1}H \n\end{bmatrix}
\]

The mean and covariance equations are then given by,

\[
\mu_{x/y} = C_{x/y}H^{T}C_{n}^{-1}y \\
C_{x/y} = (C_{x}^{-1} + H^{T}C_{n}^{-1}H)^{-1}
\]

A.2 Minimization w.r.t. ‘\( \theta \)’

\[
\theta^{k+1} = \arg\min_{\theta} KL(p(y, x) \mid\mid q(y, x; \theta)) 
\]  
\( \text{(A.5)} \)

As, \( x = s + n_{2} \) with \( s = \sqrt{z}u \) and \( \theta = \{H, C_{x}, C_{n}\} \), the minimization of eq. (A.2) is a double minimization problem. For two data distributions \( p_{2} \) and \( q_{2} \),

\[
\theta^{k+1} = \arg\min_{p(s),\theta_{2}} KL(p_{2}(x, s) \mid\mid q_{2}(x, s; \theta_{2}))) 
\]  
\( \text{(A.6)} \)

where, \( \theta_{2} = \{C_{u}, C_{n2}\} \).

Again, we can estimate this minimization by an iterative, alternative minimization scheme, consist of following two steps.
A.2.1 Minimization w.r.t. $p(s)$

For fixed $\theta_2$ the minimization is given as

$$p(s, x)^{k+1} = \arg \min_{p(x, s)} \text{KL}(p_2(x, s) \parallel q_2(x, s; \theta_2)) \quad (A.7)$$

this minimization can be realized for $p(s/x) = q(s/x; \theta_2)$. As $p(s, x) = p(s/x)p(x)$, from eq.(A.7) we can write, $p(s/x) = q(s/x; \theta_2)$, so

$$p(x, y) = q(s/x; \theta_2)p(x)$$

$$\log p(y, x) = \log q(s/x; \theta_2) + \log p(x)$$

$$= -\frac{1}{2}(s - \mu_{s/x})^T C_{s/x}^{-1}(s - \mu_{s/x}) - \frac{1}{2}x^T C_x^{-1}x \quad (A.8)$$

Then the covariance matrix can be written,

$$C_{s,x} = \left[ \begin{array}{cc} zC_u + C_{n2} & C_{n2} \\ C_{n2} & C_u \end{array} \right]$$

The inverse of this matrix is,

$$C_{s,x}^{-1} = \left[ \begin{array}{cc} C_{n2}^{-1} & -C_{n2}^{-1} \\ -C_{n2}^{-1} & zC_u^{-1} + C_{n2}^{-1} \end{array} \right]$$

The mean and covariance equations are given by,

$$\mu_{s/x} = C_{s/x}C_{n2}^{-1}x$$

$$C_{s/x} = (zC_x^{-1} + C_n^{-1})^{-1}$$

Putting them in eq.(A.8), and taking the 2nd order terms we get

$$\log p(y, x) = \left[ \begin{array}{c} x \\ s \end{array} \right]^T \left[ \begin{array}{cc} C_{s,x}^{-1} & -C_{n2}^{-1} \\ -C_{n2}^{-1} & zC_u^{-1} + C_{n2}^{-1} \end{array} \right] \left[ \begin{array}{c} x \\ s \end{array} \right] \quad (A.9)$$

The inverse of $\psi$ is

$$C_{p(x,s)} = \left[ \begin{array}{cc} C_x & C_x C_{n2}^{-1} C_{s/x} \\ -C_{s/x} C_{n2}^{-1} C_x & C_{s/x} + C_{s/x} C_{n2}^{-1} C_x C_{n2} C_{s/x} \end{array} \right]$$
As, $C_{s/x} = (Cu_{-1} + C_{n2}^{-1})^{-1}$

$$C_{p(x,s)} = \begin{bmatrix} C_x & C_x^{-1}C_u \\ C_x^{-1}C_x & Cu - C_nC_x^{-1}C_n2CuC_x^{-1}C_u \end{bmatrix}$$

Which gives $p_{s}^{k+1}(x,s) \sim \mathcal{N}(0,C_{p(x,s)})$.

A.2.2 Minimization w.r.t. $\theta_2$

For fixed $p(s)$ the minimization can be written as,

$$\theta_2^{k+1} = \arg \min_{\theta_2} KL(p_2(x,s) \parallel q_2(x,s;\theta_2)).$$

$$= \arg \min_{\theta_2} KL(\mathcal{N}(0,C_{p(x,s)}) \parallel \mathcal{N}(0,C_{s,k})).$$

This minimization is archived when $C_{p(x,s)} \approx C_{s,k}^{k+1}$.

$$\begin{bmatrix} C_x & C_x^{-1}C_u \\ C_uC_x^{-1} & Cu - C_nC_x^{-1}C_n2CuC_x^{-1}C_u \end{bmatrix} \approx \begin{bmatrix} C_{u_{k+1}} + C_{n2k+1} & C_{u_{k+1}} \\ C_{u_{k+1}} & C_{u_{k+1}} \end{bmatrix}$$

Equating these two matrix provide the equations for the unknowns parameters of $\theta$, given as

$$C_{u_{k+1}} = C_{u_k} - C_{u_k}C_xC_{u_k} + C_{u_k}C_x^{-1}C_xC_{u_k}^{-1}C_u$$

$$C_{n2k+1} = \frac{1}{p} tr(C_{x_k} - C_{u_{k+1}})$$

$$C_{s_{k+1}} = C_{u_{k+1}} + C_{n2k+1}$$

$$\hat{x}_{k+1} = C_{u_{k+1}}C_{s_{k+1}} \hat{x}_k$$

$$H_{k+1} = C_yC_y^{-1}H_kC_{s_{k+1}}^{-1}$$

$$\hat{s} = \frac{C_{s_{k+1}}}{zC_{u_{k+1}} + C_{n2k+1}} \hat{x}_{k+1}$$

$$\hat{s}_{k+1} = \frac{C_{s_{k+1}}}{zC_{u_{k+1}}} \hat{s}_k$$
Appendix B

Penalized rank-1 matrix approximation

The updates of \( d_k \) and \( a^\text{row}_k \) to be used for alternate minimization of (8.20) in chapter 8 when \( \alpha_2 = \alpha_3 = 0, \alpha_1 \neq 0 \) and \( \alpha_4 \neq 0 \) is derived in this appendix. For fixed \( a^\text{row}_k \), the \( d_k \) that minimizes (8.20) is given by

\[
d_k = \arg\min_{d_k} \| E_k - d_k a^\text{row}_k \|_F^2 + \alpha_1 \| a^\text{row}_k \|_1 + \alpha_4 a^\text{row}_k \Omega_2 a^\text{row}_k^T
\]

which with the constraint \( \|d_k\|_2 = 1 \) gives

\[
d_k = \frac{E_k a^\text{row}_k^T}{\|E_k a^\text{row}_k^T\|_2}.
\]
For fixed $\mathbf{d}_k$, $\mathbf{a}^{\text{row}}_k$ is derived by minimizing

$$
\mathbf{a}^{\text{row}}_k = \arg \min_{\mathbf{a}^{\text{row}}_k} \| \mathbf{E}_k - \mathbf{d}_k \mathbf{a}^{\text{row}}_k \|^2_F + \alpha_1 \| \mathbf{a}^{\text{row}}_k \|_1 + \alpha_4 \mathbf{a}^{\text{row}}_k \Omega_2 \mathbf{a}^{\text{row}\top}_k
$$

$$
= \arg \min_{\mathbf{a}^{\text{row}}_k} \sum_{i=1}^n \sum_{j=1}^p \left( e_{kij} - d_{kij} a_{kij}^{\text{row}} \right)^2 + \alpha_1 \sum_{j=1}^p \text{sgn}(a_{kij}^{\text{row}}) a_{kij}^{\text{row}}
$$

$$
+ \alpha_4 a_{k1}^{\text{row}^2} + \alpha_4 a_{kp}^{\text{row}^2} + \alpha_4 \sum_{j=2}^{p-1} (a_{kj+1}^{\text{row}} - 2a_{kj}^{\text{row}} + a_{kj-1}^{\text{row}})^2
$$

$$
= \arg \min_{\mathbf{a}^{\text{row}}_k} \sum_{j=1}^p \left( \sum_{i=1}^n e_{kij} - \sum_{i=1}^n e_{kij} d_{kij} a_{kij}^{\text{row}} + a_{kij}^{\text{row}^2} \right)
$$

$$
+ \alpha_1 \text{sgn}(a_{kij}^{\text{row}}) a_{kij}^{\text{row}} + \alpha_4 a_{k1}^{\text{row}^2} + \alpha_4 a_{kp}^{\text{row}^2}
$$

$$
+ \alpha_4 \sum_{j=2}^{p-1} (a_{kj+1}^{\text{row}} - 2a_{kj}^{\text{row}} + a_{kj-1}^{\text{row}})^2
$$

$$
= \arg \min_{\mathbf{a}^{\text{row}}_k} \sum_{j=1}^p \left( a_{kij}^{\text{row}^2} - 2(d_k^\top \mathbf{E}_k)_{kj} a_{kij}^{\text{row}} + \alpha_1 \text{sgn}(a_{kij}^{\text{row}}) a_{kij}^{\text{row}} \right)
$$

$$
+ \alpha_4 a_{k1}^{\text{row}^2} + \alpha_4 a_{kp}^{\text{row}^2} + \alpha_4 \sum_{j=2}^{p-1} (a_{kj+1}^{\text{row}} - 2a_{kj}^{\text{row}} + a_{kj-1}^{\text{row}})^2
$$

where $e_{kij}$ is the $ij^{th}$ entry of the matrix $\mathbf{E}_k$ and $(d_k^\top \mathbf{E}_k)_j$ is the $j^{th}$ entry of the vector $\mathbf{d}_k^\top \mathbf{E}_k$. The first entry of $\mathbf{a}^{\text{row}}_k$, $a_{k1}^{\text{row}}$ is given by

$$
a_{k1}^{\text{row}} = \arg \min_{a_{k1}^{\text{row}}} a_{k1}^{\text{row}^2} - 2(d_k^\top \mathbf{E}_k)_{1} a_{k1}^{\text{row}}
$$

$$
+ \alpha_1 \text{sgn}(a_{k1}^{\text{row}}) a_{k1}^{\text{row}} + \alpha_4 a_{k1}^{\text{row}^2}
$$

$$
= \frac{1}{(1 + \alpha_4)} \text{sgn} \left( (d_k^\top \mathbf{E}_k)_1 \right) \left( | (d_k^\top \mathbf{E}_k)_1 | - \frac{\alpha_1}{2} \right)
$$

following the same derivation we obtain the last entry $a_{kp}^{\text{row}}$ of $\mathbf{a}^{\text{row}}_k$ as

$$
a_{kp}^{\text{row}} = \frac{1}{(1 + \alpha_4)} \text{sgn} \left( (d_k^\top \mathbf{E}_k)_p \right) \left( | (d_k^\top \mathbf{E}_k)_p | - \frac{\alpha_1}{2} \right)
$$
The entries \( a_{kj}^{\text{row}}, j = 2, \ldots, P - 1 \), of \( a_k^{\text{row}} \) are given by

\[
\begin{align*}
a_{kj}^{\text{row}} &= \arg \min_{a_{kj}} a_{kj}^{\text{row}}^2 - 2(d_k^T E_k) j a_{kj}^{\text{row}} + \alpha_1 \text{sgn}(a_{kj}^{\text{row}}) a_{kj}^{\text{row}} \\
&\quad + \alpha_4 (a_{kj+1}^{\text{row}} - 2a_{kj}^{\text{row}} + a_{kj-1}^{\text{row}})^2 \\
&= \frac{1}{(1 + 2\alpha_4)} \text{sgn} \left( (d_k^T E_k) j + 2\alpha_4 (a_{kj+1}^{\text{row}} + a_{kj-1}^{\text{row}}) \right) \\
&\quad \left( |(d_k^T E_k) j + 2\alpha_4 (a_{kj+1}^{\text{row}} + a_{kj-1}^{\text{row}}) | - \frac{\alpha_1}{2} \right)_+.
\end{align*}
\]

The update of \( d_k \) and \( a_k^{\text{row}} \) to be used for alternate minimization of (20) for \( \alpha_1 = \alpha_4 = 0 \) and \( \alpha_1, \alpha_4 \neq 0 \) can be obtained following a similar derivation.
Penalized rank-1 matrix approximation


Eckart, C. and Young, G., 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1 (1936), 211–218. (cited on pages 80 and 100)


MATLAB, 2014. version 8.4.0 (R2014b). The MathWorks Inc., Natick, Massachusetts. (cited on page 46)


