Title

Statistical and Adaptive Patch-based Image Denoising

Permalink

https://escholarship.org/uc/item/76j927j2

Author

Luo, Enming

Publication Date

2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Statistical and Adaptive Patch-based Image Denoising

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Enming Luo

Committee in charge:

Professor Truong Nguyen, Chair
Professor Ery Arias-Castro
Professor Joseph Ford
Professor Bhaskar Rao
Professor Zhuowen Tu

2016

The dissertation of Enming Luo is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2016

DEDICATION

To my parents.

EPIGRAPH

All human wisdom is summed up in two words: wait and hope.

Alexandre Dumas

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Foremost, I would like to give my sincere thanks to my PhD advisor Professor Truong Nguyen. Professor Nguyen has given me much freedom and constant support to conduct research. He is incredibly patient and kind. Without his encouragement, necessary guidance and constructive discussions, the thesis would not be possible.

I am particularly grateful to our lab alumni Stanley H. Chan who is now a professor in Purdue university. It has been great pleasure collaborating with Professor Chan. His strong insights and valuable suggestions have made the research work possible. I would never forget the endless discussions, precious sharing, and positive criticisms.

I give special thanks to my other committee members, Professors Ery Arias-Castro, Joseph Ford, Bhaskar Rao and Zhuowen Tu. Thank you so much for some interesting discussions and useful feedback on many of my research problems, as well as your time and help to improve the thesis.

I would also like to thank my two other co-authors Shengjun Pan and Shibin Parameswaran. Thank you for the inspiring discussions and I have had a lot of fun working with you. I also take this opportunity to thank my wonderful fellow colleagues and friends in the video processing lab. With your existence, the friendly atmosphere in the lab has made the work pleasant and my PhD life joyous.

Last but not least, my heartfelt thanks go to my dear parents and my loving girlfriend. Words cannot describe my gratitude for your love, trust and always standing by my side.

Part of this dissertation is a reprint of published/submitted papers. Chapter 2, in part, is a reprint of a published conference paper: E. Luo, S. H. Chan, S. Pan, and T. Q. Nguyen, “Adaptive Non-local Means for Multiview Image Denoising: Searching for the Right Patches via a Statistical Approach,” in *Proc. IEEE Intl. Conf. Image Process. (ICIP’13)*, pp. 543-547, Sep. 2013.

Chapter 3, in part, is a reprint of a published journal paper: E. Luo, S. H. Chan, and T. Q. Nguyen, “Adaptive Image Denoising by Targeted Databases,” *IEEE Trans. Image Process. (TIP’15)*, vol. 24, no. 7, pp. 2167-2181, Jul. 2015,

and a published conference paper: E. Luo, S. H. Chan, and T. Q. Nguyen, “Image Denoising by Targeted External Databases,” in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Process. (ICASSP’14)*, pp. 2469-2473, May 2014.

Chapter 4, in part, is a reprint of a submitted journal paper: E. Luo, S. H. Chan, and T. Q. Nguyen, “Adaptive Image Denoising by Mixture Adaptation,” submitted to *IEEE Trans. Image Process. (TIP’16)*, Jan. 2016, and a published conference paper: S. H. Chan, E. Luo, and T. Q. Nguyen, “Adaptive Patch-based Image Denoising by EM-adaptation,” in *Proc. IEEE Global Conf. Signal and Information Process. (GlobalSIP’15)*, Dec. 2015.

VITA

2007	B. Eng. in Electrical Engineering, Jilin University, China
2009	Mphil. in Electrical Engineering, Hong Kong University of Science and Technology, Hong Kong
2016	Ph. D. in Electrical Engineering (Signal and Image Processing), University of California, San Diego, USA

PUBLICATIONS

E. Luo, S. H. Chan, and T. Q. Nguyen, “Adaptive Image Denoising by Mixture Adaptation,” submitted to *IEEE Trans. Image Process. (TIP’16)*, Jan. 2016.

E. Luo, S. H. Chan, and T. Q. Nguyen, “Adaptive Image Denoising by Targeted Databases,” *IEEE Trans. Image Process. (TIP’15)*, vol. 24, no. 7, pp. 2167-2181, Jul. 2015

S. H. Chan, E. Luo, and T. Q. Nguyen, “Adaptive Patch-based Image Denoising by EM-adaptation,” in *Proc. IEEE Global Conf. Signal and Information Process. (GlobalSIP’15)*, Dec. 2015.

E. Luo, S. H. Chan, and T. Q. Nguyen, “Image Denoising by Targeted External Databases,” in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Process. (ICASSP’14)*, pp. 2469-2473, May 2014.

E. Luo, S. H. Chan, S. Pan, and T. Q. Nguyen, “Adaptive Non-local Means for Multiview Image Denoising: Searching for the Right Patches via a Statistical Approach,” in *Proc. IEEE Intl. Conf. Image Process. (ICIP’13)*, pp. 543-547, Sep. 2013.

E. Luo, S. Pan, and T. Q. Nguyen, “Generalized Non-local Means for Iterative Denoising,”, in *Proc. 20th Euro. Signal Process. Conf. (EUSIPCO’12)*, pp. 260-264, Aug. 2012.

ABSTRACT OF THE DISSERTATION

Statistical and Adaptive Patch-based Image Denoising

by

Enming Luo

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California, San Diego, 2016

Professor Truong Nguyen, Chair

With the explosion in the number of digital images taken every day, people are demanding more accurate and visually pleasing images. However, the captured images by modern cameras are inevitably degraded by noise. Besides deteriorating image visual quality, noise also degrades the performance of high-level vision tasks such as object recognition and tracking. Therefore, image denoising is a critical preprocessing step. This thesis presents novel contributions to the field of image denoising.

Image denoising is a highly ill-posed inverse problem. To alleviate the ill-posedness, an effective prior plays an important role and is a key factor for successful image denoising. With abundance of images available online, we propose to obtain priors from external image databases. In this thesis, we perform statistical analyses and rigorous derivations on how to obtain effective priors by utilizing external databases. For three denoising applications under different external settings, we show how we can explore effective priors and accordingly we present adaptive patch-based image denoising algorithms. In specific, we propose three adaptive algorithms: (1) adaptive non-local means for multiview image denoising; (2) adaptive image denoising by targeted databases; (3) adaptive image denoising by mixture adaption.

In (1), we present how to improve the non-local prior by finding more relevant patches in the multiview image denoising setting. We propose a method that uses a robust joint-view distance metric to measure the similarity of patches and derive an adaptive procedure to determine the optimal number of patches for final non-local means denoising. In (2), we propose to switch from generic database to targeted database, *i.e.,* for specific objects to be denoised, only targeted databases with relevant images should be used. We explore both the group sparsity prior and the localized Bayesian prior, and show how a near optimal and adaptive denoising filter can be designed so that the targeted database can be maximally utilized. In (3), we propose an adaptive learning procedure called Expectation-Maximization (EM) adaptation. The adaptive process takes a generic prior learned from a generic database and transfers it to the image of interest to create a specific prior. This adapted prior better captures the distribution of the image of interest and is consistently better than the un-adapted one. For all the three denoising applications, we conduct various denoising experiments. Our proposed adaptive algorithms have some superior denoising performance than some state-of-the-art algorithms.

Chapter 1

Introduction

1.1 Image Restoration

Image restoration is a classical signal recovery problem where the goal is to restore a clean image from its degraded observation. The problem has been studied for decades, but it is still a vibrant research topic because it plays an important role in many areas such as consumer camera imaging, medical imaging, satellite imaging and military surveillance.

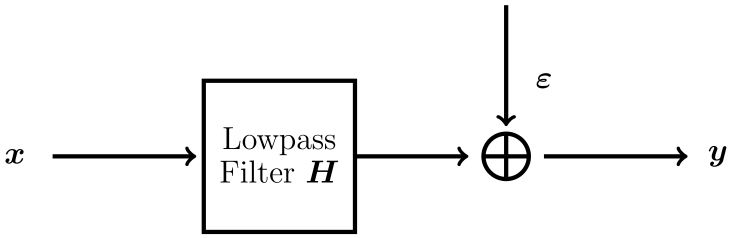


Figure 1.1: Degradation model: $\boldsymbol{y}$ is corrupted image by first passing the original image $\boldsymbol{x}$ into a lowpass filter $\boldsymbol{H}$ and then adding noise $\boldsymbol{\varepsilon}$ to it.

Due to physical limitations of the imaging system and various environmental factors, the captured image represents a degraded version of the original scene. Some common degradations include blur, noise, geometric degradations and color imperfections. While some degradations are complex, the majority can be modeled as the linear degradation model. As shown in Figure 1.1, the original high-quality image $\boldsymbol{x}$ undergoes blur $\boldsymbol{H}$ first and is then corrupted by noise $\boldsymbol{\varepsilon}$. Blurring is usually modeled as the convolution of an image with a blur kernel (also known as the point spread function, PSF). There are different types of blurs. If the PSF is the same throughout the image, the blur is termed spatially-invariant (*e.g.,* camera motion blur). Otherwise, if the PSF is different for different image pixels, the blur is termed spatially-variant (*e.g.,* lens defocus blur). Besides blur, noise causes random fluctuation in pixel intensity which obscures the original image content. The characteristics of noise depend on the noise sources. Some common types of noise include electronic noise, shot noise, salt-and-pepper noise, speckle noise, and quantization noise. To put things together, the imaging system is mathematically modeled as [5]

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\varepsilon}, \tag{1.1}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is a lexicographically ordered vector denoting the unknown high-quality image, $\boldsymbol{y} \in \mathbb{R}^N$ denotes the observed degraded image, $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ denotes the linear transformation operator, and $\boldsymbol{\varepsilon} \in \mathbb{R}^N$ denotes the random noise. Image restoration aims to reconstruct the original sharp image $\boldsymbol{f}$ from the blurry and noisy observation $\boldsymbol{y}$.

1.2 Image Denoising

If $\boldsymbol{H}$ in (1.1) is an identity matrix, image restoration becomes an image denoising problem, the goal of which is to restore a clean image from its noisy observation. In the thesis, we focus on the image denoising problem. This problem has been studied for decades, but it still remains a fundamental image processing task both as a process itself, and as a component in other processes. For one reason, noise removal improves the image visual quality and improves the performance of many subsequent computer vision applications such as object detection; For another, image denoising serves as a test bed for a wide range of other inverse problems in image processing such as image deblurring, image inpainting and super-resolution.

Though noise is introduced in various forms at different stages in the imaging system, two major types of noise are: electronic noise and shot noise. Electronic noise stems from different causes such as instability of voltage, temperature fluctuation of electronic components, quantization error of analog-to-digital conversion, etc. Typically electronic noise is modeled to be Gaussian distributed.

On the other hand, shot noise is caused by the random arrival of photons. When photons hit the image sensors, they arrive in a random way as opposed to uniformly, which results in shot noise. Typically this photon counting process is modeled to be Poisson distributed and is especially serious when the light condition is bad. However, shot noise can be transformed to be Gaussian distributed via Anscombe root transformation [6]. Therefore, shot noise can also be addressed by any denoising algorithm for Gaussian noise.

In the image denoising literature, the mixture of different noises is popularly modeled as additive white Gaussian noise (AWGN) with zero mean. For one reason, denoising methods targeted at Gaussian noise are practically applicable for other types of noise. For another, Gaussian distribution has mathematical tractability and could significantly facilitate the mathematical analysis when designing denoising methods. In our thesis, we consider the classical image denoising problem: Given an additive i.i.d. Gaussian noise model,

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon}, \tag{1.2}$$

our goal is to find an estimate of $\boldsymbol{x}$ from $\boldsymbol{y}$, where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the (unknown) clean image, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I}) \in \mathbb{R}^n$ denotes the additive i.i.d. Gaussian noise with σ^2 noise variance, and $\boldsymbol{y} \in \mathbb{R}^n$ denotes the observed noisy image.

1.3 Denoising Literature

Image denoising is a long-lasting problem and numerous denoising algorithms have been proposed in the past few decades. Among the various denoising methods, most of them can be classified as either a spatial domain denoising method or a transform domain denoising method. A spatial domain denoising method alters the pixels directly in the spatial domain such as Gaussian filtering, anisotropic filtering [7], bilateral filtering [8] and steering kernel regression [9]. A transform domain denoising method converts an image into a transform domain and then alters the transform coefficients to reduce noise such as Wiener filtering [10–12], wavelet-based techniques [13,14] and dictionary-based techniques [15,16].

In our thesis, we focus on the class of *patch-based image denoising* algorithms [11,12,17–24]. These methods are the most highly-regarded class of methods, and have drawn a lot of attention in the denoising community in recent years. The basic idea of a patch-based denoising algorithm is to partition a noisy image into overlapped or non-overlapped patches, denoise each patch through the statistics of some other reference patches, and then combine all the denoised patches to yield a final denoised image. In specific, given a $\sqrt{d} \times \sqrt{d}$ patch $\boldsymbol{q} \in \mathbb{R}^d$ from the noisy image $\boldsymbol{y}$, the algorithm finds a set of reference patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k \in \mathbb{R}^d$ and applies some linear (or non-linear) function Φ to obtain an estimate $\widehat{\boldsymbol{p}}$ of the unknown clean patch $\boldsymbol{p}$ as

$$\widehat{\boldsymbol{p}} = \Phi(\boldsymbol{q}; \boldsymbol{p}_1, \ldots, \boldsymbol{p}_k). \tag{1.3}$$

For example, in the classic non-local means (NLM) [17], Φ is a weighted average of the reference patches,

$$\widehat{\boldsymbol{p}} = \frac{\sum\limits_{i=1}^{k} e^{-\|\boldsymbol{q}-\boldsymbol{p}_i\|^2/h^2} \boldsymbol{p}_i}{\sum\limits_{i=1}^{k} e^{-\|\boldsymbol{q}-\boldsymbol{p}_i\|^2/h^2}}, \tag{1.4}$$

where h is a decay parameter.

Image denoising is an ill-posed inverse problem because the information provided by the noisy observation is not sufficient to ensure a unique and stable restored image in the right class [25]. Thus, it is necessary to regularize the inverse problem by adding prior knowledge of the image. Patch-based image denoising algorithms exploit the statistics of other reference patches for denoising, and are actually enforcing prior knowledge in solving the ill-posed inverse problem. For example, NLM in (1.4) is actually enforcing a non-local prior. In other words, the best estimate $\widehat{\boldsymbol{p}}$ in NLM should be close to the non-local reference patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k \in \mathbb{R}^d$. Many other patch-based denoising algorithms are also assuming some notions of prior, either explicitly or implicitly.

Depending on the source of the reference patches or training samples in general, the priors can be classified as *internal* prior [26] or *external* prior [27, 28]. The internal prior is trained or learned using samples from the single noisy image while the training samples for the external prior are from a database of external clean images. To date, there is no clear conclusion about which of the internal or external methods is better. However, it is generally observed that internal methods are computationally less expensive, whereas external methods have greater potential to achieve better performance by mining appropriate datasets.

The other difference between internal and external priors is the informativeness. For internal priors, since the statistics is learned specifically for the image of interest, there is no redundancy as opposed to learning from a database of generic images. However, the challenge is that the image is noisy so that the priors do not completely reflect the ground truth. On the other hand, while external priors tend to over-learn for a specific image, the priors are indeed computed from ground truth clean images.

In this thesis, we discuss image denoising under three different external settings. Thus our training samples are from external databases. We present how to obtain effective priors by either mining appropriate external databases or by combining the external prior with the internal prior in a systematic way.

1.4 Contribution

Through advancing prior modeling, we can effectively advance our capability for successful image denoising. In this thesis, we focus on how to find good priors and specifically perform rigorous statistical analyses on how to obtain effective priors through different ways. For different scenarios, we utilize the effective priors and design adaptive patch-based image denoising algorithms that achieve superior denoising performance.

This remainder of this thesis is organized as follows:

1. **Chapter 2 – Multiview Image Denoising**

In this chapter, we discuss image denoising when we have multiple noisy views of the same scene. We present an adaptive procedure derived from statistical properties of the estimates to determine the optimal number of patches to be used. The returned reference patches are more similar to the noisy patch of interest and thus provide a good non-local prior for the final non-local means denoising method.

2. **Chapter 3 – Targeted Image Denoising**

   In this chapter, we discuss image denoising when a targeted external database is available. A targeted database contains images that are only relevant to the noisy image of interest and thus provide exceptional prior for the denoising task. We explore both the group sparsity prior and the localized Bayesian prior, and present how the targeted database can be maximally utilized by proposing a near optimal and adaptive linear denoising filter.

3. **Chapter 4 – Guided Image Denoising**

   In this chapter, we discuss image denoising when only a generic database (as opposed to a targeted database) is available. We present how a generic prior learned from a generic database can be adapted to the image of interest to generate a specific prior. During the adaptation, the generic model parameters serve as a "guide" when learning the new model parameters. We show that the adapted prior is consistently better than the originally un-adapted prior for image denoising.

Chapter 2

Adaptive Non-local Means for Multiview Image Denoising

2.1 Introduction

In this chapter, we consider multiview image denoising A multiview system captures multiple but non-identical observations of the same scene. However, due to the lighting condition, physical limitations and sometimes malfunctions of the system, the captured views are often corrupted with noise. Multiview image denoising aims at denoising one view at a time by using all the similar but non-identical noisy views.

Multiview image denoising could be thought of as a special case of external denoising in the sense that the external database consists of adjacent noisy views, which are used to assist in denoising of the current view. Defining $\boldsymbol{g}^{(k)} = \boldsymbol{f}^{(k)} + \boldsymbol{n}^{(k)}$ for $k = 1, \ldots, K$ a sequence of K noisy observations, where $\boldsymbol{n}^{(k)}$ is a vector of i.i.d. Gaussian noise of variance σ^2, our goal is to recover $\boldsymbol{f}^{(1)}, \ldots, \boldsymbol{f}^{(K)} \in \mathbb{R}^N$ from $\boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(K)} \in \mathbb{R}^N$.

The method we consider in our work is the NLM [17] filtering approach. Letting $\boldsymbol{p}_i^{(k)} \in \mathbb{R}^d$ be a patch centered at the ith pixel in the kth image, NLM computes the weights as

$$W_{ij}^{(kl)} = \exp\left\{ -\frac{\|\boldsymbol{p}_i^{(k)} - \boldsymbol{p}_j^{(l)}\|^2}{h^2} \right\}, \tag{2.1}$$

and denoises the images as

$$\begin{bmatrix} \widehat{\boldsymbol{f}}^{(1)} \\ \vdots \\ \widehat{\boldsymbol{f}}^{(K)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}^{(1)} & & \\ & \ddots & \\ & & \boldsymbol{D}^{(K)} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{W}^{(11)} & \dots & \boldsymbol{W}^{(1K)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{W}^{(K1)} & \dots & \boldsymbol{W}^{(KK)} \end{bmatrix} \begin{bmatrix} \boldsymbol{g}^{(1)} \\ \vdots \\ \boldsymbol{g}^{(K)} \end{bmatrix}$$

here, $\boldsymbol{D}^{(k)} = \mathrm{diag}([\boldsymbol{W}^{(k1)}, \dots, \boldsymbol{W}^{(kK)}]^T \mathbf{1}_{KN\times 1})$ is a diagonal matrix for normalization. For notation simplicity and without loss of generality, in the rest of the chapter we consider denoising $\boldsymbol{f}^{(1)}$ using $\boldsymbol{g}^{(1)}, \dots, \boldsymbol{g}^{(K)}$.

One of the biggest issues in the above multiple-image NLM is that the Euclidean distance metric $\|\boldsymbol{p}_i^{(k)} - \boldsymbol{p}_j^{(l)}\|^2$ does not capture the true similarity between $\boldsymbol{p}_i^{(k)}$ and $\boldsymbol{p}_j^{(l)}$. In other words, two patches could be different, but their Euclidean distance might be close because of noise. Therefore, to improve the performance of NLM, one fundamental challenge is how to search for the *right* patches.

2.1.1 Overview of the Proposed Method

To tackle the problem, our first attempt is to improve the metric. In multi-view denoising problems, we consider the robust joint-view distance [29] to measure the similarity. Given a patch $\boldsymbol{p}_i^{(1)}$, we first compute the correspondences (disparity maps) from view 1 to views $2, \dots, K$ using off-the-shelf algorithms such as block matching [30], optical flow [31], or TV/L1 [32]. Denoting the disparities as $\{q_i^{(1)}, \dots, q_i^{(K)}\}$ for $i = 1, \dots, N$, we define the distance between two patches $\boldsymbol{p}_i^{(1)}$ and $\boldsymbol{p}_j^{(1)}$ as

$$D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)}) \overset{\text{def}}{=} \sum_{k=1}^{K} \left\| \boldsymbol{p}^{(k)}_{i+q_i^{(k)}} - \boldsymbol{p}^{(k)}_{j+q_j^{(k)}} \right\|^2. \tag{2.2}$$

Replacing $\|\boldsymbol{p}_i^{(1)} - \boldsymbol{p}_j^{(1)}\|^2$ by $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$ could improve the robustness of patch similarity, but as observed by Kervrann and Boulanger [18], the presence of many dissimilar patches could still cause unwanted bias in the denoising procedure. Therefore, our second modification is to keep a set of m most similar patches so that there are only m non-zero entries in each row of $\boldsymbol{W}^{(kl)}$.

To obtain a set of m most similar patches with respect to $\boldsymbol{p}_i^{(1)}$, we compute $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$ for all possible j's and keep the m best matches. We then define the set of m best matching patches in View 1:

$$\Omega_{i,m}^{(1)} = \left\{ j \mid \text{the indices of } m \text{ smallest } D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)}) \right\}, \tag{2.3}$$

and the sets of m best matching patches in other views:

$$\Omega_{i,m}^{(k)} = \left\{ j + q_j^{(k)} \mid j \in \Omega_{i,m}^{(1)} \right\}, \quad k = 2, \ldots, K. \tag{2.4}$$

Note that $\Omega_{i,m}^{(k)} \subset \Omega_{i,m+1}^{(k)}$. Restricting $W_{ij}^{(kl)}$ to $\Omega_{i,m}^{(k)}$ yields the following denoising algorithm:

$$\widehat{f}_{i,m}^{(1)} = \frac{\sum_{k=1}^{K} \sum_{j \in \Omega_{i,m}^{(k)}} W_{ij}^{(1k)} g_j^{(k)}}{\sum_{k=1}^{K} \sum_{j \in \Omega_{i,m}^{(k)}} W_{ij}^{(1k)}} \overset{\text{def}}{=} \sum_{k=1}^{K} \sum_{j \in \Omega_{i,m}^{(k)}} \widetilde{W}_{ij}^{(1k)} g_j^{(k)}, \tag{2.5}$$

where $\widetilde{W}_{ij}^{(1k)} \overset{\text{def}}{=} W_{ij}^{(1k)} / \sum_{k=1}^{K} \sum_{j \in \Omega_{i,m}^{(k)}} W_{ij}^{(1k)}$.

It can be seen that the performance of the above denoising procedure in (2.5) depends on how m is chosen. Intuitively, m should not be too small or too large, for otherwise insufficient patches or excessive dissimilar patches will be included. Therefore, we propose an adaptive scheme to choose the optimal m. Specifically, we increase from $m-1$ to m until one of the following criteria is met

- (Condition 1) Denoising Consistency:

$$\left| \widehat{f}_{i,m}^{(1)} - \widehat{f}_{i,m'}^{(1)} \right| \geq \gamma, \tag{2.6}$$

  for any $m' = 1, \ldots, m-1$, which requires that the deviation between the current and the previous estimates to be small.

- (Condition 2) Intersection of Confidence Interval:

$$\bigcap_{t=1}^{m} [\alpha_{i,t},\ \beta_{i,t}] = \emptyset, \tag{2.7}$$

  which requires that the confidence intervals $[\alpha,\ \beta]$ of the current and previous estimate intersect.

As we shall see, this adaptive scheme will give us a set of m most similar patches that improve the denoising quality.

2.1.2 Related Works

The literature on single image denoising is rich. However, state-of-the-art single image denoising methods such as NLM [17], BM3D [11], LPG-PCA [22] and many other methods reported in [33] are insufficient for multiview denoising, as these methods assume that similar patches exist at different locations within the image. Extensions of these methods such as [34, 35] are also insufficient for multiview denoising due to similar reasons.

Direct extension of single image denoising methods have been proposed to handle video denoising. In [36], Buades et al. proposed a video denoising method by allowing NLM to search for similar patches in adjacent frames. Similar ideas are applicable to BM3D, yielding the benchmark video denoising method VBM3D [19] and BM4D [37]. One problem of these methods is that displacement across different images is never explicitly used. While the authors of NLM [36], VBM3D [19] and BM4D [37] claim this as an advantage, Liu and Freeman [38] showed that reliable motion vectors are indeed helpful.

Another problem of video NLM [36], VBM3D [19] and BM4D [37] is that the number of patches increases as the number of images increases. This is undesirable because there will be many small but non-zero weights which could reduce the denoising result. Therefore, Kervrann and Boulanger [18] proposed a method to adaptively look for optimal spatial search window size. Later, in [39] they applied similar ideas to videos and demonstrated outstanding performance over other methods [40–44].

2.1.3 Contributions and Outline

Our proposed method utilizes the strength of the robust metric proposed in [29], and the spatial adaptivity proposed by Kervrann and Boulanger [18], and V. Katkovnik [45]. The key contributions are:

- **New Algorithm for Multiple Image Denoising**: Our new denoising scheme in (2.5) uses only similar patches defined in (2.3) and (2.4). As will be discussed in Section 2.3, the new algorithm out-performs existing methods.
- **Adaptive Neighborhood Selection**: We propose an adaptive scheme to determine the optimal m. The optimal m allows us to denoise the image with the right number of relevant patches, as contrast to classical NLM where all patches are used.

In the following sections, we discuss our proposed method in Section 2.2 and show experimental results in Section 2.3. Conclusion is given in Section 2.4.

2.2 Proposed Method

In this section, we describe the proposed method. For clarity we present the overall algorithm in Algorithm 1, and discuss the ideas in the following subsections.

Algorithm 1 Proposed Algorithm

```
1: Input: $\boldsymbol{g}^{(1)}, \ldots, \boldsymbol{g}^{(K)}$.
2: Output: $\widehat{\boldsymbol{f}}^{(1)}$.
3: Pre-denoise $\{\boldsymbol{g}^{(k)}\}$ by single-view methods to obtain $\{\overline{\boldsymbol{g}}^{(k)}\}$.
4: Run optical flows to obtain $\{\boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(k)}\}$.
5: for all $i$ pixels do
6:     Compute $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$ using (2.2).
7:     Compute the sets $\Omega_{i,m}^{(k)}$ using (2.3) and (2.4).
8:     Compute $\widehat{f}_{i,m}^{(1)}$ using (2.5).
9:     If $\widehat{f}_{i,m}^{(1)}$ does not satisfy Condition 1 or 2, then increase $m$ and repeat Lines 8 – 9.
10: end for
```

2.2.1 Pre-processing and Optical Flow

In order to compute $D(\boldsymbol{p}_i^{(1)}, \boldsymbol{p}_j^{(1)})$, we first need to determine the disparity maps $\boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(K)} \in \mathbb{R}^{N\times 2}$. In our work, we use the classical optical flow [31], with the MATLAB/C++ implementation by Liu [46].

Running optical flow on $\boldsymbol{g}^{(k)}$'s directly is problematic, because $\boldsymbol{g}^{(k)}$'s are noisy images. Therefore, we pre-filter $\boldsymbol{g}^{(k)}$'s to obtain cleaner images before optical flow. The pre-filtering is done using single-image NLM.

2.2.2 Bias and Variance for Multiview Image NLM

Our proposed adaptive scheme for finding optimal m requires the knowledge of the bias and variance of $\widehat{f}_{i,m}^{(1)}$. To derive the bias and variance of $\widehat{f}_{i,m}^{(1)}$, we substitute $g_i^{(1)} = f_i^{(1)} + n_i^{(1)}$ into (2.5), and we can show that

$$b_{i,m}^{(1)} \overset{\text{def}}{=} \text{Bias}(\widehat{f}_{i,m}^{(1)}) = \sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} \widetilde{W}_{ij}^{(1k)} f_j^{(1)},$$

$$\left(v_{i,m}^{(1)}\right)^2 \overset{\text{def}}{=} \text{Var}(\widehat{f}_{i,m}^{(1)}) = \sum_{k=1}^{K} \sum_{j\in\Omega_{i,m}^{(k)}} \left(\widetilde{W}_{ij}^{(1k)}\right)^2 \sigma^2.$$

We now discuss the conditions in Line 9 of Algorithm 1. For notation simplicity we drop the super-script and let $\widetilde{W}_{ij} = \widetilde{W}_{ij}^{(1k)}$, $\widehat{f}_j = \widehat{f}_j^{(1)}$, $b_{i,m} = b_{i,m}^{(1)}$, and $v_{i,m} = v_{i,m}^{(1)}$.

2.2.3 Condition 1: Denoising Consistency

The intuition of the denoising consistency is that by increasing $m-1$ to m, the changes $\widehat{f}_{i,m'} - \widehat{f}_{i,m}$ for all $m' = 1, \ldots, m-1$ cannot be too large. In other words, we want the algorithm to terminate when the following probabilistic criterion is satisfied:

$$\Pr\left[\left|\widehat{f}_{i,m'} - \widehat{f}_{i,m}\right| \leq \varepsilon\right] \leq \lambda, \quad m' = 1, \ldots, m-1, \tag{2.8}$$

where ε is a threshold, and $\lambda \ll 1$ defines the probability (typically ≈ 0.01). The probability on the left hand side of (2.8) can be determined through the following proposition.

Proposition 1. *The probability inequality* $\Pr\left[\left|\widehat{f}_{i,m'} - \widehat{f}_{i,m}\right| \leq \varepsilon\right] \leq \lambda$ *holds if and only if*

$$\left|\widehat{f}_{i,m'} - \widehat{f}_{i,m}\right| \leq Q^{-1}\left(\frac{1-\lambda}{2}\right)\Delta v_{i,m'}, \tag{2.9}$$

where $Q(\cdot)$ *is the Q-function of standard normal distribution, and* $\Delta v_{i,m'}^2 \overset{def}{=} v_{i,m}^2 - v_{i,m'}^2$.

Proof. First, we define $\Delta\widehat{f}_{i,m'} = \widehat{f}_{i,m} - \widehat{f}_{i,m'}$. Consequently, the corresponding bias and variance can be defined as

$$\Delta b_{i,m'} \overset{\text{def}}{=} b_{i,m} - b_{i,m'} = \sum_{k=1}^{K}\left[\sum_{j\in\Omega_{i,m}^{(k)}} \widetilde{W}_{i,j} f_j - \sum_{j\in\Omega_{i,m'}^{(k)}} \overline{W}_{i,j} f_j\right],$$

$$\Delta v_{i,m'}^2 \overset{\text{def}}{=} v_{i,m}^2 - v_{i,m'}^2 = \sum_{k=1}^{K}\left[\sum_{j\in\Omega_{i,m}^{(k)}} \sigma^2\widetilde{W}_{i,j}^2 - \sum_{j\in\Omega_{i,m'}^{(k)}} \sigma^2\overline{W}_{i,j}^2\right],$$

where $\widetilde{W}$ is the normalized version of W (m non-zero entries), and $\overline{W}$ is the normalized version of W (m' non zero entries).

Since $\widehat{f}_{i,m'} = \sum_{k=1}^{K}\sum_{j\in\Omega_{i,m'}^{(k)}} \widetilde{W}_{i,j}(f_j + n_j)$, it can be shown that $\widehat{f}_{i,m'} \sim \mathcal{N}(b_{i,m'}, v_{i,m'}^2)$ and hence $\Delta\widehat{f}_{i,m'} \sim \mathcal{N}(\Delta b_{i,m'}, \Delta v_{i,m'}^2)$. Substitute this into (2.9) yields

$$\begin{aligned}\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] &= \Pr\left[\Delta\widehat{f}_{i,m'} \geq -\varepsilon\right] - \Pr\left[\Delta\widehat{f}_{i,m'} \geq \varepsilon\right] \\ &= Q\left(-\frac{-\varepsilon - \Delta b_{i,m'}}{\Delta v_{i,m}}\right) - Q\left(-\frac{\varepsilon - \Delta b_{i,m'}}{\Delta v_{i,m'}}\right).\end{aligned}$$

Assuming that $b_{i,m} = b_{i,m'}$, we have $\Delta b_{i,m'} = 0$ and hence

$$\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] = 1 - 2Q\left(\frac{\varepsilon}{\Delta v_{i,m'}}\right).$$

Finally, setting $\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] \leq \lambda$ yields $1 - 2Q\left(\frac{\varepsilon}{\Delta v_{i,m'}}\right) \leq \lambda$, which in turn requires that

$$\varepsilon \leq Q^{-1}\left(\frac{1-\lambda}{2}\right)\Delta v_{i,m'} \overset{\text{def}}{=} \gamma.$$

This implies that $\Pr\left[\left|\Delta\widehat{f}_{i,m'}\right| \leq \varepsilon\right] \leq \lambda$ iff $\left|\Delta\widehat{f}_{i,m'}\right| \leq \gamma$. □

As a consequence of Proposition 1, the algorithm should terminate if $\left|\Delta\widehat{f}_{i,m'}\right| \leq \gamma$. Hence, the optimal m is the smallest integer such that $\left|\Delta\widehat{f}_{i,m'}\right| \geq \gamma$, i.e.,

$$m^* = \underset{m}{\operatorname{argmin}} \left\{\left|\widehat{f}_{i,m} - \widehat{f}_{i,m'}\right| \geq \gamma, \quad 1 \leq m' \leq m\right\}. \tag{2.10}$$

2.2.4 Condition 2: Intersection of Confidence Interval

Our second condition is based on the intuition that m should be increased as long as $\widehat{f}_{i,m}$ has similar confidence interval with $\widehat{f}_{i,m'}$ for $m' = 1, \ldots, m-1$. Since $\widehat{f}_{i,m} \sim \mathcal{N}(b_{i,m}, v_{i,m}^2)$, the confidence interval of $\widehat{f}_{i,m}$ is

$$[\widehat{f}_{i,m} - \tau v_{i,m}, \widehat{f}_{i,m} + \tau v_{i,m}].$$

where τ controls the likelihood that the true value f_i lies in the interval:

$$\alpha_{i,m} \overset{\text{def}}{=} \widehat{f}_{i,m} - \tau v_{i,m} \leq f_i \leq \widehat{f}_{i,m} + \tau v_{i,m} \overset{\text{def}}{=} \beta_{i,m}. \tag{2.11}$$

Therefore, the smallest intersection before having no feasible solution determines the optimal m:

$$m^* = \underset{m}{\operatorname{argmin}} \left\{ m \;:\; \bigcap_{t=1}^{m} [\alpha_{i,t},\ \beta_{i,t}] = \emptyset \right\}. \tag{2.12}$$

2.3 Experimental Results

In this subsection we present some experimental results for multiview image denoising and video denoising using Algorithm 1.

2.3.1 Multi-view Image Denoising

We downloaded 4 sets of images from Middlebury Computer Vision Page http : //vision.middlebury.edu/stereo/. Each set of images consists of 5 views, so that $K = 5$. In our experiments, we add noise of variance $\sigma = 20$ (out of 255).

We compared our proposed algorithm with four existing methods, namely Video NLM [47], BM3D [11], Video BM3D [19] and BM4D [37]. The results are shown in Table 2.1 and snapshots of the images are shown in Figure 2.1.

Table 2.1: PSNR and SSIM (value in the parenthesis) results using Video NLM, BM3D, Video-BM3D, BM4D and the proposed methods for denoising images in Middlebury dataset.

	Barn	Cone	Teddy	Venus
Video NLM [47]	28.95 (0.8363)	28.81 (0.8358)	29.95 (0.8688)	30.55 (0.8941)
BM3D [11]	28.97 (0.8442)	28.90 (0.8443)	30.18 (0.8844)	30.51 (0.9038)
Video BM3D [19]	30.38 (0.8840)	28.42 (0.8353)	29.70 (0.8755)	31.54 (0.9192)
BM4D [37]	28.70 (0.8368)	27.93 (0.8127)	29.28 (0.8575)	29.96 (0.8920)
Ours - Cond 1	**30.67** **(0.9000)**	**30.04** **(0.8905)**	**31.11** **(0.9000)**	**32.00** (0.9189)
Ours - Cond 2	30.48 (0.8946)	29.91 (0.8850)	31.05 (0.8992)	31.91 **(0.9192)**
Ours - Cond 1 (true disparity)	30.74 (0.9034)	30.13 (0.8977)	31.23 (0.9058)	32.08 (0.9224)
Ours - Cond 2 (true disparity)	30.54 (0.8977)	29.98 (0.8915)	31.14 (0.9042)	31.98 (0.9224)

The results suggest the following observations. First, the proposed method (using either conditions) out-performs the competitors by a big margin. Compared with NLM (which uses all patches in the neighborhood), the results indicate that a large number of dissimilar patches, despite having small weights, could severely reduce the denoising quality as a whole. Compared with BM4D (which groups temporal patches at the same locations of consecutive frames as similar patches), the results indicate that if similar patches cannot be grouped accurately across

Table 2.2: PSNR and SSIM values for video denoising.

Video Sequences	BM3D [11]	VBM3D [19]	BM4D [37]	Ours Cond 1
Facility Management	31.48 (0.89)	32.73 (0.91)	31.07 (0.88)	**32.84** **(0.92)**
Jacob School	25.83 (0.84)	**28.12** (**0.91**)	25.23 (0.81)	27.97 **(0.91)**
Market Place	31.69 (0.93)	32.61 **(0.94)**	**32.97** (**0.94**)	32.80 (0.92)
SuperLoop - Big	30.81 **(0.94)**	30.79 (0.93)	30.13 (0.92)	**31.45** (0.93)
SuperLoop - Small	29.04 (0.89)	31.48 (0.92)	**31.95** **(0.93)**	31.30 (0.92)
Voigt Drive 1	29.81 (0.92)	31.32 (0.94)	**31.69** **(0.95)**	31.19 (0.93)
Voigt Drive 2	29.89 (0.91)	31.80 (**0.94**)	**31.98** **(0.94)**	31.48 (0.93)

views, additional patches would deteriorate the denoising performace.

2.3.2 Video Denoising

Our proposed method is primarily designed for multiview denoising where displacement is large. When the displacement is small, our proposed method still works. However, the marginal gain compared to existing video denoising algorithms becomes smaller. Nevertheless, as indicated in Table 2.2 and Figure 2.2, the PSNR values of our proposed method is competitive with existing video denoising algorithms. Averaged over the 7 video sequences we tested, the PSNR improvements of the proposed method over BM3D, VBM3D and BM4D are +1.5dB, +0.05 and +0.57dB, respectively.

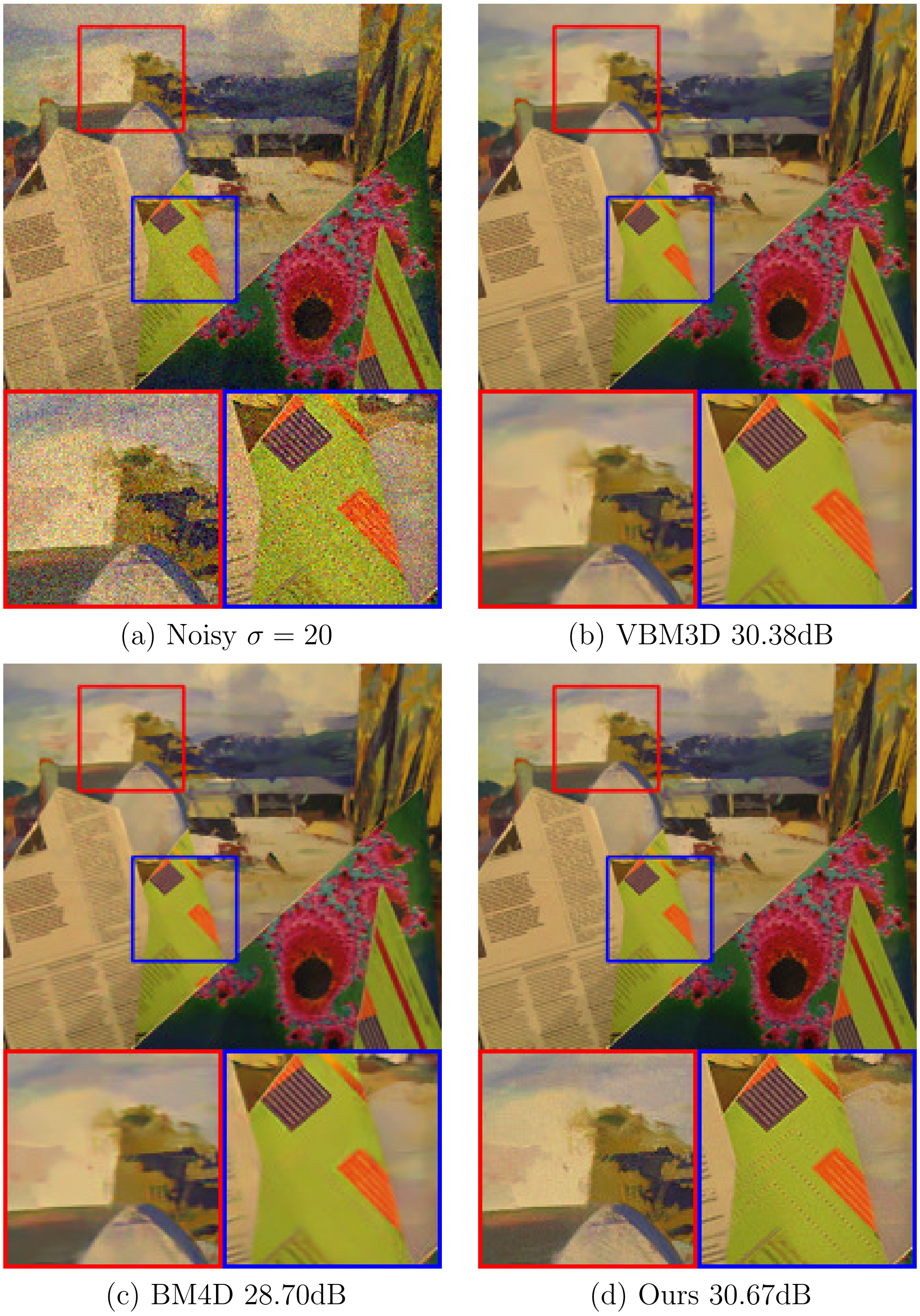
2.4 Conclusion

We presented an adaptive non-local means denoising method for multiview images in which similar patches are carefully chosen according to the local statistics of the estimates. Dissimilar patches are discarded in order to achieve a trade-off between bias and variance. Experimental results showed that the proposed method outperforms state-of-the-art denoising algorithms in multiview denoising settings, and performs competitively in video denoising settings.

2.5 Acknowledgement

This chapter, in part, is a reprint of the following paper:

E. Luo, S. H. Chan, S. Pan, and T. Q. Nguyen, "Adaptive Non-local Means for Multiview Image Denoising: Searching for the Right Patches via a Statistical Approach," in *Proc. IEEE Intl. Conf. Image Process. (ICIP'13)*, pp. 543-547, Sep. 2013.

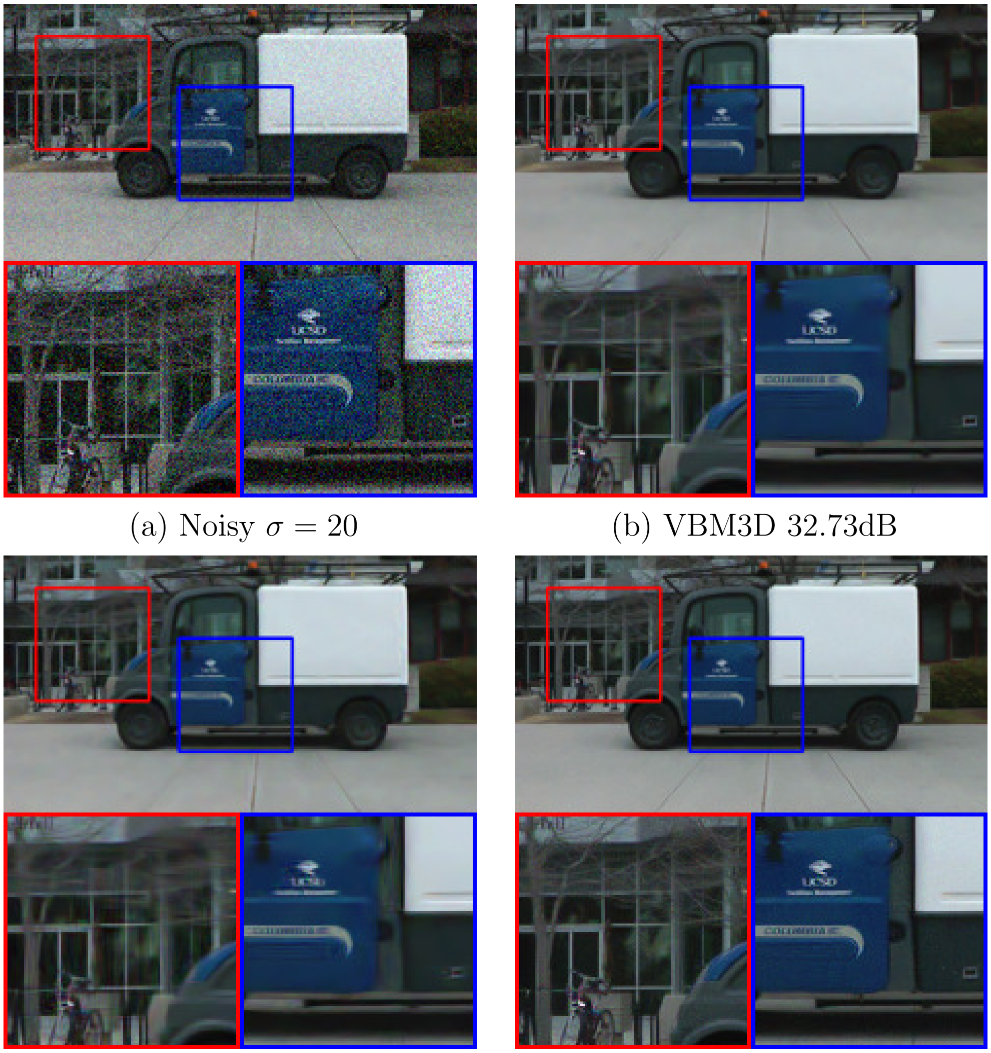

(a) Noisy $\sigma = 20$

(b) VBM3D 30.38dB

(c) BM4D 28.70dB

(d) Ours 30.67dB

Figure 2.1: Multiview denoising using VBM3D, BM4D and the proposed method for the image "Barn".



(a) Noisy $\sigma = 20$

(b) VBM3D 32.73dB

(c) BM4D 31.07dB

(d) Ours 32.84dB

Figure 2.2: Multiview denoising using VBM3D, BM4D and the proposed method for the image “Facility Management”.

Chapter 3

Adaptive Image Denoising by Targeted Databases

3.1 Introduction

In multiview image denoising, to denoise one noisy view, we consider its adjacent noisy views as a "database" of reference images. Equivalently, we may extend this idea to a general database, which consists of external noise-free images. In general, with the increasing amount of high-quality image data available online (*e.g.,* google image, flicker, and instagram), such databases could be easily built. For example, a popular image database ImageNet has hundreds of thousands of clean images. These databases are referred to as *generic* databases in the sense that they contain various different scenes. From a theoretic point of view, if the database covers the entire space of patches, then the complete prior distribution of the patches can be computed. We thereby can achieve optimal denoising with minimum mean squared error (MMSE) [48]. Unfortunately, all databases in reality have finite sizes. A finite generic database, though large in volume, does not necessarily contain enough useful information for the noisy image of interest. Our strategy is to switch from generic database to *targeted* database. Here, a targeted database refers to a database that contains images *relevant* to the noisy image only. With the rapid development of image retrieval technology, retrieving relevant images to form a targeted database for the query image becomes more and more plausible. In addition, as will be illustrated in later parts of this chapter, in many practical scenarios, building targeted databases is less difficult. For example, targeted databases can be built for text images (*e.g.*, newspapers and documents), human faces (under certain conditions), and images captured by multiview camera systems. Other possible scenarios include images of license plates, medical CT and MRI images, and images of landmarks. We term this strategy as database adaptivity.

The concept of using targeted databases has been proposed in various occasions, *e.g.*, [49–54]. However, none of these methods are tailored for image denoising problems. The objective of this chapter is to bridge the gap by addressing the following question:

> Suppose we are given a targeted external database, how should we design a denoising algorithm which can *maximally* utilize the database?

Here, we assume that the reference patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$ are *given*. We emphasize that this assumption is application specific — for the examples we mentioned earlier (*e.g.*, text, multiview, face, etc), the assumption is typically true because these images have relatively less variety in content.

3.1.1 Related Work

When the reference patches are given, the above question perhaps becomes a simple one: We can extend existing internal denoising algorithms in a brute-force way to handle external databases. For example, one can modify existing algorithms, *e.g.*, [11, 12, 17, 55], so that the patches are searched from a database instead of the noisy image. Likewise, one can also treat an external database as a "video" and feed the data to multi-image denoising algorithms, *e.g.*, [19,29,36,56]. However, the problem of these approaches is that the brute force modifications are heuristic. There is no theoretical guarantee of performance.

An alternative response to the above question is to train a statistical prior of the targeted database, *e.g.*, [15, 21, 57–60]. The merit of this approach is that the performance often has theoretical guarantee because the denoising problem can now be formulated as a maximum a posteriori (MAP) estimation. However, the drawback is that many of these methods require a large number of training samples which is not always available in practice.

3.1.2 Contributions and Outline

In view of the above seemingly easy yet challenging question, we propose a new denoising algorithm using targeted external databases. Compared to existing methods, the proposed method achieves better performance and only requires a small number of external images. In this chapter, we offer the following contributions:

1. Generalization of Existing Methods. We propose a generalized framework which encapsulates a number of denoising algorithms. In particular, we show (in Section 3.3) that the proposed group sparsity minimization generalizes both fixed basis and PCA methods. We also show (in Section 3.4) that the proposed local Bayesian MSE solution is a generalization of many spectral operations in existing methods.

2. Improvement Strategies. We propose two improvement strategies for the generalized denoising framework. In Section 3.3.4, we present a patch selection optimization to improve the patch search process. In Section 3.4.4, we present a soft-thresholding and a hard-thresholding method to improve the spectral coefficients learned by the algorithm.

3. Perturbation Analysis (in Section 3.5). We analyze the sensitivity of the proposed denoising framework with respect to the variability of the external database. To our knowledge, similar analysis has not been discussed in the denoising literature.

4. Detailed Proofs. Proofs of the results are presented in the Appendix.

The rest of the chapter is organized as follows. After outlining the design framework in Section 3.2, we present the above contributions in Section 3.3 – 3.5.

Experimental results are discussed in Section 3.6, and concluding remarks are given in Section 3.7.

3.2 Optimal Linear Denoising Filter

The foundation of our proposed method is the classical optimal linear denoising filter design problem [48]. In this section, we give a brief review of the design framework and highlight its limitations.

3.2.1 Optimal Filter

The design of an optimal denoising filter can be posed as follows: Given a noisy patch $\boldsymbol{q} \in \mathbb{R}^d$, and assuming that the noise is i.i.d. Gaussian with zero mean and variance σ^2, we want to find a linear operator $\boldsymbol{A} \in \mathbb{R}^{d\times d}$ such that the estimate $\widehat{\boldsymbol{p}} = \boldsymbol{A}\boldsymbol{q}$ has the minimum mean squared error (MSE) compared to the ground truth $\boldsymbol{p} \in \mathbb{R}^d$. That is, we want to solve the optimization

$$\boldsymbol{A} = \operatorname*{arg\,min}_{\boldsymbol{A}} \; \mathbb{E}\left[\|\boldsymbol{A}\boldsymbol{q} - \boldsymbol{p}\|_2^2\right]. \tag{3.1}$$

Here, we assume that $\boldsymbol{A}$ is symmetric, or otherwise the Sinkhorn-Knopp iteration [61] can be used to symmetrize $\boldsymbol{A}$. Because $\boldsymbol{A}$ is symmetric, one can apply the eigen-decomposition, $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$, where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d] \in \mathbb{R}^{d\times d}$ is the basis matrix and $\boldsymbol{\Lambda} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_d\} \in \mathbb{R}^{d\times d}$ is the diagonal matrix containing the spectral coefficients. With $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$, the optimization problem in (3.1) becomes

$$(\boldsymbol{U}, \boldsymbol{\Lambda}) = \operatorname*{arg\,min}_{\boldsymbol{U},\boldsymbol{\Lambda}} \; \mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q} - \boldsymbol{p}\right\|_2^2\right], \tag{3.2}$$

subject to the constraint that $\boldsymbol{U}$ is an orthonormal matrix.

The joint optimization (3.2) can be solved by noting the following Lemma.

Lemma 1. *Let $\boldsymbol{u}_i$ be the ith column of the matrix $\boldsymbol{U}$, and λ_i be the (i,i)th entry* *of the diagonal matrix* $\boldsymbol{\Lambda}$*. If* $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{\eta}$*, where* $\boldsymbol{\eta} \overset{\text{iid}}{\sim} \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$*, then*

$$\mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q} - \boldsymbol{p}\right\|_2^2\right] = \sum_{i=1}^{d}\left[(1-\lambda_i)^2(\boldsymbol{u}_i^T\boldsymbol{p})^2 + \sigma^2\lambda_i^2\right]. \tag{3.3}$$

Proof. See Appendix A.1 or [62]. □

With Lemma 1, the denoised patch as a consequence of (3.2) is as follows.

Lemma 2. *The denoised patch* $\widehat{\boldsymbol{p}}$ *using the optimal* $\boldsymbol{U}$ *and* $\boldsymbol{\Lambda}$ *of* (3.2) *is*

$$\widehat{\boldsymbol{p}} = \boldsymbol{U}\left(\text{diag}\left\{\frac{\|\boldsymbol{p}\|^2}{\|\boldsymbol{p}\|^2 + \sigma^2}, 0, \ldots, 0\right\}\right)\boldsymbol{U}^T\boldsymbol{q},$$

where $\boldsymbol{U}$ *is any orthonormal matrix with the first column* $\boldsymbol{u}_1 = \boldsymbol{p}/\|\boldsymbol{p}\|_2$*, and* diag $\{\cdot\}$ *denotes the diagonalization operator.*

Proof. See Appendix A.2. □

Lemma 2 states that if hypothetically we are given the ground truth $\boldsymbol{p}$, the optimal denoising process is to first project the noisy observation $\boldsymbol{q}$ onto the subspace spanned by $\boldsymbol{p}$, perform a Wiener shrinkage $\|\boldsymbol{p}\|^2/(\|\boldsymbol{p}\|^2 + \sigma^2)$, and then re-project the shrinkage coefficients to obtain the denoised estimate. However, since in reality we never have access to the ground truth $\boldsymbol{p}$, this optimal result is not achievable.

3.2.2 Problem Statement

Since the oracle optimal filter is not achievable in practice, the question becomes whether it is possible to find a surrogate solution that does not require the ground truth $\boldsymbol{p}$.

To answer this question, it is helpful to separate the joint optimization (3.2) by first fixing $\boldsymbol{U}$ and minimize the MSE with respect to $\boldsymbol{\Lambda}$. In this case, the optimal filter can be found by minimizing (3.3) with respect to each λ_i, which amounts to determining the root of the derivative of (3.3):

$$\frac{\partial}{\partial \lambda_i} \sum_{i=1}^{n} (1-\lambda_i)^2 (\boldsymbol{u}_i^T \boldsymbol{p})^2 + \sigma^2 \lambda_i^2 = 0.$$

One can show that (3.3) achieves the minimum when

$$\lambda_i = \frac{(\boldsymbol{u}_i^T \boldsymbol{p})^2}{(\boldsymbol{u}_i^T \boldsymbol{p})^2 + \sigma^2}, \tag{3.4}$$

in which the minimum MSE estimator is given by

$$\widehat{\boldsymbol{p}} = \boldsymbol{U} \left(\text{diag} \left\{ \frac{(\boldsymbol{u}_1^T \boldsymbol{p})^2}{(\boldsymbol{u}_1^T \boldsymbol{p})^2 + \sigma^2}, \ldots, \frac{(\boldsymbol{u}_d^T \boldsymbol{p})^2}{(\boldsymbol{u}_d^T \boldsymbol{p})^2 + \sigma^2} \right\} \right) \boldsymbol{U}^T \boldsymbol{q}, \tag{3.5}$$

where $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d\}$ are the columns of $\boldsymbol{U}$.

Inspecting (3.5), we identify two parts of the problem:

1. Determine $\boldsymbol{U}$. The choice of $\boldsymbol{U}$ plays a critical role in the denoising performance. In literature, $\boldsymbol{U}$ are typically chosen as the FFT or the DCT bases [11,20]. In [12,22,23], the PCA bases of various data matrices are proposed. However, the optimality of these bases is not fully understood.

2. Determine $\boldsymbol{\Lambda}$. Even if $\boldsymbol{U}$ is fixed, the optimal $\boldsymbol{\Lambda}$ in (3.4) still depends on the unknown ground truth $\boldsymbol{p}$. In [11], $\boldsymbol{\Lambda}$ is determined by hard-thresholding a stack of DCT coefficients or applying an empirical Wiener filter constructed from a first-pass estimate. In [22], $\boldsymbol{\Lambda}$ is formed by the PCA coefficients of a set of relevant noisy patches. Again, it is unclear which of these is optimal.

Motivated by the problems about $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$, in the following two sections we present our proposed method for each of these problems. We discuss its relationship to prior works, and present ways to further improve it.

3.3 Determine $\boldsymbol{U}$

In this section we present our proposed method to determine the basis matrix $\boldsymbol{U}$ and show that it is a generalization of a number of existing denoising algorithms. We also discuss ways to improve $\boldsymbol{U}$.

3.3.1 Database Reduction by k Nearest Neighbors

Our first task to determine the basis matrix $\boldsymbol{U}$ is to reduce the database. This step is necessary because while all patches in the database come from images that contain similar content to the noisy image, a big portion of the patches are still not useful to denoise a particular noisy patch.

The database reduction is performed by selecting the k most similar patches to the noisy patch (*i.e.*, the one to be denoised). The similarity is measured based on the ℓ_2 distance between the noisy patch $\boldsymbol{q}$ and the database $\{\boldsymbol{p}_j\}_{j=1}^n$, where $n > k$, as

$$d(\boldsymbol{q}, \boldsymbol{p}_j) = \|\boldsymbol{q} - \boldsymbol{p}_j\|_2, \quad \text{for } j = 1, \ldots, n.$$

Effectively, this amounts to searching k nearest neighbors (kNN) from a set of n data points.

The kNN procedure is effective to select a subset of the database. However, it has the drawback that under the ℓ_2 distance, some of the k selected patches could be irrelevant. We will address this issue more thoroughly in Section 3.3.4 by discussing methods to improve the robustness of the kNN.

3.3.2 Group Sparsity

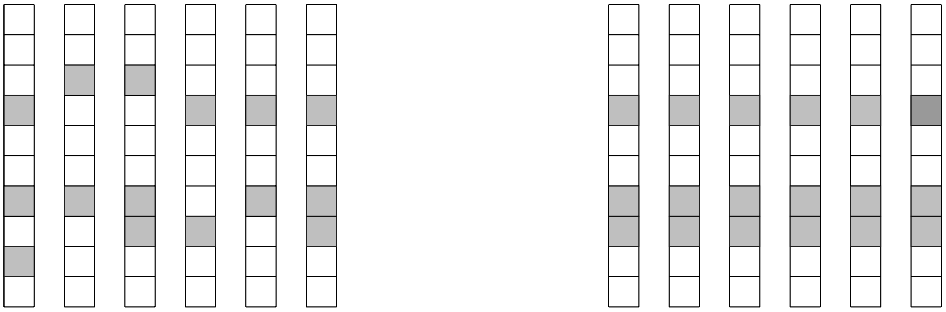
Given $\{\boldsymbol{p}_j\}_{j=1}^k$ from the kNN search, we postulate that a good projection matrix $\boldsymbol{U}$ should be the one that makes the projected vectors $\{\boldsymbol{U}^T\boldsymbol{p}_j\}_{j=1}^k$ similar in both *magnitude* and *location*. This hypothesis follows from the observation that since $\{\boldsymbol{p}_j\}_{j=1}^k$ have small ℓ_2 distances from $\boldsymbol{q}$, it must hold that any $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ (hence $\boldsymbol{U}^T\boldsymbol{p}_i$ and $\boldsymbol{U}^T\boldsymbol{p}_j$) in the set should also be similar.

In addition to being self-similar, we require each projected vector $\boldsymbol{U}^T\boldsymbol{p}_j$ to contain as few non-zero entries as possible, *i.e.*, *sparse*. The reason is related to the shrinkage step to be discussed in Section 3.4, because a vector of few non-zero coefficients has higher energy concentration and hence is more effective for denoising.

In order to satisfy these two requirements, we propose to consider the idea of *group sparsity*[1], which is characterized by the matrix $\ell_{1,2}$ norm, defined as

$$\|\boldsymbol{X}\|_{1,2} \overset{\text{def}}{=} \sum_{i=1}^{d} \|\boldsymbol{x}_i\|_2, \tag{3.6}$$

for any matrix $\boldsymbol{X} \in \mathbb{R}^{d\times k}$, where $\boldsymbol{x}_i$ is the ith row of a matrix $\boldsymbol{X}$. In words, a small $\|\boldsymbol{X}\|_{1,2}$ makes sure that $\boldsymbol{X}$ has few non-zero entries, and the non-zero entries are located similarly in each column. A pictorial illustration is shown in Figure 3.1.



(a) sparse

(b) group sparse

Figure 3.1: Comparison between sparsity (where columns are sparse, but do not coordinate) and group sparsity (where all columns are sparse with similar locations).

Going back to our problem, we propose to minimize the $\ell_{1,2}$-norm of the matrix $\boldsymbol{U}^T\boldsymbol{P}$:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{minimize}} \quad & \|\boldsymbol{U}^T\boldsymbol{P}\|_{1,2} \\ \text{subject to} \quad & \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}, \end{aligned} \tag{3.7}$$

where $\boldsymbol{P} \overset{\text{def}}{=} [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k]$. Here, the equality constraint ensures that $\boldsymbol{U}$ is orthonormal. Thus, the solution of (3.7) is the projection matrix that generates the most group sparse vector.

An interesting fact of this problem is that the solution is identical to the classical principal component analysis (PCA) result, which is given in the following

[1] Group sparsity was first proposed by Cotter et al. for group sparse reconstruction [63] and later used by Mairal et al. for denoising [21], but towards a different end from the method presented in this paper.

lemma.

Lemma 3. *The solution to* (3.7) *is that*

$$[\boldsymbol{U}, \boldsymbol{S}] = eig(\boldsymbol{P}\boldsymbol{P}^T), \tag{3.8}$$

where $\boldsymbol{S}$ *is the corresponding eigenvalue matrix.*

Proof. See Appendix A.3. □

Remark 1. *In practice, we note that the* k *reference patches might have deviations in terms of similarity with* $\boldsymbol{q}$*. Thus, we improve the data matrix* $\boldsymbol{P}$ *by introducing a diagonal weight matrix*

$$\boldsymbol{W} = \frac{1}{Z} \operatorname{diag}\left\{ e^{-\|\boldsymbol{q}-\boldsymbol{p}_1\|^2/h^2}, \ldots, e^{-\|\boldsymbol{q}-\boldsymbol{p}_k\|^2/h^2} \right\}, \tag{3.9}$$

for some user tunable parameter h *and normalization constant* $Z \overset{def}{=} \mathbf{1}^T \boldsymbol{W} \mathbf{1}$*. In this case,* (3.7) *becomes*

$$\begin{array}{ll} \underset{\boldsymbol{U}}{\text{minimize}} & \|\boldsymbol{U}^T \boldsymbol{P} \boldsymbol{W}^{1/2}\|_{1,2} \\ \text{subject to} & \boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}, \end{array} \tag{3.10}$$

of which the solution is given by

$$[\boldsymbol{U}, \boldsymbol{S}] = eig(\boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T). \tag{3.11}$$

3.3.3 Relationship to Prior Works

The fact that (3.11) is the solution to a group sparsity minimization problem allows us to understand the performance of a number of existing denoising algorithms to some extent.

3.3.3.1 BM3D [11]

It is perhaps a misconception that the underlying principle of BM3D is to enforce sparsity of the 3-dimensional data volume (which we shall call it a 3-way tensor). However, what BM3D enforces is the *group sparsity* of the slices of the tensor, not the sparsity of the tensor.

To see this, we note that the 3-dimensional transforms in BM3D are separable (*e.g.*, DCT2 + Haar in its default setting). Thus, unless the reference patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$ are highly dissimilar, the DCT2 coefficients will be similar in *both* magnitude and location. That means if we fix a location and trace the DCT2 coefficients along the third axis, the signal we observe is almost flat. Hence, applying the Haar transform returns a sparse vector. Clearly, such sparsity is based on the stationarity of the DCT2 coefficients along the third axis. In essence, this is equivalent to being group sparse.

3.3.3.2 HOSVD [24]

The true tensor sparsity can only be utilized by the high order singular value decomposition (HOSVD), which is recently studied in [24]. Let $\mathcal{P} \in \mathbb{R}^{\sqrt{d}\times\sqrt{d}\times k}$ be the tensor by stacking the patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$ into a 3-dimensional array. HOSVD seeks three orthonormal matrices $\boldsymbol{U}^{(1)} \in \mathbb{R}^{\sqrt{d}\times\sqrt{d}}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{\sqrt{d}\times\sqrt{d}}$, $\boldsymbol{U}^{(3)} \in \mathbb{R}^{k\times k}$ and an array $\mathcal{S} \in \mathbb{R}^{\sqrt{d}\times\sqrt{d}\times k}$, such that

$$\mathcal{S} = \mathcal{P} \times_1 \boldsymbol{U}^{(1)^T} \times_2 \boldsymbol{U}^{(2)^T} \times_3 \boldsymbol{U}^{(3)^T},$$

where $\times_k$ denotes a tensor mode-k multiplication [64].

HOSVD ignores the fact that image patches tend to be group sparse instead of being tensor sparse. Consequently, its performance is worse than BM3D, as we observe in [24].

3.3.3.3 Shape-adaptive BM3D [20]

As a variation of BM3D, SA-BM3D groups similar patches according to a mask defined by $\boldsymbol{q}$. The mask modifies the standard ℓ_2 distance between patches to a weighted ℓ_2 distance by masking out irrelevant sub-regions.

Shape-adaptive BM3D can be easily generalized in our proposed framework by defining an additional weight matrix $\boldsymbol{W}_s \in \mathbb{R}^{d\times d}$ (where the subscript s denotes a *spatial* weight) and consider the weighted data

$$\overline{\boldsymbol{P}} = \boldsymbol{W}_s^{1/2} \boldsymbol{P} \boldsymbol{W}^{1/2},$$

where $\boldsymbol{W} \in \mathbb{R}^{k \times k}$ is defined in (3.9). Here the matrix $\boldsymbol{W}_s$ is used to control the relative emphasis of each pixel in the spatial coordinate. For the rest of the chapter, we let $\boldsymbol{W}_s = \boldsymbol{I}$ to improve computational efficiency.

3.3.3.4 BM3D-PCA [12] and LPG-PCA [22]

The idea of both BM3D-PCA and LPG-PCA is that given $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$, $\boldsymbol{U}$ is determined as the principal components of $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k]$. Incidentally, such approaches arrive at the same result as (3.11), *i.e.*, the principal components are indeed the solution of a group sparse minimization. However, the key of using the group sparsity is not noticed in [12] and [22]. This provides additional theoretical justifications for both methods.

3.3.4 Improvement: Patch Selection Refinement

The optimization problem (3.10) suggests that the $\boldsymbol{U}$ computed from (3.11) is the optimal basis with respect to the reference patches $\{\boldsymbol{p}_j\}_{j=1}^k$. However, one issue that remains is how to improve the selection of the k patches.

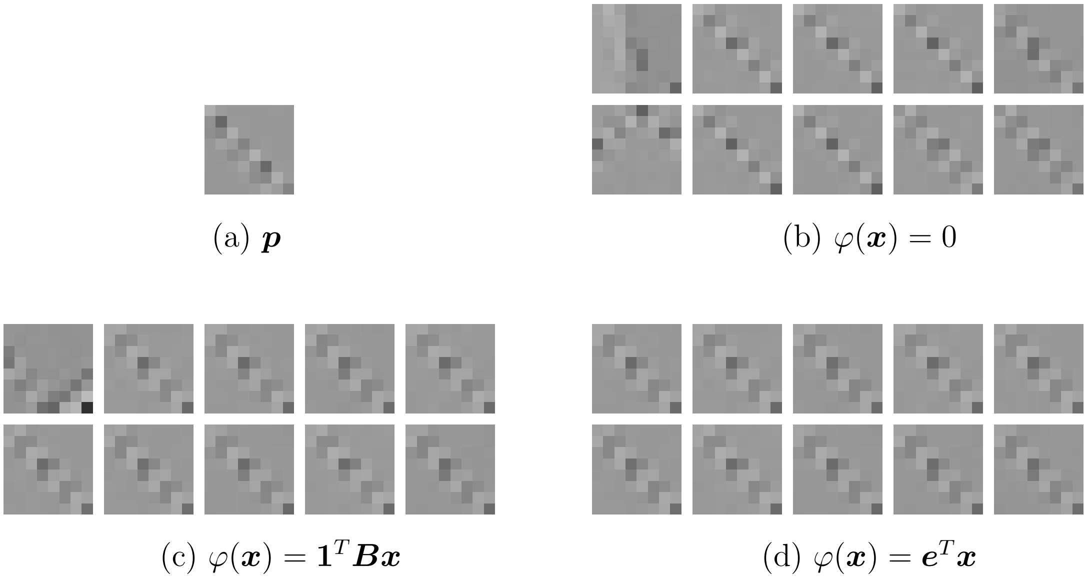
3.3.4.1 Patch Selection as Linear Programming

To facilitate the discussion of our proposed scheme, it is useful to revisit the kNN search from an optimization perspective.

It is not difficult to see that the kNN search can be formulated as the following optimization problem

$$\begin{array}{ll} \underset{\boldsymbol{x} \in \{0,1\}^n}{\text{minimize}} & \sum_{j=1}^n x_j \|\boldsymbol{q} - \boldsymbol{p}_j\|_2 \\ \text{subject to} & \sum_{j=1}^n x_j = k. \end{array} \tag{3.12}$$

Here, the optimization variables $x_j \in \{0,\, 1\}$ form a vector of indicators. If $x_j = 1$,



(a) $\boldsymbol{p}$

(b) $\varphi(\boldsymbol{x}) = 0$

(c) $\varphi(\boldsymbol{x}) = \mathbf{1}^T\boldsymbol{B}\boldsymbol{x}$

(d) $\varphi(\boldsymbol{x}) = \boldsymbol{e}^T\boldsymbol{x}$

Figure 3.2: Refined patch matching results: (a) ground truth, (b) 10 best reference patches using $\boldsymbol{q}$ ($\sigma = 50$), (c) 10 best reference patches using $\varphi(\boldsymbol{x}) = \mathbf{1}^T\boldsymbol{B}\boldsymbol{x}$ (where $\tau = 1/(2n)$), (d) 10 best reference patches using $\varphi(\boldsymbol{x}) = \boldsymbol{e}^T\boldsymbol{x}$ (where $\tau = 1$).

then the corresponding $\boldsymbol{p}_j$ should be selected. Thus, by minimizing $\sum_{j=1}^{n} x_j \|\boldsymbol{q} - \boldsymbol{p}_j\|_2$ we obtain the k nearest neighbors of $\boldsymbol{q}$.

Problem (3.12) is a combinatorial problem as it seeks for one out of the $\binom{n}{k}$ configurations that minimizes the objective. However, a close inspection reveals that such a combinatorial search is unnecessary. In fact, (3.12) is equivalent to a relaxed convex optimization (more specifically, a linear programming problem)

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} & \boldsymbol{x}^T\mathbf{1} = k, \quad 0 \leq \boldsymbol{x} \leq 1, \end{array} \tag{3.13}$$

where we define $\boldsymbol{c} = [c_1, \cdots, c_n]^T$ with $c_j \overset{\text{def}}{=} \|\boldsymbol{q} - \boldsymbol{p}_j\|_2$. To see the equivalence between (3.12) and (3.13), we consider a simple case where $n = 2$ and $k = 1$. In this case, the constraints $\boldsymbol{x}^T\mathbf{1} = k$ and $0 \leq \boldsymbol{x} \leq 1$ form a closed line segment in the positive quadrant. Since the objective function $\boldsymbol{c}^T\boldsymbol{x}$ is linear, the optimal point must be at one of the vertices of the line segment, which is either $\boldsymbol{x} = [0, 1]^T$, or $\boldsymbol{x} = [1, 0]^T$. Thus, by checking which of c_1 or c_2 is smaller, we can determine the optimal solution. A similar argument holds for higher dimensions, and this justifies our claim.

Remark 2. *Because the optimal solution must be a vertex of the polytope defined by the constraints* $\boldsymbol{x}^T\mathbf{1} = k$ *and* $0 \leq \boldsymbol{x} \leq 1$, (3.13) *can be solved efficiently by locating the* k *smallest entries in* $\boldsymbol{c}$, *eliminating the need of an iterative linear programming solver.*

3.3.4.2 Regularization by Cross Similarity

Our proposed patch selection scheme is to modify (3.13) by adding an appropriate penalty term to the objective function:

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \boldsymbol{c}^T\boldsymbol{x} + \tau\varphi(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{x}^T\mathbf{1} = k, \quad 0 \leq \boldsymbol{x} \leq 1, \end{array} \tag{3.14}$$

where $\varphi(\boldsymbol{x})$ is the penalty function and $\tau > 0$ is a parameter. In our work we present two possible choices of $\varphi(\boldsymbol{x})$.

The first choice of $\varphi(\boldsymbol{x})$ is to consider $\varphi(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x}$, where $\boldsymbol{B} \in \mathbb{R}^{n\times n}$ is a symmetric matrix with $B_{ij} \overset{\text{def}}{=} \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2$. Writing (3.14) explicitly, we see that the optimization problem (3.12) becomes

$$\underset{0\leq\boldsymbol{x}\leq 1,\, \boldsymbol{x}^T\mathbf{1}=k}{\text{minimize}} \quad \sum_j x_j\|\boldsymbol{q} - \boldsymbol{p}_j\|_2 + \tau\sum_{i,j} x_i x_j \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2. \tag{3.15}$$

The penalized problem (3.15) suggests that the optimal k reference patches should not be determined merely from $\|\boldsymbol{q} - \boldsymbol{p}_j\|_2$ (which could be problematic due to the noise present in $\boldsymbol{q}$). Instead, a good reference patch should also be similar to all other patches that are selected. The cross similarity term $x_i x_j\|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2$ provides a way for such measure. This shares some similarities to the patch ordering concept proposed by Cohen and Elad [55]. The difference is that the patch ordering proposed in [55] is a shortest path problem that tries to organize the noisy patches, whereas ours is to solve a regularized optimization.

Problem (3.15) is in general not convex because the matrix $\boldsymbol{B}$ is not positive semidefinite. One way to relax the formulation is to consider $\varphi(\boldsymbol{x}) = \mathbf{1}^T\boldsymbol{B}\boldsymbol{x}$. Geometrically, the solution of using $\varphi(\boldsymbol{x}) = \mathbf{1}^T\boldsymbol{B}\boldsymbol{x}$ tends to identify patches that are close to *sum* of all other patches in the set. In many cases, this is similar to $\varphi(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x}$ which finds patches that are similar to every *individual* patch in the set. In practice, we find that the difference between $\varphi(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{B}\boldsymbol{x}$ and $\varphi(\boldsymbol{x}) = \mathbf{1}^T\boldsymbol{B}\boldsymbol{x}$ is not significant. Thus, for computational efficiency we choose $\varphi(\boldsymbol{x}) = \mathbf{1}^T\boldsymbol{B}\boldsymbol{x}$.

3.3.4.3 Regularization by First-pass Estimate

The second choice of $\varphi(\boldsymbol{x})$ is based on a *first-pass estimate* $\overline{\boldsymbol{p}}$ using some denoising algorithms, for example, BM3D or the proposed method without this patch selection step. In this case, by defining $e_j \overset{\text{def}}{=} \|\overline{\boldsymbol{p}} - \boldsymbol{p}_j\|_2$ we consider the penalty function $\varphi(\boldsymbol{x}) = \boldsymbol{e}^T\boldsymbol{x}$, where $\boldsymbol{e} = [e_1, \cdots, e_n]^T$. This implies the following optimization problem

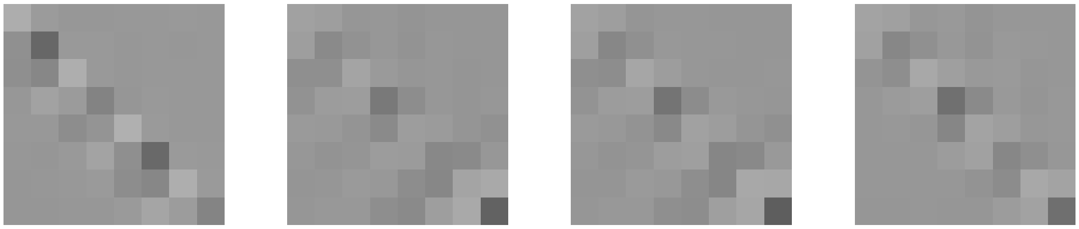
$$\underset{0 \leq \boldsymbol{x} \leq 1,\, \boldsymbol{x}^T\mathbf{1} = k}{\text{minimize}} \quad \textstyle\sum_j x_j \|\boldsymbol{q} - \boldsymbol{p}_j\|_2 + \tau \sum_j x_j \|\overline{\boldsymbol{p}} - \boldsymbol{p}_j\|_2, \tag{3.16}$$

which takes the same form as (3.13). Therefore, (3.16) can be solved in closed form by identifying the k smallest entries of the vector $\boldsymbol{c} + \tau\boldsymbol{e}$.

The interpretation of (3.16) is straight forward: The linear combination of $\|\boldsymbol{q} - \boldsymbol{p}_j\|_2$ and $\|\overline{\boldsymbol{p}} - \boldsymbol{p}_j\|_2$ shows a competition between the noisy patch $\boldsymbol{q}$ and the first-pass estimate $\overline{\boldsymbol{p}}$. In most of the common scenarios, $\|\boldsymbol{q} - \boldsymbol{p}_j\|_2$ is preferred when noise level is low, whereas $\overline{\boldsymbol{p}}$ is preferred when noise level is high. This in turn suggests a plausible choice τ, by which empirically we find that choosing $\tau = 0.01$ when $\sigma < 30$ and $\tau = 1$ when $\sigma > 30$ is a good balance between the performance and generality.

3.3.4.4 Comparisons

To demonstrate the effectiveness of the two proposed patch selection steps, we consider a ground truth (clean) patch shown in Figure 3.2 (a). From a pool of $n = 200$ reference patches, we apply an exhaustive search algorithm to choose



$\boldsymbol{p}$ Ground Truth

$\varphi(\boldsymbol{x}) = 0$ 28.29 dB

$\varphi(\boldsymbol{x}) = \boldsymbol{1}^T\boldsymbol{B}\boldsymbol{x}$ 28.50 dB

$\varphi(\boldsymbol{x}) = \boldsymbol{e}^T\boldsymbol{x}$ 29.30 dB

Figure 3.3: Denoising results: A ground truth patch cropped from an image, and the denoised patches of using different improvement schemes. Noise standard deviation is $\sigma = 50$.

$k = 40$ patches that best match with the noisy observation $\boldsymbol{q}$, where the first 10 patches are shown in Figure 3.2 (b). The results of the two selection refinement methods are shown in Figure 3.2 (c)-(d), where in both cases the parameter τ is adjusted for the best performance. For the case of $\varphi(\boldsymbol{x}) = \boldsymbol{1}^T\boldsymbol{B}\boldsymbol{x}$, we set $\tau = 1/(200n)$ when $\sigma < 30$ and $\tau = 1/(2n)$ when $\sigma > 30$. For the case of $\varphi(\boldsymbol{x}) = \boldsymbol{e}^T\boldsymbol{x}$, we use the denoised result of BM3D as the first-pass estimate $\overline{\boldsymbol{p}}$, and set $\tau = 0.01$ when $\sigma < 30$ and $\tau = 1$ when $\sigma > 30$. The results show that the PSNR increases from 28.29 dB to 28.50 dB if we use $\varphi(\boldsymbol{x}) = \boldsymbol{1}^T\boldsymbol{B}\boldsymbol{x}$, and further increases to 29.30 dB if we use $\varphi(\boldsymbol{x}) = \boldsymbol{e}^T\boldsymbol{x}$. The full performance comparison is shown in Figure 3.4, where we show the PSNR curve for a range of noise levels of an image.

From the results of Figure 3.3 and Figure 3.4, we observe that in general $\varphi(\boldsymbol{x}) = \boldsymbol{e}^T\boldsymbol{x}$ performs better than $\varphi(\boldsymbol{x}) = \boldsymbol{1}^T\boldsymbol{B}\boldsymbol{x}$. This suggests that the first-pass estimate tends to be a better prior than the cross similarity, as cross similarity depends on the database (which does not adapt to the noisy data), whereas first-pass estimate depends on the noisy image. A thorough theoretical analysis of the performance is an interesting direction for future work.

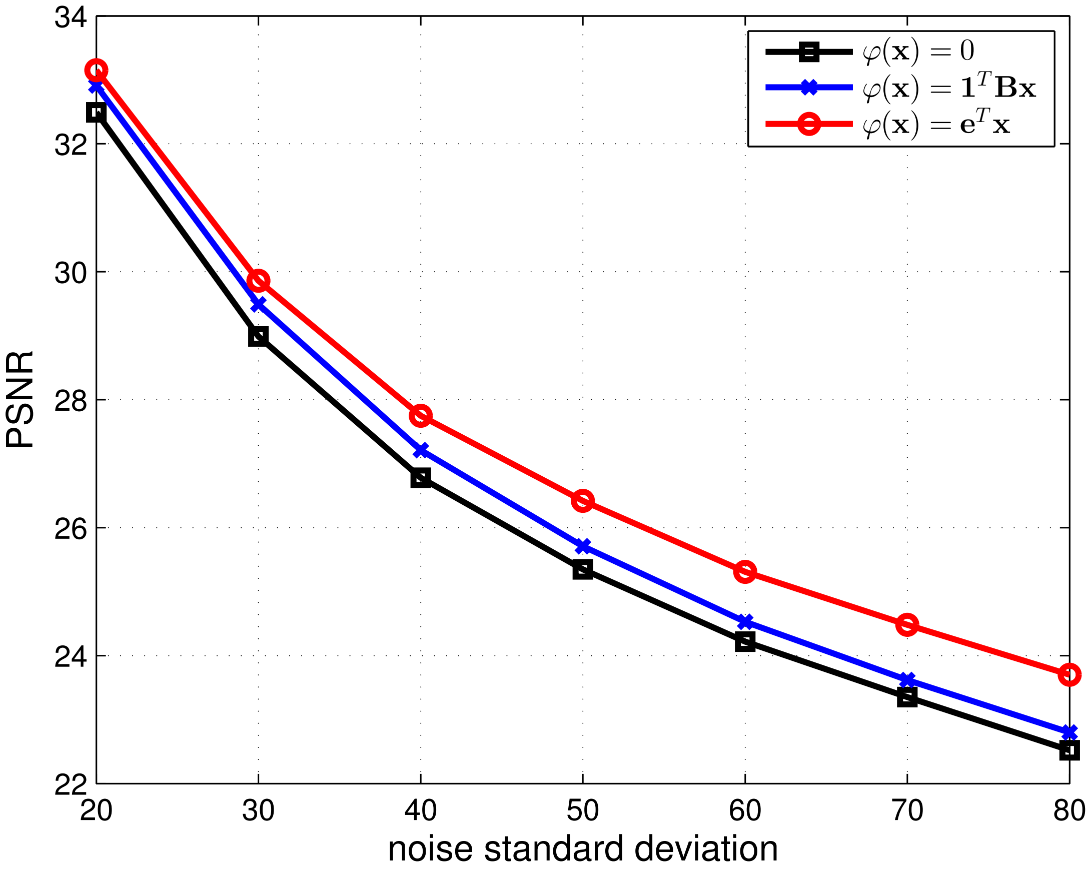


Figure 3.4: Denoising results of three patch selection improvement schemes. The PSNR value is computed from a 432 × 381 image, containing 4536 patches (overlapped by 2 pixels horizontally and vertically).

3.4 Determine $\boldsymbol{\Lambda}$

In this section we present our proposed method to determine $\boldsymbol{\Lambda}$ for a fixed $\boldsymbol{U}$. Our proposed method is based on the concept of a Bayesian MSE estimator.

3.4.1 Bayesian MSE Estimator

Recall that the noisy patch is related to the latent clean patch as $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{\eta}$, where $\boldsymbol{\eta} \overset{\text{iid}}{\sim} \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ denotes the noise. Therefore, the conditional distribution of $\boldsymbol{q}$ given $\boldsymbol{p}$ is

$$f(\boldsymbol{q} \,|\, \boldsymbol{p}) = \mathcal{N}(\boldsymbol{p}, \, \sigma^2 \boldsymbol{I}). \tag{3.17}$$

Assuming that the prior distribution $f(\boldsymbol{p})$ is known, it is natural to consider the Bayesian mean squared error (BMSE) between the estimate $\widehat{\boldsymbol{p}} \overset{\text{def}}{=} \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q}$ and the ground truth $\boldsymbol{p}$:

$$\text{BMSE} \overset{\text{def}}{=} \mathbb{E}_{\boldsymbol{p}}\left[\mathbb{E}_{\boldsymbol{q}|\boldsymbol{p}}\left[\|\widehat{\boldsymbol{p}} - \boldsymbol{p}\|_2^2 \;\middle|\; \boldsymbol{p}\right]\right]. \tag{3.18}$$

Here, the subscripts remark the distributions under which the expectations are taken.

The BMSE defined in (3.18) suggests that the optimal $\boldsymbol{\Lambda}$ should be the minimizer of the optimization problem

$$\boldsymbol{\Lambda} = \underset{\boldsymbol{\Lambda}}{\arg\min}\;\; \mathbb{E}_{\boldsymbol{p}}\left[\mathbb{E}_{\boldsymbol{q}|\boldsymbol{p}}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q} - \boldsymbol{p}\right\|_2^2 \;\middle|\; \boldsymbol{p}\right]\right]. \tag{3.19}$$

In the next subsection we discuss how to solve (3.19).

3.4.2 Localized Prior from the Targeted Database

Minimizing BMSE over $\boldsymbol{\Lambda}$ involves knowing the prior distribution $f(\boldsymbol{p})$. However, in general, the exact form of $f(\boldsymbol{p})$ is never known. This leads to many popular models in the literature, *e.g.*, Gaussian mixture model [60], the field of expert model [58,65], and the expected patch log-likelihood model (EPLL) [59,66].

One common issue of all these models is that the prior $f(\boldsymbol{p})$ is built from a generic database of patches. In other words, the $f(\boldsymbol{p})$ models *all* patches in the database. As a result, $f(\boldsymbol{p})$ is often a high dimensional distribution with complicated shapes.

In our targeted database setting, the difficult prior modeling becomes a much simpler task. The reason is that while the shape of the distribution $f(\boldsymbol{p})$ is still unknown, the subsampled reference patches (which are few but highly representative) could be well approximated as samples drawn from a single Gaussian centered around some mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. Therefore, by appropriately estimating $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of this *localized* prior, we can derive the optimal $\boldsymbol{\Lambda}$ as given by the following Lemma:

Lemma 4. *Let* $f(\boldsymbol{q}\,|\,\boldsymbol{p}) = \mathcal{N}(\boldsymbol{p}, \sigma^2\boldsymbol{I})$*, and let* $f(\boldsymbol{p}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ *for any vector* $\boldsymbol{\mu}$

and matrix $\boldsymbol{\Sigma}$*, then the optimal* $\boldsymbol{\Lambda}$ *that minimizes* (3.18) *is*

$$\boldsymbol{\Lambda} = \frac{\mathrm{diag}\left\{\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U}\right\}}{\mathrm{diag}\left\{\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U} + \sigma^2\boldsymbol{I}\right\}}, \tag{3.20}$$

where the division operation is element-wise.

Proof. See Appendix A.4. □

To specify $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, we let

$$\boldsymbol{\mu} = \sum_{j=1}^{k} w_j\boldsymbol{p}_j, \quad \boldsymbol{\Sigma} = \sum_{j=1}^{k} w_j(\boldsymbol{p}_j - \boldsymbol{\mu})(\boldsymbol{p}_j - \boldsymbol{\mu})^T, \tag{3.21}$$

where w_j is the jth diagonal entry of $\boldsymbol{W}$ defined in (3.9). Intuitively, an interpretation of (3.21) is that $\boldsymbol{\mu}$ is the non-local mean of the reference patches. However, the more important part of (3.21) is $\boldsymbol{\Sigma}$, which measures the *uncertainty* of the reference patches with respect to $\boldsymbol{\mu}$. This uncertainty measure makes some fundamental improvements to existing methods which will be discussed in Section IV-C.

We note that Lemma 4 holds even if $f(\boldsymbol{p})$ is not Gaussian. In fact, for any distribution $f(\boldsymbol{p})$ with the first cumulant $\boldsymbol{\mu}$ and the second cumulant $\boldsymbol{\Sigma}$, the optimal solution in (A.10) still holds. This links our work to the classical linear minimum MSE (LMMSE) estimation [67].

From a computational perspective, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ defined in (3.21) lead to a very efficient implementation as illustrated by the following lemma.

Lemma 5. *Using* $\boldsymbol{\mu}$ *and* $\boldsymbol{\Sigma}$ *defined in* (3.21)*, the optimal* $\boldsymbol{\Lambda}$ *is given by*

$$\boldsymbol{\Lambda} = \frac{\boldsymbol{S}}{\boldsymbol{S} + \sigma^2\boldsymbol{I}}, \tag{3.22}$$

where $[\boldsymbol{U}, \boldsymbol{S}] = eig(\boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T)$ *is the eigen-decomposition of the weighted matrix* $\boldsymbol{P}\boldsymbol{W}^{1/2}$.

Proof. See Appendix A.5. □

Combining Lemma 5 with Lemma 3, we observe that for any set of reference patches $\{\boldsymbol{p}_j\}_{j=1}^k$, $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ can be determined *simultaneously* through the eigen-decomposition of $\boldsymbol{PWP}^T$. Therefore, we arrive at the overall algorithm shown in Algorithm 2.

Algorithm 2 Proposed Algorithm

```
Input: Noisy patch $\boldsymbol{q}$, noise variance $\sigma^2$, and clean reference patches $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k$
Output: Estimate $\widehat{\boldsymbol{p}}$
Learn $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$
    • Form data matrix $\boldsymbol{P}$ and weight matrix $\boldsymbol{W}$
    • Compute eigen-decomposition $[\boldsymbol{U}, \boldsymbol{S}] = \text{eig}(\boldsymbol{PWP}^T)$
    • Compute $\boldsymbol{\Lambda} = \frac{\boldsymbol{S}}{\boldsymbol{S}+\sigma^2\boldsymbol{I}}$. (division is element-wise)
Denoise: $\widehat{\boldsymbol{p}} = \boldsymbol{U\Lambda U}^T\boldsymbol{q}$.
```

3.4.3 Relationship to Prior Works

It is interesting to note that many existing patch-based denoising algorithms assume some notions of prior, either explicitly or implicitly. In this subsection, we mention a few of the important ones. For notational simplicity, we will focus on the ith diagonal entry of $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_d\}$.

3.4.3.1 BM3D [11], Shape-Adaptive BM3D [20] and BM3D-PCA [12]

BM3D and its variants have two denoising steps. In the first step, the algorithm applies a basis matrix $\boldsymbol{U}$ (either a pre-defined basis such as DCT, or a basis learned from PCA). Then, it applies a hard-thresholding to the projected coefficients to obtain a filtered image $\overline{\boldsymbol{p}}$. In the second step, the filtered image $\overline{\boldsymbol{p}}$ is used as a pilot estimate to the desired spectral component

$$\lambda_i = \frac{(\boldsymbol{u}_i^T\overline{\boldsymbol{p}})^2}{(\boldsymbol{u}_i^T\overline{\boldsymbol{p}})^2 + \sigma^2}. \tag{3.23}$$

Following our proposed Bayesian framework, we observe that the role of using $\overline{\boldsymbol{p}}$ in (3.23) is equivalent to assuming a dirac delta prior

$$f(\boldsymbol{p}) = \delta(\boldsymbol{p} - \overline{\boldsymbol{p}}). \tag{3.24}$$

In other words, the prior that BM3D assumes is concentrated at one point, $\overline{\boldsymbol{p}}$, and there is no measure of uncertainty. As a result, the algorithm becomes highly sensitive to the first-pass estimate. In contrast, (3.21) suggests that the first-pass estimate can be defined as a non-local mean solution. Additionally, we incorporate a covariance matrix $\boldsymbol{\Sigma}$ to measure the uncertainty of observing $\boldsymbol{\mu}$. These provide a more robust estimate to the denoising algorithm which is absent from BM3D and its variants.

3.4.3.2 LPG-PCA [22]

In LPG-PCA, the ith spectral component λ_i is defined as

$$\lambda_i = \frac{(\boldsymbol{u}_i^T\boldsymbol{q})^2 - \sigma^2}{(\boldsymbol{u}_i^T\boldsymbol{q})^2}, \tag{3.25}$$

where $\boldsymbol{q}$ is the noisy patch. The (implicit) assumption in [22] is that $(\boldsymbol{u}_i^T\boldsymbol{q})^2 \approx (\boldsymbol{u}_i^T\boldsymbol{p})^2 + \sigma^2$, and so by substituting $(\boldsymbol{u}_i^T\boldsymbol{p})^2 \approx (\boldsymbol{u}_i^T\boldsymbol{q})^2 - \sigma^2$ into (3.4) yields (3.25). However, the assumption implies the existence of a perturbation $\Delta\boldsymbol{p}$ such that $(\boldsymbol{u}_i^T\boldsymbol{q})^2 = (\boldsymbol{u}_i^T(\boldsymbol{p} + \Delta\boldsymbol{p}))^2 + \sigma^2$. Letting $\overline{\boldsymbol{p}} = \boldsymbol{p} + \Delta\boldsymbol{p}$, we see that LPG-PCA implicitly assumes a dirac prior as in (3.23) and (3.24). The denoising result depends on the magnitude of $\Delta\boldsymbol{p}$.

3.4.3.3 Generic Global Prior [15, 21, 57, 59]

As a comparison to methods using generic databases such as [15,21,57,59], we note that the key difference lies in the usage of a *global* prior versus a *local* prior. Figure 3.5 illustrates the concept pictorially. The generic (global) prior $f(\boldsymbol{p})$ covers the entire space, whereas the targeted (local) prior is concentrated at its mean. The advantage of the local prior is that it allows one to denoise an image with few reference patches. It saves us from the intractable computation of learning the

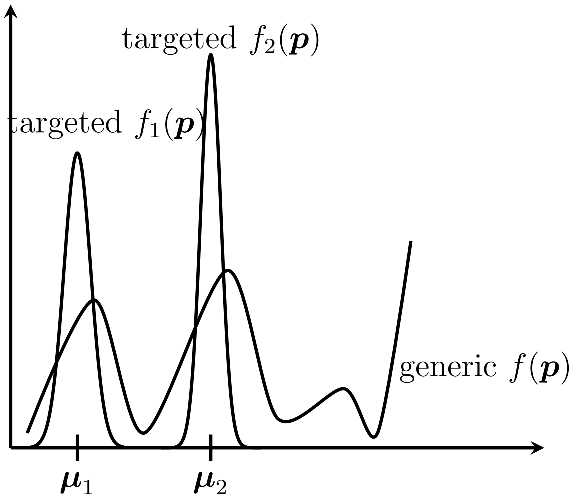

targeted $f_2(\boldsymbol{p})$
argeted $f_1(\boldsymbol{p})$
generic $f(\boldsymbol{p})$
$\boldsymbol{\mu}_1$
$\boldsymbol{\mu}_2$

Figure 3.5: Generic prior vs targeted priors: Generic prior has an arbitrary shape spanned over the entire space; Targeted priors are concentrated at the means. In this figure, $f_1(\boldsymbol{p})$ and $f_2(\boldsymbol{p})$ illustrate two targeted priors corresponding to two patches of an image.

global prior, which is a high-dimensional non-parametric function.

3.4.4 Improving $\boldsymbol{\Lambda}$

The Bayesian framework proposed above can be generalized to further improve the denoising performance. Referring to (3.19), we observe that the BMSE optimization can be reformulated to incorporate a penalty term in $\boldsymbol{\Lambda}$. Here, we consider the following ℓ_α penalized BMSE:

$$\text{BMSE}_\alpha \overset{\text{def}}{=} \mathbb{E}_{\boldsymbol{p}}\left[\mathbb{E}_{\boldsymbol{q}|\boldsymbol{p}}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q}-\boldsymbol{p}\right\|_2^2\middle|\boldsymbol{p}\right]\right]+\gamma\|\boldsymbol{\Lambda}\boldsymbol{1}\|_\alpha, \tag{3.26}$$

where $\gamma > 0$ is the penalty parameter, and $\alpha \in \{0, 1\}$ controls which norm to be used. The solution to the minimization of (3.26) is given by the following lemma.

Lemma 6. *Let s_i be the ith diagonal entry in $\boldsymbol{S}$, where $\boldsymbol{S}$ is the eigenvalue matrix of $\boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T$, then the optimal $\boldsymbol{\Lambda}$ that minimizes BMSE$_\alpha$ is* $\text{diag}\{\lambda_1, \cdots, \lambda_d\}$, *where*

$$\lambda_i = \max\left(\frac{s_i-\gamma/2}{s_i+\sigma^2}, 0\right), \qquad \textit{for } \alpha = 1, \tag{3.27}$$

and

$$\lambda_i = \frac{s_i}{s_i + \sigma^2} \mathbb{1} \left(\frac{s_i^2}{s_i + \sigma^2} > \gamma \right), \qquad \textit{for } \alpha = 0. \tag{3.28}$$

Proof. See Appendix A.6. □

The motivation of introducing an ℓ_α-norm penalty in (3.26) is related the group sparsity used in defining $\boldsymbol{U}$. Recall from Section 3.3 that since $\boldsymbol{U}$ is the optimal solution to a group sparsity optimization, only few of the entries in the ideal projection $\boldsymbol{U}^T\boldsymbol{p}$ should be non-zero. Consequently, it is desired to require $\boldsymbol{\Lambda}$ to be also sparse so that the reconstruction $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q}$ has similar spectral components as that of $\boldsymbol{p}$.

To demonstrate the effectiveness of the proposed ℓ_α formulation, we consider the example patch shown in Figure 3.3. For a refined database of $k = 40$ patches, we consider the original minimum BMSE solution ($\gamma = 0$), the ℓ_0 solution with $\gamma = 0.02$, and the ℓ_1 solution with $\gamma = 0.02$. The results in Figure 3.6 show that with the proposed penalty term, the new BMSE$_\alpha$ solution performs consistently better than the original BMSE solution.

3.5 Performance Analysis

There are two important aspects of the algorithm that we must analyze. The first one is the sensitivity of the proposed algorithm to the choice of the patches $\boldsymbol{P} = [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k]$. We want to establish the relationship between the perturbation $\Delta\boldsymbol{P}$ on $\boldsymbol{P}$ and the corresponding change in MSE. This will allow us to decide how much effort should be spent in seeking a good $\boldsymbol{P}$. The second aspect is the difficulty of finding a good $\boldsymbol{P}$, or in other words, what are the factors that affect the probability of finding a good database?

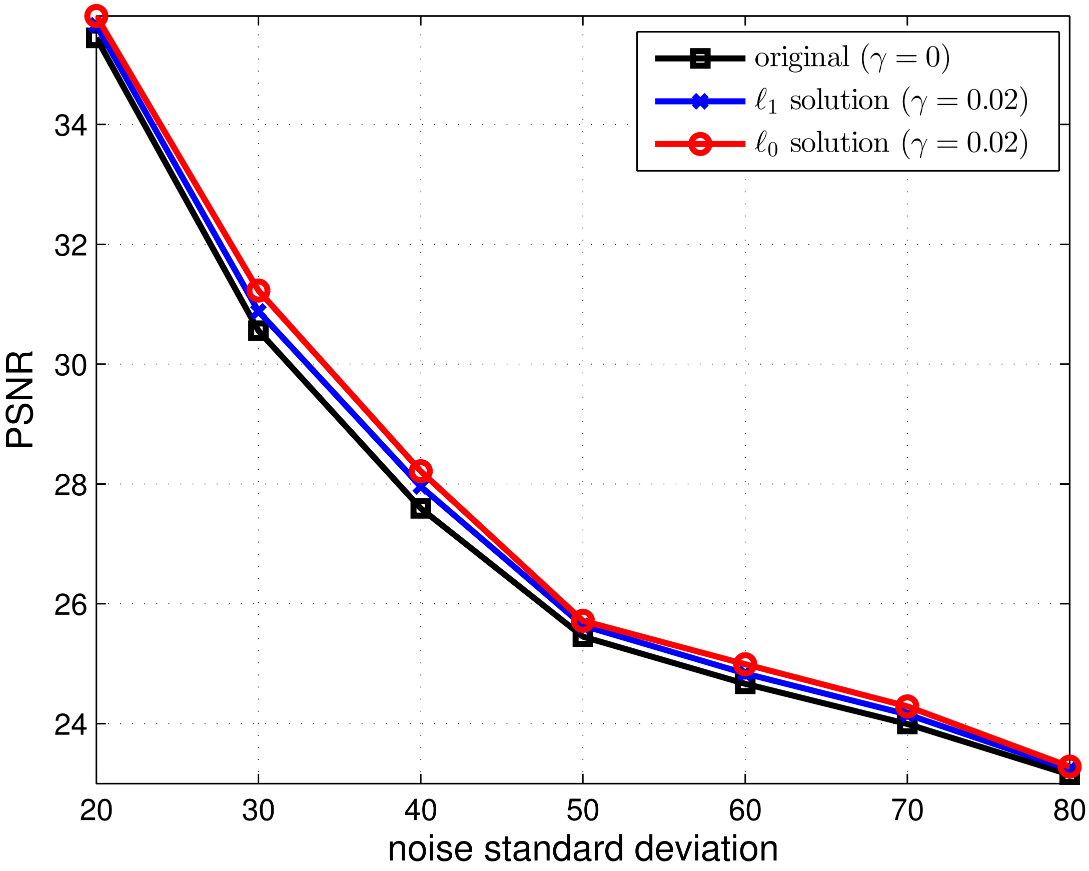


Figure 3.6: Comparisons of the ℓ_1 and ℓ_0 adaptive solutions over the original solution with $\gamma = 0$. The PSNR value for each noise level is averaged over 100 independent trials to reduce the bias due to a particular noise realization.

3.5.1 Sensitivity Analysis

The purpose of sensitivity analysis is to study the influence on the denoising quality by using different patches. More specifically, we consider

$$\begin{aligned} \boldsymbol{Q} &= [\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k] + [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_k] \\ &= \boldsymbol{P} + \Delta \boldsymbol{P} \end{aligned} \tag{3.29}$$

to be a perturbed version of $\boldsymbol{P}$ with perturbation

$$\Delta \boldsymbol{P} \overset{\text{def}}{=} [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_k]. \tag{3.30}$$

Our goal is to study the change in MSE:

$$\Delta\mathrm{MSE} = \mathbb{E}\left[\|\widehat{\boldsymbol{p}}_P - \widehat{\boldsymbol{p}}_Q\|^2\right], \tag{3.31}$$

where $\widehat{\boldsymbol{p}}_P$ is the denoised signal using $\boldsymbol{P}$ and $\widehat{\boldsymbol{p}}_Q$ is the denoised signal using $\boldsymbol{Q}$.

To make our analysis less tedious, we use a common weight matrix $\boldsymbol{W}$ to both $\boldsymbol{P}$ and $\boldsymbol{Q}$ instead of two different weights. Consequently, we define

$$\widetilde{\boldsymbol{P}} \overset{\text{def}}{=} \boldsymbol{P}\boldsymbol{W}^{1/2}, \qquad \text{and} \qquad \widetilde{\boldsymbol{Q}} \overset{\text{def}}{=} \boldsymbol{Q}\boldsymbol{W}^{1/2}, \tag{3.32}$$

and

$$\Delta\widetilde{\boldsymbol{P}} = \widetilde{\boldsymbol{P}} - \widetilde{\boldsymbol{Q}}. \tag{3.33}$$

Our main result is the following.

Theorem 1. *The MSE difference using the two different sets of patches* $\boldsymbol{P}$ *and* $\boldsymbol{Q}$ *is bounded as*

$$\Delta\mathrm{MSE} \leq \left(\gamma\|\boldsymbol{p}\|_2^2 + \sigma^2\right)\left\|\Delta\widetilde{\boldsymbol{P}}\right\|_F^2, \tag{3.34}$$

for some constant $\gamma > 0$.

Proof. See Appendix A.7. □

The implication of Theorem 1 is that

$$\Delta\mathrm{MSE} \asymp \mathcal{O}\left(\left\|\Delta\widetilde{\boldsymbol{P}}\right\|_F^2\right). \tag{3.35}$$

Therefore, if $\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_k\}$ are the k best patches found in a database, then the best MSE is upper bounded by $\left\|\Delta\widetilde{\boldsymbol{P}}\right\|_F^2$.

3.5.2 Database Analysis

Our second theoretical question concerns about the probability of getting a good database. Let $\boldsymbol{p}_0 \in \mathbb{R}^n$ be the target patch to be matched, and let the non-parametric distribution of any patch $\boldsymbol{p}$ in the entire $\mathbb{R}^n$ be $f(\boldsymbol{p})$. If we draw N i.i.d. samples $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N$ from $f(\boldsymbol{p})$, the question is: How large should N be so that at least one of the N samples is close to $\boldsymbol{p}_0$? That is, for any fixed $\varepsilon > 0$, we want to study the probability

$$\Pr\left[\min_{1\le i\le N} \|\boldsymbol{p}_i - \boldsymbol{p}_0\|_2 > \varepsilon\right]. \tag{3.36}$$

The answer to (3.36) is the following theorem.

Theorem 2.

$$\Pr\left[\min_{1\le i\le N} \|\boldsymbol{p}_i - \boldsymbol{p}_0\|_2 > \varepsilon\right] \le \exp\left\{-C\; N\; \varepsilon^n\; f(\boldsymbol{p}_0)\right\}, \tag{3.37}$$

where $C = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}$ *is a constant.*

Proof. See Appendix A.8. □

- Relation to N: When N increases, the probability of large deviation reduces exponentially. Therefore, asymptotically as the number of samples $N \to \infty$, the probability of finding a $\boldsymbol{p} \approx \boldsymbol{p}_0$ approaches 1.

- Relation to ε: ε is the precision of the deviation. Therefore, as $\varepsilon \to 0$, N has to be increased in order to retain the same convergence rate.

- Relation to $f(\boldsymbol{p}_0)$: $f(\boldsymbol{p}_0)$ is the distribution of the data point at $\boldsymbol{p}_0$. If $\boldsymbol{p}_0$ represents a rare patch, then a larger N is needed.

- Relation to n: The patch dimensionality n affects both the precision ε^n and the constant C, because $C \to 0$ as $n \to \infty$. Therefore, increased n also requires increased N.

3.6 Experimental Results

In this section, we present a set of denoising experiments to evaluate the performance of the proposed algorithm against several existing methods.

3.6.1 Comparison Methods

The methods we choose for comparison are BM3D [11], BM3D-PCA [12], LPG-PCA [22], NLM [17] and EPLL [59]. Except for EPLL, all other four methods are internal denoising methods. We re-implement and modify the internal methods so that patch search is performed over the targeted external databases. These methods are iterated for two times where the solution of the first step is used as a basic estimate for the second step. The specific settings of each algorithm are as follows:

1. BM3D-PCA [12] and LPG-PCA [22]: $\boldsymbol{U}$ is learned from the best k external patches, which is the same as in our proposed method. $\boldsymbol{\Lambda}$ is computed following (3.23) for BM3D-PCA and (3.25) for LPG-PCA. In BM3D-PCA's first step, the threshold is set to 2.7σ.

2. NLM [17]: The weights in NLM are computed according to a Gaussian function of the ℓ_2 distance of two patches [68, 69]. However, instead of using all reference patches in the database, we use the best k patches following [18].

3. BM3D [11]: As a benchmark, we run the original BM3D code[2] of Dabov et al. to show the performance of internal image denoising. For a fair comparison, we adjust the patch size, step size and search window size so that they are consistent with other methods. We use the default settings of BM3D for other parameters.

4. EPLL [59]: EPLL is an external denoising method, but the default patch prior is learned from a generic database. For a fair comparison, we use a targeted database that is used by the proposed method. We train the prior distribution using the EM algorithm mentioned in [59].

To emphasize the difference between the original algorithms (which are single-image denoising algorithms) and the corresponding new implementations for external databases, we denote the original, (single-image) denoising algorithms

[2] http://www.cs.tut.fi/~foi/GCF-BM3D/

with “i” (internal), and the corresponding new implementations when using external databases with “e” (external).

We add zero-mean Gaussian noise with standard deviations from $\sigma = 20$ to $\sigma = 80$ to the test images. The patch size is set as 8×8 ($i.e., d = 64$), and the sliding step size is 6 in the first step and 4 in the second step. Two quality metrics, namely Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) are used to evaluate the objective quality of the denoised images.

3.6.2 Denoising Text and Documents

Our first experiment considers denoising a text image. The purpose is to simulate the case where we want to denoise a noisy document with the help of other similar but non-identical texts. This idea can be easily generalized to other scenarios such as handwritten signatures, bar codes and license plates.

To prepare this scenario, we capture a region (127×104) of a document and add noise. We then build the targeted external database by cropping 9 arbitrary portions from a different document but with the same font sizes.

3.6.2.1 Denoising Performance

Figure 3.7 shows the denoising results when we add excessive noise ($\sigma = 100$) to the query image. Among all the methods, the proposed method yields the highest PSNR and SSIM values. The PSNR is 5 dB better than the benchmark BM3D (internal) denoising algorithm. Some existing training-based methods, such as EPLL, do not perform well due to the insufficient training samples from the targeted database. Compared to other external denoising methods, the proposed method shows a better utilization of the targeted database.

We plot and compare the PSNR values over a range of noise levels in Figure 3.8. Our proposed method outperforms other competitors, especially at high noise levels. For example, for $\sigma = 60$, our restored result is 0.82 dB better than the second best result by eBM3D-PCA.

3.6.2.2 Database Quality

We show how the database quality affects our denoising performance. Given a database, we compute its average distance from the clean image of interest. Specifically, for each patch $\boldsymbol{p}_i \in \mathbb{R}^d$ in a clean image containing m patches and a database $\mathcal{P}$ of n patches, we compute its minimum distance

$$d(\boldsymbol{p}_i, \mathcal{P}) \overset{\text{def}}{=} \min_{\boldsymbol{p}_j \in \mathcal{P}} \|\boldsymbol{p}_i - \boldsymbol{p}_j\|_2 / \sqrt{d}.$$

The average patch-database distance is then defined as $\overline{d}(\mathcal{P}) \overset{\text{def}}{=} (1/m) \sum_{i=1}^{m} d(\boldsymbol{p}_i, \mathcal{P})$. Therefore, a smaller $\overline{d}(\mathcal{P})$ indicates that the database is more relevant to the ground truth (clean) image.

Figure 3.9 shows the results. For all noise levels, when the average distance between an image and a database decreases, the proposed method benefits from the database quality improvement and thus enhances the denoising performance.

3.6.3 Denoising Multiview Images

Our second experiment considers the scenario of capturing images using a multiview camera system. The multiview images are captured at different viewing positions. Suppose that one or more cameras are not functioning properly so that some images are corrupted with noise. Our goal is to demonstrate that with the help of the other clean views, the noisy view could be restored.

To simulate the experiment, we download 2 multivew datasets from Middlebury Computer Vision Page[3]. Each set of images consists of 5 views. We add i.i.d. Gaussian noise to one view and then use the rest 4 views to assist in denoising.

In Figure 3.10, we visually show the denoising results of the “Barn” and “Cone” multiview datasets. In comparison to the competing methods, our proposed method has the highest PSNR values. The magnified areas indicate that our proposed method removes the noise significantly and better reconstructs some fine details. In Figure 3.11, we plot and compare the PSNR values over a range of

[3] http://vision.middlebury.edu/stereo/

noise levels. The proposed method is consistently better than its competitors. For example, for $\sigma = 50$, our proposed method is 0.80 dB better than eBM3D-PCA and 1.94 dB better than iBM3D.

3.6.4 Denoising Human Faces

Our third experiment considers denoising human face images. In low light conditions, images captured are typically corrupted by noise. To facilitate other high-level vision tasks such as recognition and tracking, denoising is a necessary pre-processing step. This experiment demonstrates the ability of denoising face images.

In this experiment, we use the Gore face database from [1], of which some examples are shown in the top row of Figure 3.12 (each image is 60×80). We simulate the denoising task by adding noise to a randomly chosen image and then use the other images (19 other face images in our experiment) in the database to assist in denoising.

In the bottom row of Figure 3.12, we show the noisy face and denoising results. We observe that while the facial expressions are different and there are misalignments between images, the proposed method still generates robust results. In Figure 3.13, we plot the PSNR curves, where we see consistent gain compared to other methods.

3.7 Conclusion

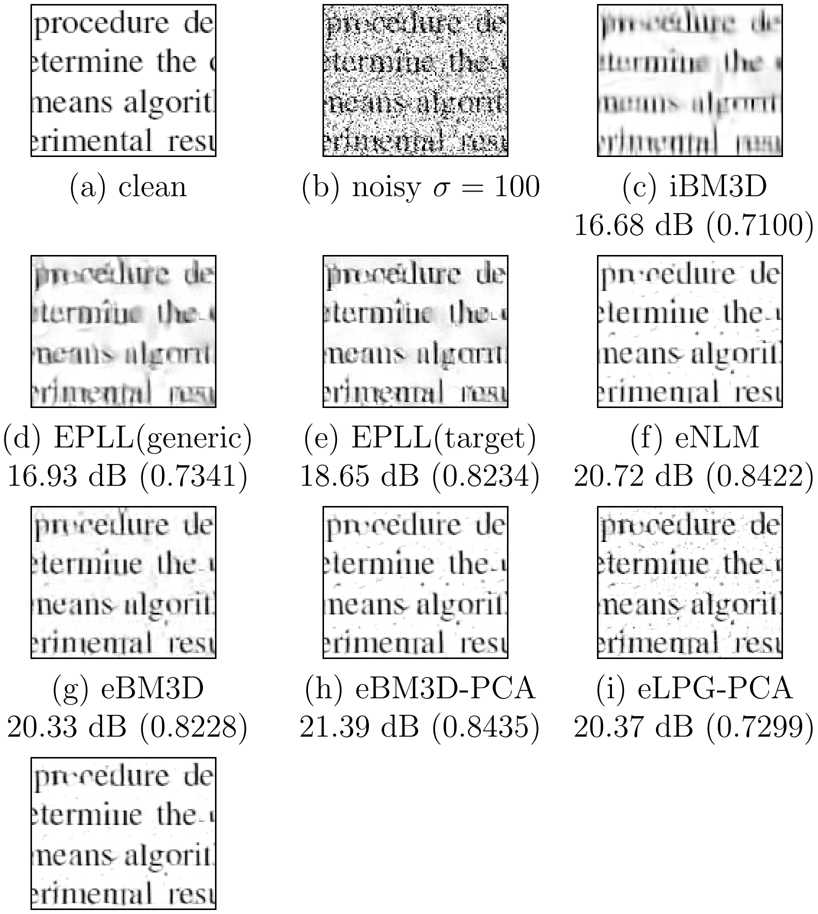
Classical image denoising methods based on a single noisy input or generic databases are approaching their performance limits. We proposed an adaptive image denoising algorithm using targeted databases. The proposed method applies a group sparsity minimization and a localized prior to learn the basis matrix and the spectral coefficients of the optimal denoising filter, respectively. We show that the new method generalizes a number of existing patch-based denoising algorithms such as BM3D, BM3D-PCA, Shape-adaptive BM3D, LPG-PCA, and EPLL. Based on the new framework, we proposed improvement schemes, namely an improved patch selection procedure for determining the basis matrix and a penalized minimization for determining the spectral coefficients. For a variety of scenarios including text, multiview images and faces, we demonstrated empirically that the proposed method has superior performance over existing methods. With the increasing amount of image data available online, we anticipate that the proposed method is an important first step towards a data-dependent generation of denoising algorithms.

3.8 Acknowledgement

This chapter, in part, is a reprint of the following papers

E. Luo, S. H. Chan, and T. Q. Nguyen, "Image Denoising by Targeted External Databases," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Process. (ICASSP'14)*, pp. 2469-2473, May 2014.

E. Luo, S. H. Chan, and T. Q. Nguyen, "Adaptive Image Denoising by Targeted Databases," *IEEE Trans. Image Process. (TIP'15)*, vol. 24, no. 7, pp. 2167-2181, Jul. 2015



(a) clean

(b) noisy $\sigma = 100$

(c) iBM3D
16.68 dB (0.7100)

(d) EPLL(generic)
16.93 dB (0.7341)

(e) EPLL(target)
18.65 dB (0.8234)

(f) eNLM
20.72 dB (0.8422)

(g) eBM3D
20.33 dB (0.8228)

(h) eBM3D-PCA
21.39 dB (0.8435)

(i) eLPG-PCA
20.37 dB (0.7299)

(j) ours
22.20 dB (0.9069)

Figure 3.7: Denoising text images: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). Prefix "*i*" stands for internal denoising (single-image denoising), and prefix "*e*" stands for external denoising (using external databases).

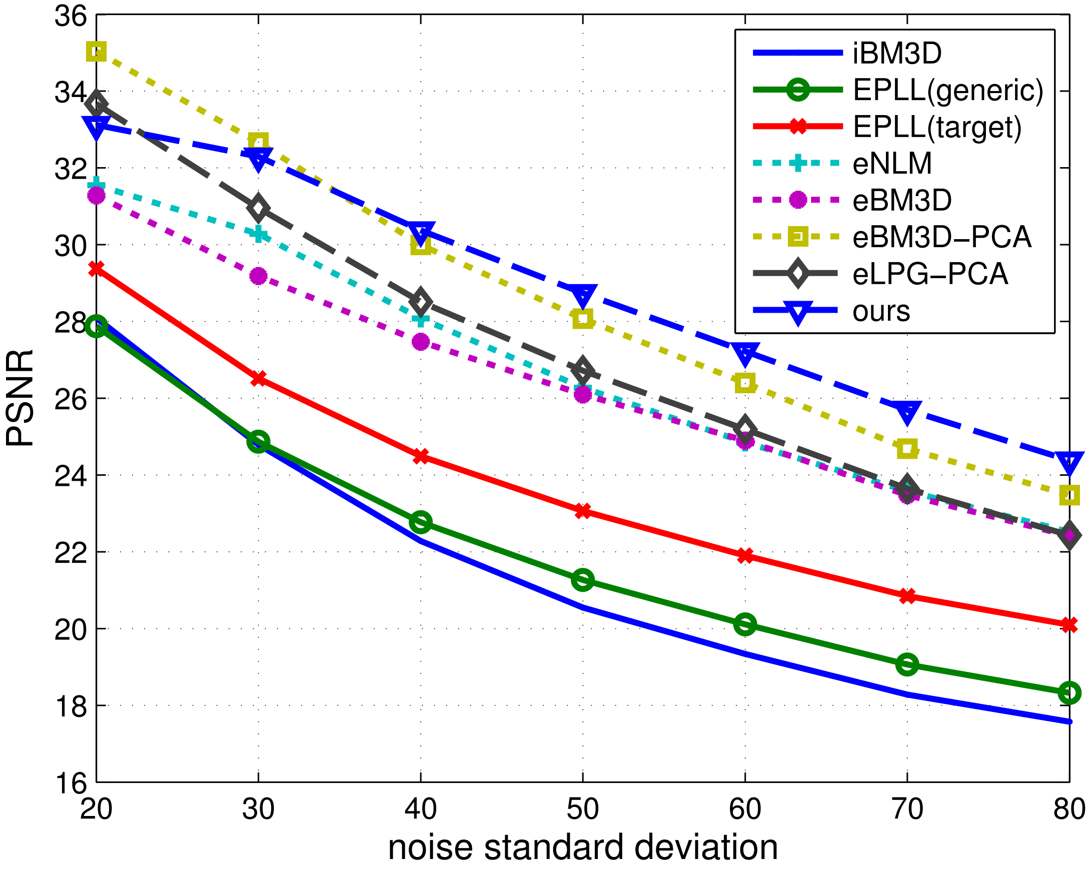


Figure 3.8: Text image denoising: PSNR vs noise levels. In this plot, each PSNR value is averaged over 8 independent Monte-Carlo trials to reduce the bias due to a particular noise realization.

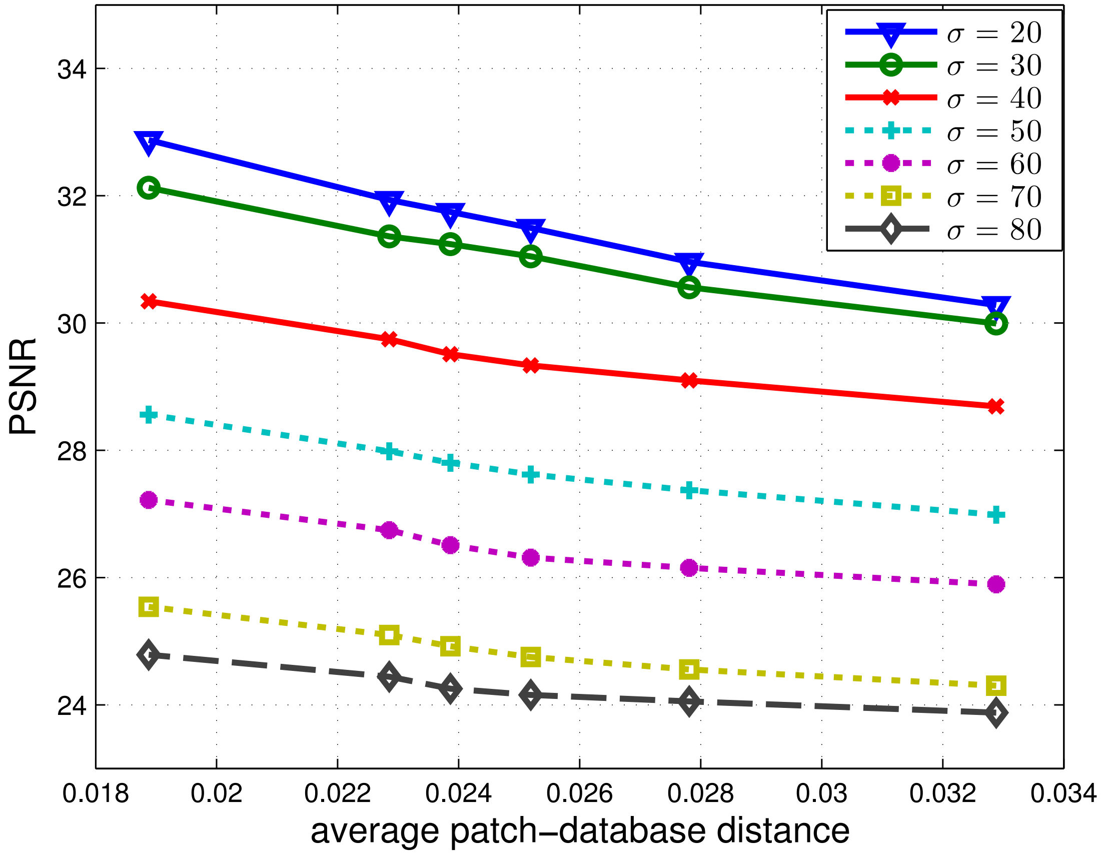
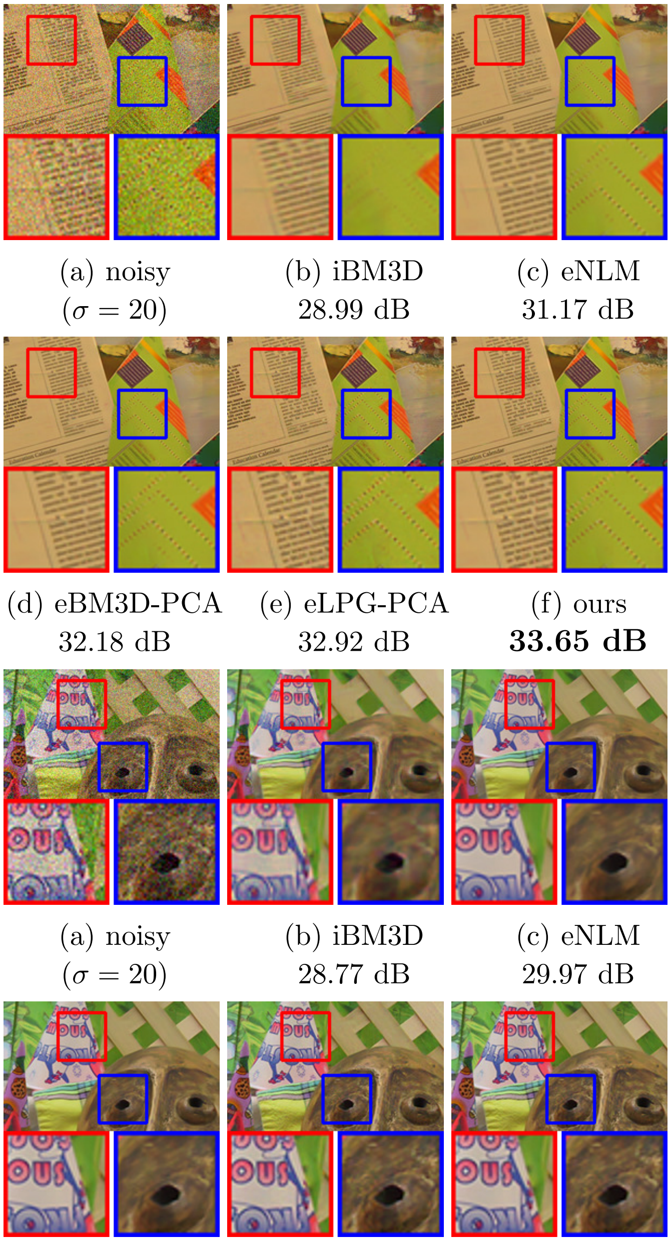


Figure 3.9: Denoising performance in terms of the database quality. The average patch-database distance $\overline{d}(\mathcal{P})$ is a measure of the database quality.



(a) noisy ($\sigma = 20$)

(b) iBM3D 28.99 dB

(c) eNLM 31.17 dB

(d) eBM3D-PCA 32.18 dB

(e) eLPG-PCA 32.92 dB

(f) ours **33.65 dB**

(a) noisy ($\sigma = 20$)

(b) iBM3D 28.77 dB

(c) eNLM 29.97 dB

(d) eBM3D-PCA 31.03 dB

(e) eLPG-PCA 31.80 dB

(f) ours **32.18 dB**

Figure 3.10: Multiview image denoising: Visual comparison and objective comparison (PSNR). [Top two rows] “Barn”; [Bottom two rows] “Cone”.

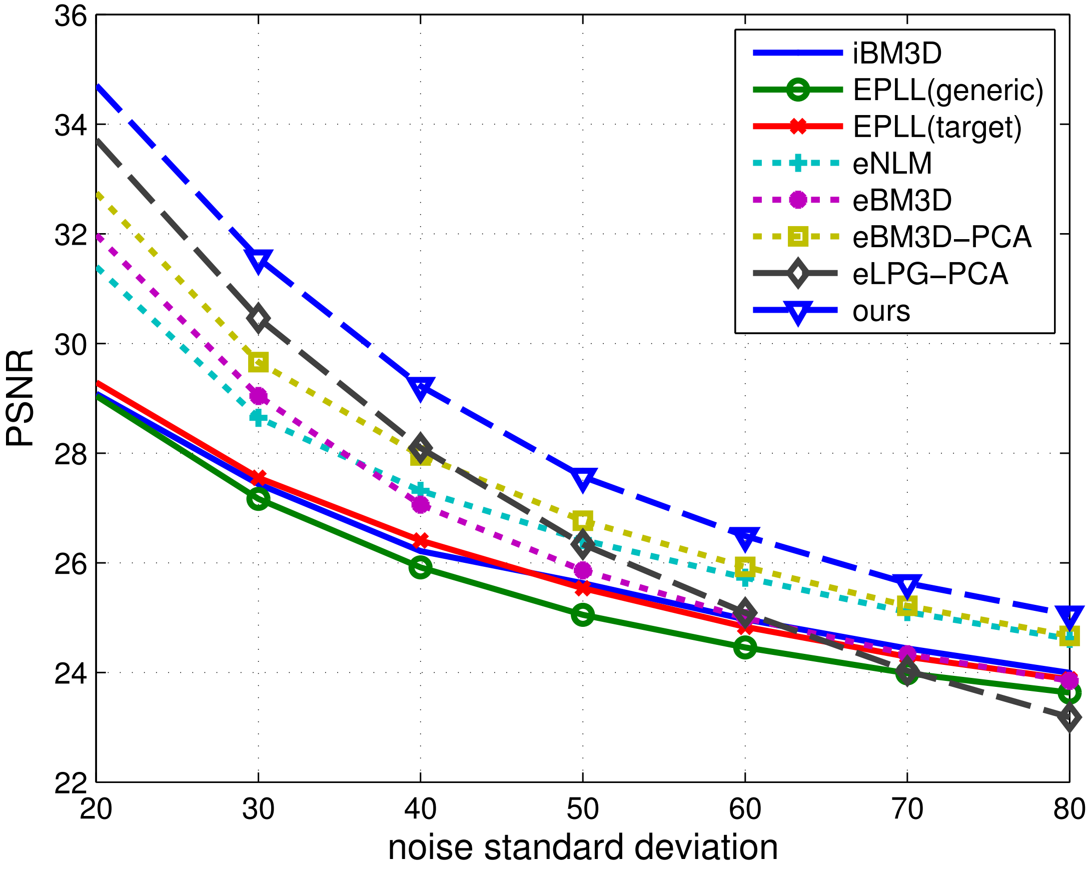


Figure 3.11: Multiview image denoising for "Barn": PSNR vs noise levels. In this plot, each PSNR value is averaged over 8 independent Monte-Carlo trials to reduce the bias due to a particular noise realization.

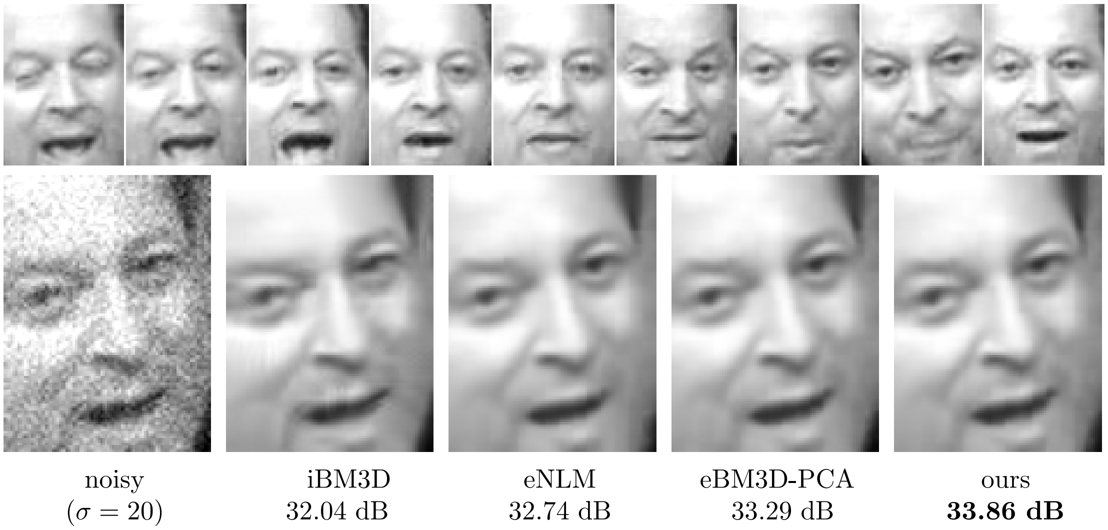


Figure 3.12: Face denoising of Gore dataset [1]. [Top] Database images; [Bottom] Denoising results.

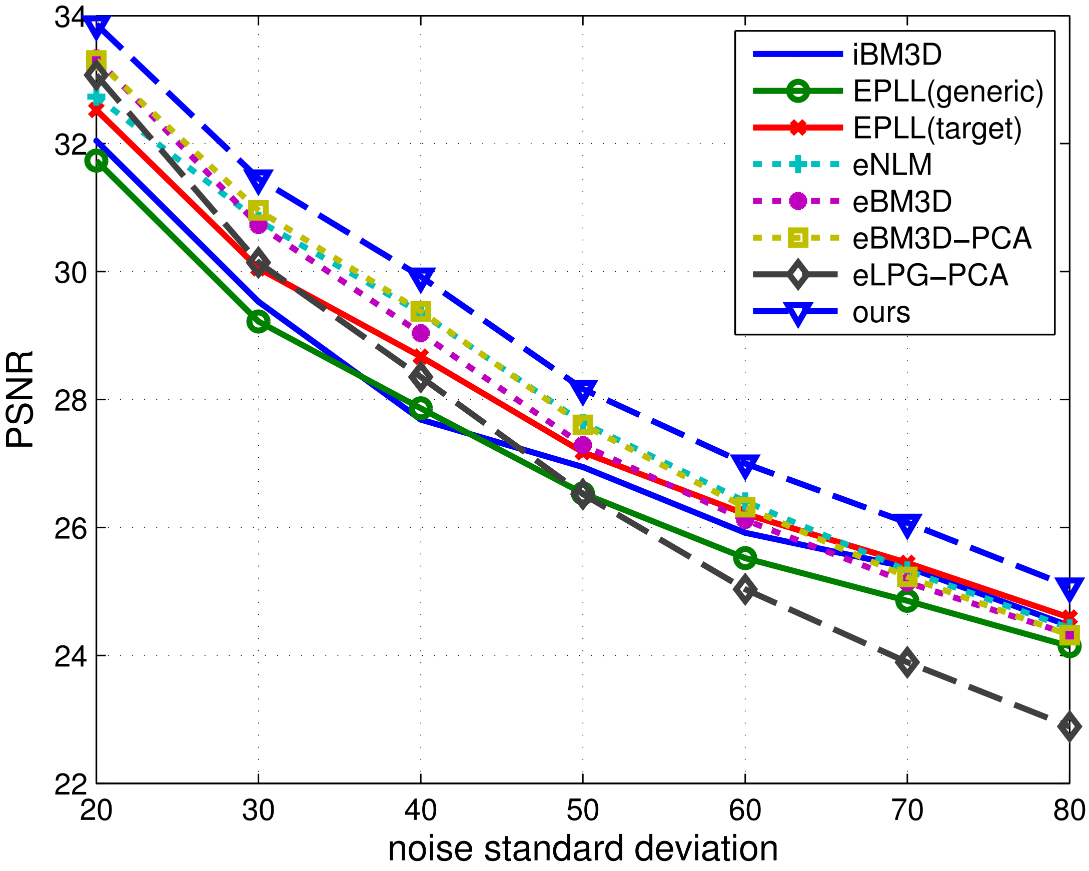


Figure 3.13: Face denoising results: PSNR vs noise levels. In this plot, each PSNR value is averaged over 8 independent Monte-Carlo trials to reduce the bias due to a particular noise realization.

Chapter 4

Adaptive Image Denoising by Mixture Adaptation

4.1 Introduction

In Chapter 3, we propose to switch from generic database to targeted database. We demonstrate that by mining appropriate databases exceptional external priors can be learned. In some practical scenarios such as text image denoising and face image denoising, building a targeted database is plausible. However, the challenge is that building a targeted database for any image of interest can be difficult. In this chapter, we consider the scenario when only a generic database (as opposed to a targeted database) is available. We propose to learn a generic prior from the generic database and then adapt the generic prior to the image of interest to create a specific prior.

Before discussing the proposed prior adaptation idea, we introduce the denoising framework – maximum-a-posteriori (MAP) approach [15,59]. Let the noisy model be

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon}, \tag{4.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ denotes the (unknown) clean image, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}) \in \mathbb{R}^n$ denotes the additive i.i.d. Gaussian noise with σ^2 noise variance, and $\boldsymbol{y} \in \mathbb{R}^n$ denotes the observed noisy image. MAP is a Bayesian approach which formulates the image denoising problem by maximizing the posterior probability

$$\operatorname*{argmax}_{\boldsymbol{x}} \; f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x}) = \operatorname*{arg\,min}_{\boldsymbol{x}} \left\{ \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 - \log f(\boldsymbol{x}) \right\},$$

where the first term is a quadratic due to the Gaussian noise model, and the second term is the negative log of the prior of the latent clean image.

We choose to use MAP because of its ability to explicitly formulate the prior knowledge about the image via the prior distribution $f(\boldsymbol{x})$. Thus, finding a good prior $f(\boldsymbol{x})$ is of vital importance for successful MAP optimization [58, 60, 65]. However, modeling the whole image $\boldsymbol{x}$ is extremely difficult if not impossible because of the high dimensionality of $\boldsymbol{x}$. To alleviate the problem, we adopt the common wisdom by approximating $f(\boldsymbol{x})$ using a collection of small patches [11, 17, 59]. Such prior is known as the *patch prior*. Mathematically, letting $\boldsymbol{P}_i \in \mathbb{R}^{d\times n}$ be a patch-extract operator which extracts the i-th d-dimensional patch from the image $\boldsymbol{x}$, a patch prior expresses the negative log of the image prior as a sum of the log patch priors. Therefore, the MAP framework becomes

$$\operatorname*{arg\,min}_{\boldsymbol{x}} \left\{ \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 - \frac{1}{n}\sum_{i=1}^{n} \log f(\boldsymbol{P}_i\boldsymbol{x}) \right\}, \tag{4.2}$$

where the second term in (4.2) is called the expected patch log likelihood (EPLL) [59].

The focus of this work is a robust and efficient way of learning the model parameters of $f(\boldsymbol{P}_i\boldsymbol{x})$. Generally speaking, estimating the model parameter requires a good training set of data, which can be either obtained internally (*i.e.*, from the single noisy image) or externally (*i.e.*, from a database of images). Our approach combines the power of internal [26] and external priors [27, 28, 70–72]. It is different from the existing *fusion* approaches which merely combine the results of the internal and the external methods. For example, Mosseri *et al.* [27] used a patch signal-to-noise ratio as a quantitative metric to decide whether a patch should be denoised internally or externally; Burger *et al.* [28] applied a neural network approach to learn the weights to combine internal and external denoising results; Yue *et al.* [73] fused the internal and external denoising results in the frequency domain. In all these approaches, there is no theoretically optimal way to calculate the weights.

4.1.1 Contribution and Organization

Our proposed algorithm is an *adaptation* approach. Like many external methods, we assume that we have an external database of images for training. However, we do not simply compute the statistics of the external database. Instead, we use the external statistics as a "guide" for learning the internal statistics. As will be illustrated in the subsequent sections, this can be formally done using a Bayesian framework.

This chapter is an extension of our previous work reported in [74]. The three new contributions are:

1. Derivation of the EM adaptation algorithm. We rigorously derive the proposed EM adaptation algorithm from a full Bayesian hyper-prior perspective. Our derivation complements the work of Gauvain and Lee [75] by providing additional simplifications and justifications to reduce computational complexity. We further provide discussion of the convergence.

2. Handling of noisy data. We provide detailed discussion of how to perform EM adaptation for noisy images. In particular, we demonstrate how to automatically adjust the internal parameters of the algorithm using pre-filtered images.

3. Extended denoising applications. We demonstrate how the proposed EM adaptation algorithm can be used to adapt noisy images, external databases, and targeted databases.

Our work is similar to a very recent work of Lu *et al.* [76]. In comparison with [76], we provide significantly more technical insights, in particular, the full Bayesian derivation, computational simplification, convergence analysis, noise handling, and significantly broader range of applications.

The rest of the chapter is organized as follows. Section 4.2 gives a brief review of Gaussian mixture model. Section 4.3 presents the proposed EM adaptation algorithm. Section 4.4 discusses how the EM adaptation algorithm should be modified when the image is noisy. Experimental results are presented in Section 4.5.

4.2 Mathematical Preliminaries

In this section we provide a brief review of the Gaussian mixture model (GMM) and the corresponding image denoising algorithm under the MAP framework, which will serve as foundation for our subsequent discussions of the proposed adaptation algorithm.

4.2.1 GMM and MAP Denoising

For notational simplicity, we shall denote $\boldsymbol{p}_i \overset{\text{def}}{=} \boldsymbol{P}_i\boldsymbol{x} \in \mathbb{R}^d$ as the i-th patch from $\boldsymbol{x}$. We say that $\boldsymbol{p}_i$ is generated from a GMM if the distribution $f(\boldsymbol{p}_i \,|\, \boldsymbol{\Theta})$ is

$$f(\boldsymbol{p}_i \,|\, \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{p}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4.3}$$

where $\sum_{k=1}^{K} \pi_k = 1$ with π_k being the weight of the k-th Gaussian component, and

$$\begin{aligned} &\mathcal{N}(\boldsymbol{p}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &\overset{\text{def}}{=} \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\Big(-\frac{1}{2}(\boldsymbol{p}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{p}_i - \boldsymbol{\mu}_k) \Big) \end{aligned} \tag{4.4}$$

is the k-th Gaussian distribution with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$. We denote $\boldsymbol{\Theta} \overset{\text{def}}{=} \{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^{K}$ as the GMM parameter.

With the GMM defined in (4.3), we can specify the denoising procedure by solving the optimization problem in (4.2). Here, we follow [77, 78] by using the *Half Quadratic Splitting* strategy. The idea is to replace (4.2) with the following equivalent minimization

$$\begin{aligned}\underset{\boldsymbol{x},\{\boldsymbol{v}_i\}_{i=1}^n}{\arg\min}\Bigg\{&\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{x}\|^2\\&+\frac{1}{n}\sum_{i=1}^n\Big(-\log f(\boldsymbol{v}_i)+\frac{\beta}{2}\|\boldsymbol{P}_i\boldsymbol{x}-\boldsymbol{v}_i\|^2\Big)\Bigg\},\end{aligned}\tag{4.5}$$

where $\{\boldsymbol{v}_i\}_{i=1}^n$ are some auxiliary variables and β is a penalty parameter. By assuming that $f(\boldsymbol{v}_i)$ is dominated by the mode of the Gaussian mixture, the solution to (4.5) is given in the following proposition.

Proposition 2. *Assuming $f(\boldsymbol{v}_i)$ is dominated by the k_i^*-th components, where $k_i^* \overset{def}{=} \underset{k}{argmax}\, \pi_k\mathcal{N}(\boldsymbol{v}_i|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$, the solution of* (4.5) *is*

$$\begin{aligned}\boldsymbol{x} &= \Big(n\sigma^{-2}\boldsymbol{I}+\beta\sum_{i=1}^n\boldsymbol{P}_i^T\boldsymbol{P}_i\Big)^{-1}\Big(n\sigma^{-2}\boldsymbol{y}+\beta\sum_{i=1}^n\boldsymbol{P}_i^T\boldsymbol{v}_i\Big),\\ \boldsymbol{v}_i &= \left(\beta\boldsymbol{\Sigma}_{k_i^*}+\boldsymbol{I}\right)^{-1}\Big(\boldsymbol{\mu}_{k_i^*}+\beta\boldsymbol{\Sigma}_{k_i^*}\boldsymbol{P}_i\boldsymbol{x}\Big).\end{aligned}$$

Proof. See Appendix B.1 or [59]. □

Proposition 2 is a general procedure for denoising images using a GMM under the MAP framework. There are, of course, other possible denoising procedures which also use GMM under the MAP framework, *e.g.*, using surrogate methods [79]. However, we shall not elaborate on these options. Our focus is on how to obtain the GMM.

4.2.2 EM Algorithm

The GMM parameter $\boldsymbol{\Theta}=\{(\pi_k,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)\}_{k=1}^K$ is typically learned using the Expectation-Maximization (EM) algorithm from a large collection of training samples. EM is a known method and so we shall skip the introduction. Interested readers can refer to [80] for a comprehensive tutorial. What is more important are the limitations of EM when applied to image denoising:

1. **Adaptivity**: For a fixed image database, the GMM parameters are specifically trained for this particular database. We call it the *generic* parameter. If, for example, we are given an image which does not necessarily belong to the database, then it becomes unclear how one can adapt the generic parameter to the image.

2. **Computational cost**: Learning a good GMM requires a large number of training samples. For example, the GMM in [59] is learned from 2,000,000 randomly sampled patches. If our goal is to adapt a generic parameter to a particular image, then it would be more desirable to bypass the computational intensive procedure.

3. **Finite samples**: When training samples are few, the learned GMM will be over-fitted; some components will even become singular. This problem needs to be resolved because a noisy image contains much fewer patches than a database of patches.

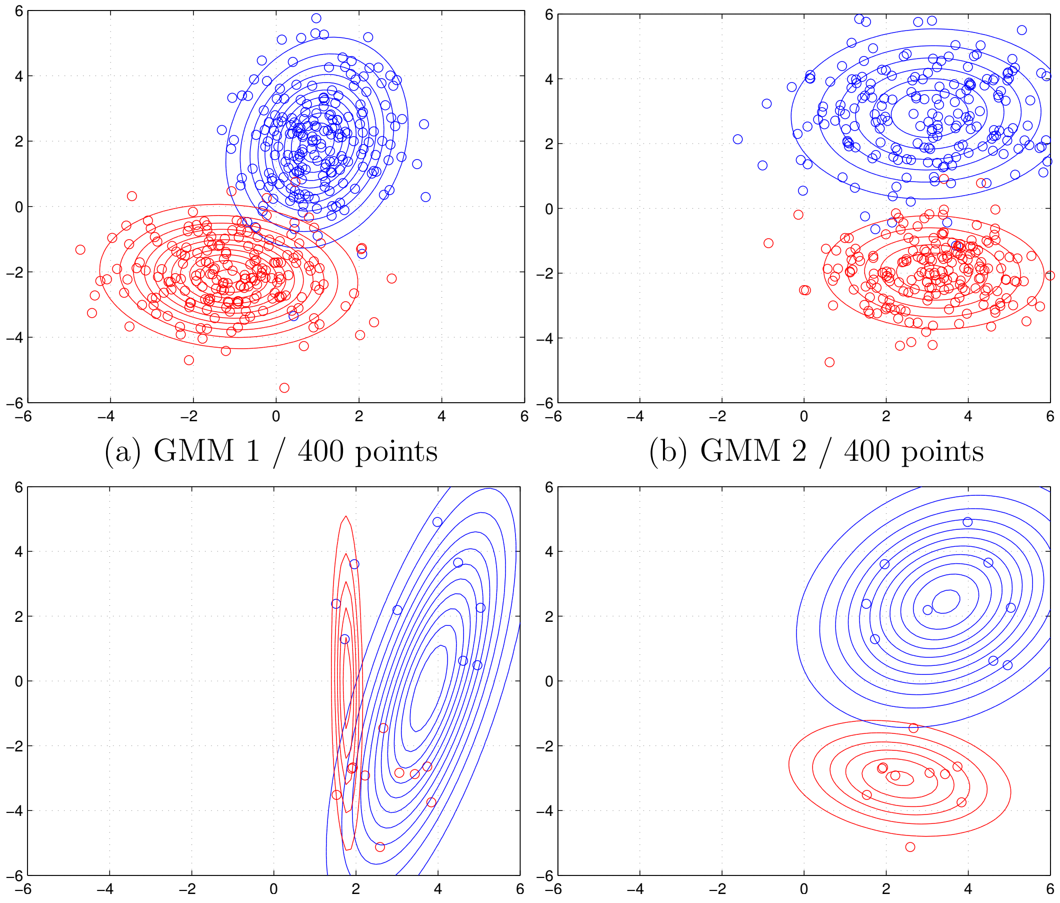
4. **Noise**: In image denoising, the observed image always contains noise. It is not clear how to mitigate the noise while running the EM algorithm.

4.3 EM Adaptation

The proposed EM adaptation takes a generic prior and adapts it to create a specific prior using very few samples. Before giving the details of the EM adaptation, we first provide a toy example to illustrate the idea.

4.3.1 Toy Example

Suppose we are given two 2-dimensional GMMs with 2 clusters in each GMM. From each GMM, we synthetically generate 400 data points with each point representing a 2D coordinate shown in Figure 4.1 (a) and (b). Imagine that the data points in (a) come from an external database whereas the data points in (b) come from a clean image of interest.



(a) GMM 1 / 400 points

(b) GMM 2 / 400 points

(c) GMM 2 / 20 points

(d) Adapted / 20 points

Figure 4.1: (a) and (b): Two GMMs, each learned using the EM algorithm from 400 data points of 2D coordinates. (c): A GMM learned from a subset of 20 data points drawn from (b). (d): An adapted GMM using the same 20 data points in (c).

With the two sets of data, we apply EM to learn the two individual GMMs. Since we have enough samples, the GMMs are estimated reasonably well shown in (a) and (b). However, imagine that we only have 20 data points from (b), as shown in (c). If we learn a GMM from these 20 data points, then the learned GMM becomes over-fitted to these 20 data points. This is reflected in the very different behavior of (c) compared to (b). So we ask a question: Can we start with GMM 1 and adapt it to create a specific GMM for the finite 20 data points? EM adaptation provides a solution. We observe in (d) that the adapted GMM is significantly better than (c), despite the fact that it only uses 20 data points.

4.3.2 Bayesian Hyper-prior

As illustrated in the toy example, what EM adaptation does is to use the generic model parameters as a "guide" when learning the new model parameters. Mathematically, suppose that $\{\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n\}$ are patches from a single image parameterized by a GMM with a parameter $\widetilde{\boldsymbol{\Theta}} \overset{\text{def}}{=} \{(\widetilde{\pi}_k, \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^K$. Our goal is to estimate $\widetilde{\boldsymbol{\Theta}}$ with the aid of some generic GMM parameter $\boldsymbol{\Theta}$. Before we discuss how this is done, we present a brief overview of a Bayesian inference framework.

From a Bayesian inference perspective, estimation of the parameter $\widetilde{\boldsymbol{\Theta}}$ can be formulated as

$$\begin{aligned}\widetilde{\boldsymbol{\Theta}} &= \underset{\widetilde{\boldsymbol{\Theta}}}{\operatorname{argmax}}\ \log f(\widetilde{\boldsymbol{\Theta}} \,|\, \widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \\ &= \underset{\widetilde{\boldsymbol{\Theta}}}{\operatorname{argmax}} \Big(\log f(\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n \,|\, \widetilde{\boldsymbol{\Theta}}) + \log f(\widetilde{\boldsymbol{\Theta}}) \Big),\end{aligned} \tag{4.6}$$

where

$$f(\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n \,|\, \widetilde{\boldsymbol{\Theta}}) = \prod_{i=1}^{n} \left\{ \sum_{k=1}^{K} \widetilde{\pi}_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i | \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k) \right\}$$

is the joint distribution of the samples, and $f(\widetilde{\boldsymbol{\Theta}})$ is some prior of $\widetilde{\boldsymbol{\Theta}}$. We note that (4.6) is also a MAP problem. However, the MAP for (4.6) is the estimation of the model parameter $\widetilde{\boldsymbol{\Theta}}$, which is different from the MAP for denoising used in (4.2). Although the difference seems subtle, there is a drastic different implication which we should be aware of.

In (4.6), $f(\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n \,|\, \widetilde{\boldsymbol{\Theta}})$ denotes the distribution of a collection of patches conditioned on the parameter $\widetilde{\boldsymbol{\Theta}}$. It is the likelihood of observing $\{\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n\}$ given the model parameter $\widetilde{\boldsymbol{\Theta}}$. $f(\widetilde{\boldsymbol{\Theta}})$ is a distribution of the parameter, which is called hyper-prior in machine learning [81]. Since $\widetilde{\boldsymbol{\Theta}}$ is the model parameter, the hyper-prior $f(\widetilde{\boldsymbol{\Theta}})$ defines the probability density of $\widetilde{\boldsymbol{\Theta}}$.

Same as the usual Bayesian modeling, hyper-priors are chosen according to a subjective belief. However, for efficient computation, hyper-priors are usually chosen as the *conjugate priors* of the likelihood function $f(\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n \,|\, \widetilde{\boldsymbol{\Theta}})$ so that the posterior distribution $f(\widetilde{\boldsymbol{\Theta}} \,|\, \widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n)$ has the same functional form as the prior distribution. For example, Beta distribution is a conjugate prior for a

Bernoulli likelihood function, Gaussian distribution is a conjugate prior for a likelihood function that is also Gaussian, etc. For more discussions on conjugate priors we refer the readers to [81].

4.3.3 $f(\widetilde{\boldsymbol{\Theta}})$ for GMM

For GMM, no joint conjugate prior can be found through the sufficient statistic approach [75]. However, we can separately model the mixture weight vector and the parameters for each individual Gaussian and then combine.

First, the mixture gains can be modeled as a multinomial distribution so that the corresponding conjugate prior for the mixture weight vector $(\widetilde{\pi}_1, \cdots, \widetilde{\pi}_K)$ is a Dirichlet density

$$\widetilde{\pi}_1, \cdots, \widetilde{\pi}_K \sim \mathrm{Dir}(v_1, \cdots, v_k), \tag{4.7}$$

where $v_i > 0$ is a pseudo-count for the Dirichlet distribution.

For mean and covariance $(\widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)$, a practical solution is the normal-inverse-Wishart density so that

$$(\widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k) \sim \mathrm{NIW}(\boldsymbol{\vartheta}_k, \tau_k, \boldsymbol{\Psi}_k, \varphi_k), \quad \text{for } k = 1, \cdots, K, \tag{4.8}$$

where $(\boldsymbol{\vartheta}_k, \tau_k, \boldsymbol{\Psi}_k, \varphi_k)$ are the parameters for the normal-inverse-Wishart density such that $\boldsymbol{\vartheta}_k$ is a vector of dimension d, $\tau_k > 0$, $\boldsymbol{\Psi}_k$ is a $d \times d$ positive definite matrix, and $\varphi_k > d - 1$.

Remark 3. *The choice of the normal-inverse-Wishart is important here, for it is the conjugate prior of a multivariate normal distribution with unknown mean and unknown covariance matrix. This choice is slightly different from [75] where the authors choose a normal-Wishart, which, in our opinion, is less efficient.*

Assuming all the parameters are independent, we can model $f(\widetilde{\boldsymbol{\Theta}})$ as a product of (4.7) and (4.8). By ignoring the scaling constants, it is not difficult to show that

$$\begin{aligned}&f(\widetilde{\boldsymbol{\Theta}}) \propto \textstyle\prod_{k=1}^{K} \Big\{ \widetilde{\pi}_k^{v_k-1} |\widetilde{\boldsymbol{\Sigma}}_k|^{-(\varphi_k+d+2)/2} \\ &\exp\left(-\tfrac{\tau_k}{2} \left(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k\right)^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} \left(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k\right) - \tfrac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_k \widetilde{\boldsymbol{\Sigma}}_k^{-1})\right) \Big\}.\end{aligned} \tag{4.9}$$

The importance of (4.9) is that it is a conjugate prior of the complete data. As a result, the posterior density $f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n)$ belongs to the same distribution family as $f(\widetilde{\boldsymbol{\Theta}})$. This can be formally described in the following Proposition.

Proposition 3. *Given the prior in* (4.9)*, the posterior* $f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n)$ *is given by*

$$\begin{aligned}&f(\widetilde{\boldsymbol{\Theta}} \,|\, \widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \propto \textstyle\prod_{k=1}^{K} \Big\{ \widetilde{\pi}_k^{v'_k-1} |\widetilde{\boldsymbol{\Sigma}}_k|^{-(\varphi'_k+d+2)/2} \\ &exp\left(-\tfrac{\tau'_k}{2} \left(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k\right)^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} \left(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k\right) - \tfrac{1}{2}tr(\boldsymbol{\Psi}'_k \widetilde{\boldsymbol{\Sigma}}_k^{-1})\right) \Big\}\end{aligned} \tag{4.10}$$

where

$$\begin{aligned} &v'_k = v_k + n_k, \quad \varphi'_k = \varphi_k + n_k, \quad \tau'_k = \tau_k + n_k, \\ &\boldsymbol{\vartheta}'_k = \frac{\tau_k \boldsymbol{\vartheta}_k + n_k \bar{\boldsymbol{\mu}}_k}{\tau_k + n_k}, \\ &\boldsymbol{\Psi}'_k = \boldsymbol{\Psi}_k + \boldsymbol{S}_k + \frac{\tau_k n_k}{\tau_k + n_k}(\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)^T, \\ &\bar{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i, \quad \boldsymbol{S}_k = \sum_{i=1}^{n} \gamma_{ki}(\widetilde{\boldsymbol{p}}_i - \bar{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \bar{\boldsymbol{\mu}}_k)^T \end{aligned}$$

are the parameters for the posterior density.

Proof. See Appendix B.2. □

4.3.4 Solve for $\widetilde{\boldsymbol{\Theta}}$

Solving for the optimal $\widetilde{\boldsymbol{\Theta}}$ is equivalent to solving the following optimization problem

$$\begin{array}{ll} \underset{\widetilde{\boldsymbol{\Theta}}}{\text{maximize}} & L(\widetilde{\boldsymbol{\Theta}}) \overset{\text{def}}{=} \log f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \\ \text{subject to} & \sum_{k=1}^{K} \widetilde{\pi}_k = 1. \end{array} \tag{4.11}$$

The constrained problem (4.11) can be solved by considering the Lagrange function and taking derivatives with respect to each individual parameter. We summarize the optimal solutions in the following Proposition.

Proposition 4. *The optimal* $(\widetilde{\pi}_k, \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)$ *for* (4.11) *are*

$$\begin{aligned}\widetilde{\pi}_k &= \frac{n}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{n_k}{n} \\ &\quad + \frac{\sum_{k=1}^{K} v_k - K}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{v_k - 1}{\sum_{k=1}^{K} v_k - K}, \end{aligned} \tag{4.12}$$

$$\widetilde{\boldsymbol{\mu}}_k = \frac{1}{\tau_k + n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i + \frac{\tau_k}{\tau_k + n_k} \boldsymbol{\vartheta}_k, \tag{4.13}$$

$$\begin{aligned}\widetilde{\boldsymbol{\Sigma}}_k &= \frac{n_k}{\varphi_k + d + 2 + n_k} \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T \\ &\quad + \frac{1}{\varphi_k + d + 2 + n_k} \left(\boldsymbol{\Psi}_k + \tau_k (\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \end{aligned} \tag{4.14}$$

Proof. See Appendix B.3. □

Remark 4. *The results we showed in Proposition 4 are different from [75]. In particular, the denominator for* $\widetilde{\boldsymbol{\Sigma}}_k$ *in [75] is* $\varphi_k - d + n_k$ *whereas ours is* $\varphi_k + d + 2 + n_k$*. However, by using the following simplification, we can obtain the same result for both cases.*

4.3.5 Simplification of $\widetilde{\boldsymbol{\Theta}}$

The results in Proposition 4 are general expressions for any hyper-parameters. We now discuss how to simplify the result with the help of the generic prior. First, since $\frac{v_k - 1}{\sum_{k=1}^{K} v_k - K}$ is the mode of the Dirichlet distribution, a good surrogate for it is π_k. Second, $\boldsymbol{\vartheta}_k$ denotes the prior mean in the normal-inverse-Wishart distribution and thus can be appropriately approximated by $\boldsymbol{\mu}_k$. Moreover, since $\boldsymbol{\Psi}_k$ is the scale matrix on $\widetilde{\boldsymbol{\Sigma}}_k$ and τ_k denotes the number of prior measurements in the normal-inverse-Wishart distribution, they can be reasonably chosen as $\boldsymbol{\Psi}_k = (\varphi_k + d + 2)\boldsymbol{\Sigma}_k$

and $\tau_k = \varphi_k + d + 2$. Plugging these approximations in the results of Proposition 4, we summarize the simplification results as follows.

Proposition 5. *Define* $\rho \overset{def}{=} \frac{n_k}{n}(\sum_{k=1}^K v_k - K) = \tau_k = \varphi_k + d + 2$. *Let*

$$\boldsymbol{\vartheta}_k = \boldsymbol{\mu}_k, \quad \boldsymbol{\Psi}_k = (\varphi_k + d + 2)\boldsymbol{\Sigma}_k, \quad \frac{v_k - 1}{\sum_{k=1}^K v_k - K} = \pi_k,$$

and $\alpha_k = \frac{n_k}{\rho + n_k}$, *then* (4.12)-(4.14) *become*

$$\widetilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k)\pi_k, \tag{4.15}$$

$$\widetilde{\boldsymbol{\mu}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \widetilde{\boldsymbol{p}}_i + (1 - \alpha_k)\boldsymbol{\mu}_k, \tag{4.16}$$

$$\begin{aligned} \widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T \\ + (1 - \alpha_k)\left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T\right). \end{aligned} \tag{4.17}$$

Remark 5. *We note that Reynold* et al. *[82] presented similar simplification results (without derivations) as ours. However, their results are only for the scalar case or when the covariance matrices are diagonal. In contrast, our results support full covariance matrices and thus are more general. As will be seen, for our denoising application, since the image pixels (especially adjacent pixels) are correlated, full matrix GMMs are required.*

Comparing (4.17) *with the work of Lu* et al. *[76], we note that in [76] the covariance is*

$$\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T + (1 - \alpha_k)\boldsymbol{\Sigma}_k. \tag{4.18}$$

This result, although looks reasonable, is generally not valid if we follow the Bayesian hyper-prior approach, unless $\boldsymbol{\mu}_k$ *and* $\widetilde{\boldsymbol{\mu}}_k$ *are equal to 0.*

4.3.6 EM Adaptation Algorithm

The proposed EM adaptation algorithm is summarized in Algorithm 3. EM adaptation shares many similarities with the standard EM algorithm. To better understand the differences, we take a closer look at each step.

E-Step: E-step in the EM adaptation is the same as in EM algorithm: We compute the likelihood of $\widetilde{\boldsymbol{p}}_i$ conditioned on the generic parameter $(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ as

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \tag{4.19}$$

M-Step: The more interesting step of the adaptation is the M-step. From (4.22) to (4.24), $(\widetilde{\pi}_k, \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)$ are updated through a linear combination of the contributions from the new data and the generic parameters. On one extreme when $\alpha_k = 1$, the M-step turns exactly back to the M-step in EM algorithm. On the other extreme when $\alpha_k = 0$, all emphasis is put on the generic parameters. For α_k that lies in between, the updates are a weighted averaging of the new data and the generic parameters. Taking the mean as an example, the EM adaptation updates the mean according to

$$\widetilde{\boldsymbol{\mu}}_k = \underbrace{\alpha_k \left(\frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i \right)}_{\textbf{new data}} + \underbrace{(1 - \alpha_k) \boldsymbol{\mu}_k}_{\textbf{generic prior}} . \tag{4.20}$$

The updated mean in (4.20) is a linear combination of two terms, where the first term is an empirical data average with the fractional weight γ_{ki} from each data point $\widetilde{\boldsymbol{p}}_i$ and the second term is the generic mean $\boldsymbol{\mu}_k$. Similarly for the covariance update in (4.24), the first term computes an empirical covariance with each data point weighted by γ_{ki} which is the same as in the M-step of EM algorithm, and the second term includes the generic covariance along with an adjustment term $(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T$. These two terms are then linearly combined to yield the updated covariance.

4.3.7 Convergence

The EM adaptation shown in Algorithm 3 is an EM algorithm. Therefore, its convergence is guaranteed by the classical theory, which we state without proof

Algorithm 3 EM adaptation Algorithm

```
Input: $\boldsymbol{\Theta} = \{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^K$, $\{\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n\}$.
Output: Adapted parameters $\widetilde{\boldsymbol{\Theta}} = \{(\widetilde{\pi}_k, \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^K$.
E-step : Compute, for $k = 1, \ldots, K$ and $i = 1, \ldots, n$
    $\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_i | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}, \quad n_k = \sum_{i=1}^{n} \gamma_{ki}.$    (4.21)
M-step : Compute, for $k = 1, \ldots, K$
    $\widetilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k)\pi_k,$    (4.22)
    $\widetilde{\boldsymbol{\mu}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i + (1 - \alpha_k)\boldsymbol{\mu}_k,$    (4.23)
    $\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T$
        $+ (1 - \alpha_k) \left( \boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right).$    (4.24)
Postprocessing: Normalize $\{\widetilde{\pi}_k\}_{k=1}^K$ so that they sum to 1,
    and ensure $\{\widetilde{\boldsymbol{\Sigma}}_k\}_{k=1}^K$ is positive semi-definite.
```

as follows.

Proposition 6. *Let* $L(\widetilde{\boldsymbol{\Theta}}) = \log f(\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) | \widetilde{\boldsymbol{\Theta}})$ *be the log-likelihood function,* $f(\widetilde{\boldsymbol{\Theta}})$ *be the prior distribution, and* $Q\big(\widetilde{\boldsymbol{\Theta}} | \widetilde{\boldsymbol{\Theta}}^{(m)}\big)$ *be the Q function in the m-th iteration of the EM iteration. If*

$$Q\big(\widetilde{\boldsymbol{\Theta}} | \widetilde{\boldsymbol{\Theta}}^{(m)}\big) + \log f(\widetilde{\boldsymbol{\Theta}}) \geq Q\big(\widetilde{\boldsymbol{\Theta}}^{(m)} | \widetilde{\boldsymbol{\Theta}}^{(m)}\big) + \log f\big(\widetilde{\boldsymbol{\Theta}}^{(m)}\big),$$

then

$$L(\widetilde{\boldsymbol{\Theta}}) + \log f(\widetilde{\boldsymbol{\Theta}}) \geq L\big(\widetilde{\boldsymbol{\Theta}}^{(m)}\big) + \log f\big(\widetilde{\boldsymbol{\Theta}}^{(m)}\big).$$

Proof. See [80]. □

While classical EM algorithm requires many iterations to converge, we observe that the proposed EM adaptation usually settles down in very few iterations. To demonstrate this observation, we conduct experiments on different testing images. Figure 4.2 shows the result of one testing image. For all noise levels ($\sigma = 20$ to 100), PSNR increases as more iterations are applied and converges after about 4 iterations. We also observe that for most testing images, the improvement becomes marginal after one single iteration.

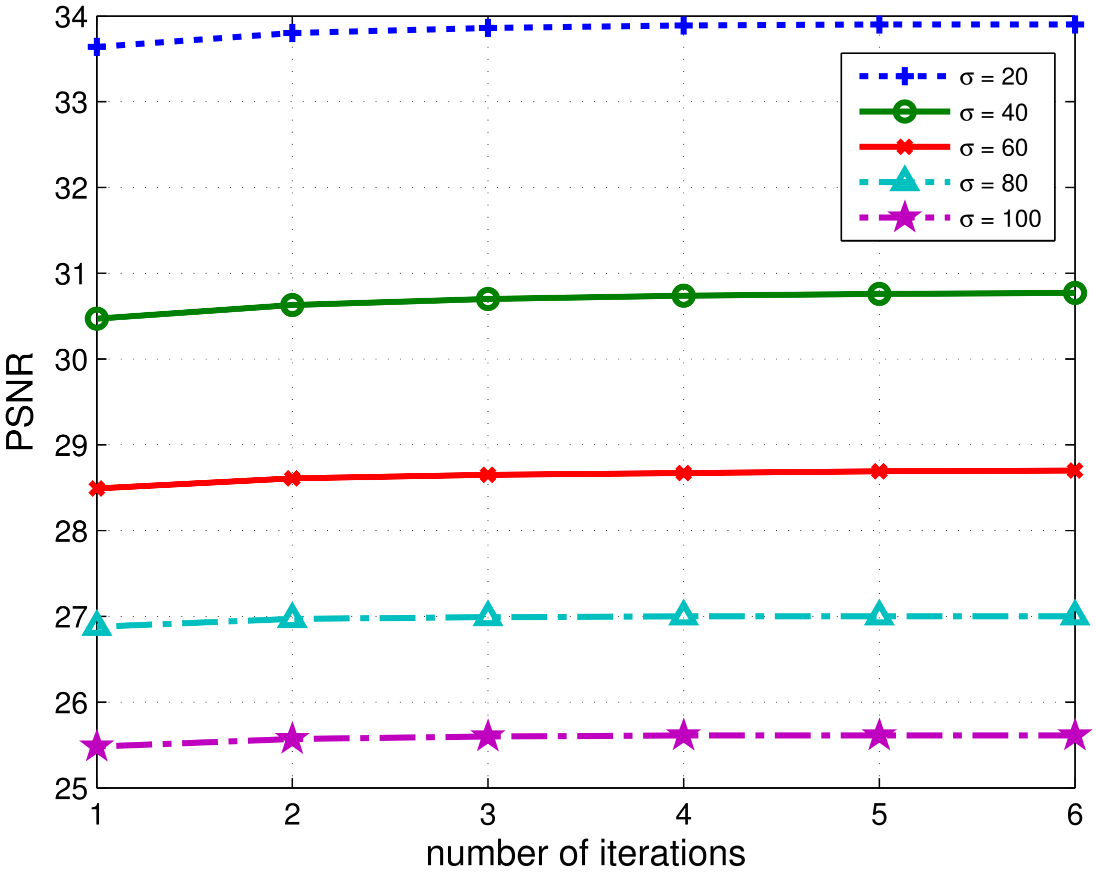


Figure 4.2: Image denoising using EM adaptation: The PSNR only improves marginally after the first iteration, confirming that the EM adaptation can typically be performed in a single iteration. Test image: House (256 × 256). Noise levels: $\sigma = 20, \ldots, 100$.

4.4 EM Adaptation for Denoising

The proposed Algorithm 3 works only when the training patches $\{\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n\}$ are from the *clean* ground-truth image $\boldsymbol{x}$. In this section, we discuss how to modify the EM adaptation algorithm for noisy images.

4.4.1 Adaptation to a Pre-filtered Image

To deal with the presence of noise, we adopt a two-stage approach similar to BM3D [11]. In the first stage, we apply an existing denoising algorithm to obtain a pre-filtered image. The adaptation is then applied to the pre-filtered image to generate an adapted prior. In the second stage, we apply the MAP denoising as described in Section II-A to obtain the final denoised image. However, since a pre-filtered image is not the same as the latent clean image, we must quantify the residual noise remaining in the pre-filtered image and revise the adaptation equations accordingly.

To this end, we let $\overline{\boldsymbol{x}}$ be the pre-filtered image. The distribution of the residue $\overline{\boldsymbol{x}} - \boldsymbol{x}$ is typically unknown but empirically we observe that it can be reasonably approximated by a single Gaussian. Thus, we model $(\overline{\boldsymbol{x}} - \boldsymbol{x}) \sim \mathcal{N}(\mathbf{0}, \widetilde{\sigma}^2 \boldsymbol{I})$, where $\widetilde{\sigma}^2 \overset{\text{def}}{=} \mathbb{E}\|\overline{\boldsymbol{x}} - \boldsymbol{x}\|^2$ is the variance of $\overline{\boldsymbol{x}}$. By incorporating the residual noise, we modify (4.21) as

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k + \widetilde{\sigma}^2 \boldsymbol{I})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_l \,|\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \widetilde{\sigma}^2 \boldsymbol{I})}, \tag{4.25}$$

and (4.24) as

$$\begin{aligned} \widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \left((\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T - \widetilde{\sigma}^2 \boldsymbol{I} \right) \\ + (1 - \alpha_k) \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \end{aligned} \tag{4.26}$$

It now remains to determine the parameter $\widetilde{\sigma}^2$.

4.4.2 Estimating $\widetilde{\sigma}^2$

By definition, $\widetilde{\sigma}^2$ is the variance of the pre-filtered image with the mean $\boldsymbol{x}$. In another point of view, $\widetilde{\sigma}^2$ is also the mean squared error of $\overline{\boldsymbol{x}}$ compared to $\boldsymbol{x}$. Therefore, if we would like to estimate $\widetilde{\sigma}^2$, we only need to estimate the amount of "noise" remaining in $\overline{\boldsymbol{x}}$. This is a challenging task because we do not have the ground truth $\boldsymbol{x}$.

In the absence of the clean image, one strategy is to use the Stein's Unbiased Risk Estimator (SURE) [83]. SURE provides a way for unbiased estimation of the true MSE. The analytical expression of SURE is

$$\widetilde{\sigma}^2 \approx \text{SURE} \overset{\text{def}}{=} \frac{1}{n}\|\boldsymbol{y} - \overline{\boldsymbol{x}}\|^2 - \sigma^2 + \frac{2\sigma^2}{n}\text{div}, \tag{4.27}$$

where div denotes the divergence of the denoising algorithm with respect to the noisy measurements. However, not all denoising algorithms have a closed form for the divergence term. To alleviate this issue, we adopt the Monte-Carlo SURE [84] to approximate the divergence. We shall not repeat Monte-Carlo SURE here but we summarize the steps in Algorithm 4.

Algorithm 4 Monte-Carlo SURE for Estimating $\widetilde{\sigma}^2$

```
  Input: noisy image $\boldsymbol{y} \in \mathbb{R}^n$, noise variance $\sigma^2$, a small $\delta = 0.01$.
  Output: $\widetilde{\sigma}^2$.
  Generate $\boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \in \mathbb{R}^n$.
  Construct $\boldsymbol{y}' = \boldsymbol{y} + \delta\boldsymbol{b}$.
  Apply a denoising algorithm on $\boldsymbol{y}$ and $\boldsymbol{y}'$ to get two pre-filtered images $\overline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}'$, respectively.
  Compute div $= \frac{1}{\delta}\boldsymbol{b}^T(\overline{\boldsymbol{x}}' - \overline{\boldsymbol{x}})$.
  Compute $\widetilde{\sigma}^2 = \text{SURE} \overset{\text{def}}{=} \frac{1}{n}\|\boldsymbol{y} - \overline{\boldsymbol{x}}\|^2 - \sigma^2 + \frac{2\sigma^2}{n}\text{div}$.
```

To demonstrate the effectiveness of Monte-Carlo SURE, we compare the estimates for $\widetilde{\sigma}/\sigma$ when we use the true MSE and Monte-Carlo SURE. As is observed in Figure 4.3, over a large range of noise levels, the Monte-Carlo SURE curves are quite similar to the true MSE curves. The pre-filtering method in Figure 4.3 is EPLL. For other methods such as BM3D, we have similar observations for different noise levels.

4.4.3 Estimating α_k

Besides the pre-filtering for noisy images, we should also determine the combination weight α_k for the EM adaptation. From the derivation of the algorithm, the combination weight $\alpha_k = \frac{n_k}{n_k+\rho}$ is determined by both the probabilistic count n_k

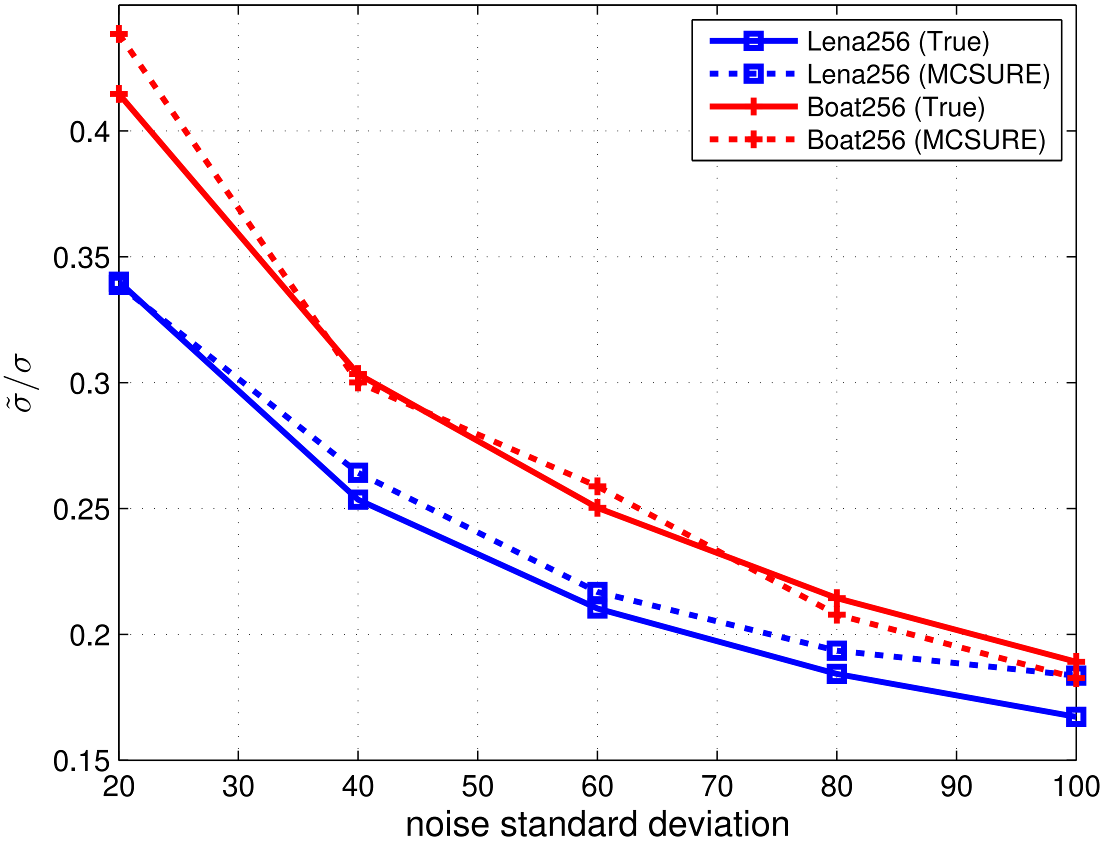


Figure 4.3: Comparison between the true MSE and Monte-Carlo SURE when estimating $\widetilde{\sigma}/\sigma$ over a large range of noise levels. The pre-filtering method is EPLL.

and the relevance factor ρ. The factor ρ is adjusted to allow different adaptation rates. For example, in the application of speaker verification [82,85], ρ is set to 16 and experiments show that the performance is insensitive to ρ being in the range of 8 and 20.

For our denoising task, we empirically determine the influence of ρ on the denoising performance. Given a pre-filtered image, we adjust ρ for the EM adaptation algorithm and check the corresponding denoising result. In Figure 4.4, we show how PSNR changes in terms of ρ. The PSNR curves indicate that for a testing image of 64 × 64, a large ρ for EM adaptation is better. As the testing images become large, we observe that the optimal ρ becomes small. Empirically, we find that ρ in the range of 1 and 10 works well for a variety of images (over

200 × 200) for different noise levels.

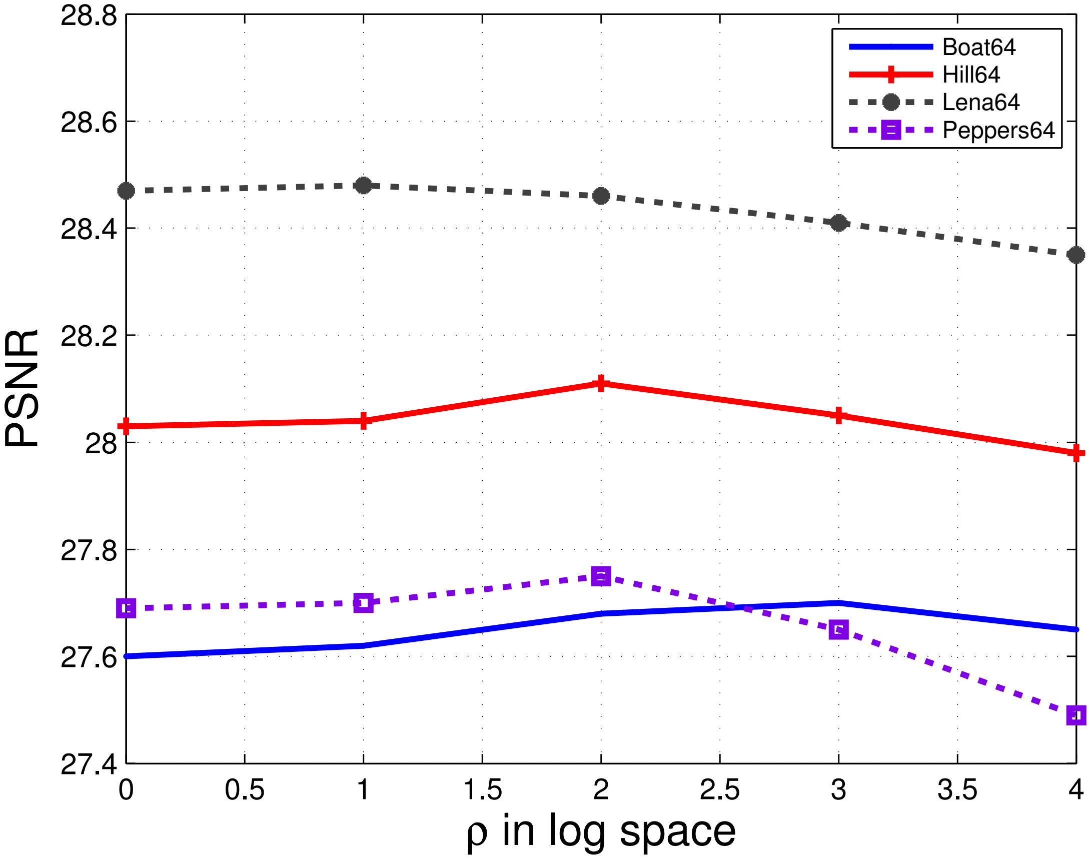


Figure 4.4: The effect of ρ on denoising performance. The pre-filtered image is used for EM adaptation algorithm. The testing images are of size 64 × 64 with noise $\sigma = 20$.

4.4.4 Computational Improvement

Finally, we comment on a simple but very effective way of improving the computational speed. If we take a closer look at the M-step in Algorithm 3, we observe that $\widetilde{\pi}_k$ and $\widetilde{\boldsymbol{\mu}}_k$ are easy to compute. However, $\widetilde{\boldsymbol{\Sigma}}_k$ is time-consuming to compute, because updating each of the K covariance matrices requires n time-consuming outer product operations $\sum_{i=1}^{n} \gamma_{ki}(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T$. Most previous works mitigate the problem by assuming that the covariance is diagonal [82,85,86]. However, this assumption is not valid in our case because image pixels (especially neighboring pixels) are correlated.

Our solution to this problem is shown in the following Proposition. The new result is an *exact* computation of (4.24) but with significantly less operations. The idea is to exploit the algebraic structure of the covariance matrix.

Proposition 7. *The full covariance adaptation in* (4.24) *can be simplified as*

$$\begin{aligned}\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T - \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T \\ + (1-\alpha_k)(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T).\end{aligned} \tag{4.28}$$

Proof. See Appendix B.4. □

The simplification is very rewarding because computing $\alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T$ does not involve $\widetilde{\boldsymbol{\mu}}_k$ and thus can be pre-computed for each component, which makes the computation of $\widetilde{\boldsymbol{\Sigma}}_k$ much more efficient. In Table 4.1, we list the averaging runtime when computing (4.24) and (4.28) for two image sizes.

Table 4.1: Runtime comparison between (4.24) and (4.28) for different image sizes.

	image size	Eq. (4.24)	Eqn. (4.28)	Speedup
		(original)	(ours)	factor
runtime (sec)	(64 × 64)	31.34	0.30	104.5
runtime (sec)	(128 × 128)	136.58	1.32	103.2

4.5 Experimental Results

In this section, we present experimental results for *single* and *example*-based image denoising. *Single* refers to using the single noisy image for training, whereas *example* refers to using an external reference image for training.

4.5.1 Experiment Settings

For comparison, we consider two state-of-the-art methods: BM3D [11] and EPLL [59]. For both methods, we run the original codes provided by the authors with the default parameters. The GMM prior in EPLL is learned from 2,000,000 randomly chosen 8×8 patches. For a fair comparison, we use the same GMM as the generic GMM for the proposed EM adaptation. We consider three versions of EM adaptation: (1) An oracle adaptation by adapting the generic prior to the ground-truth image, denoted as *aGMM-clean*; (2) A pre-filtered adaptation by adapting the generic prior to the EPLL result, denoted as *aGMM-EPLL*; (3) A pre-filtered adaptation by adapting the generic prior to the BM3D result, denoted as *aGMM-BM3D*. In the example-based image denoising, we adapt the generic prior to an example image and denote it as *aGMM-example*. We set the parameter $\rho = 1$ and experimental results show that the performance is insensitive to ρ being in the range of 1 and 10. We run denoising experiments on a variety of images and for a large range of noise standard deviations ($\sigma = 20, 40, 60, 80, 100$). To reduce the bias due to a particular noise realization, each reported PSNR result is averaged over 8 independent trials.

4.5.2 Single Image Denoising

We use 6 standard images of size 256×256, and 6 natural images of size 481×321 randomly chosen from [2] for the single image denoising experiments. The testing images are shown in Figure 4.5.

Figure 4.6 shows the denoising results for three standard testing images and Figure 4.7 shows the denoising results for three natural testing images. In comparison to the competing methods, our proposed method yields the highest PSNR values. The magnified areas indicate that the proposed method removes the noise while preserves image details better.

We report the detailed PSNR results for different noise variances in Table 4.2 for the standard images and in Table 4.3 for the natural images. Three key observations could be noted here. First, comparing aGMM-EPLL with EPLL, the denoising results from aGMM-EPLL are consistently better than EPLL with an average gain of about 0.3 dB. This validates the usefulness of the adapted GMM through the proposed EM adaptation. Second, the quality of the image used for EM adaptation affects the final denoising performance. For example, it is obvious that using the ground-truth clean image for EM adaptation is much better than using the denoised images such as the EPLL or BM3D denoised image. In some cases, aGMM-BM3D yields larger PSNR values than aGMM-EPLL due to the fact that the denoised image from BM3D is better than that from EPLL. Third, the PSNR differences between the oracle case and EPLL for standard images are larger than those for natural images. The reason is that the generic prior in EPLL is already learned from thousands of natural image patches, and thus the adaptation to another natural image may not further improve the performance much.

4.5.3 External Image Denoising

In this subsection, we evaluate the denoising performance when an *example* image is available for EM adaptation. An example image refers to a clean image and is relevant to the noisy image of interest. In [71,72], it is shown that obtaining reference images is feasible in some scenarios such as text images and face images. We consider the following three scenarios for our experiments.

1. Flower image denoising: We use the 102 flowers dataset from [3] which consists of 102 different categories of flowers. We randomly pick one category and then sample two flower images: one as the testing image with additive i.i.d. Gaussian noise and the other as the example image for the EM adaptation.

2. Face image denoising: We use the FEI face dataset from [4] which consists of 100 aligned and frontal face images of size 260×360. We randomly pick one face image as the image of interest. We then randomly sample another image from the dataset and treat it as the example image for our EM adaptation.

3. Text image denoising: To prepare for this scenario, we randomly crop a 200×200 region from a document and add noise to it. We then crop another 200×200 region from a very different document and use it as the example image.

In Figure 4.8, 4.9 and 4.10, we show the denoising results for the three different scenarios. As shown, the example images in the second column are similar but differ from the testing images. We compare the three denoising methods. The major difference lies in how the default GMM is adapted: In EPLL there is no EM adaptation, *i.e.*, the default generic GMM is used for denoising. In aGMM-example the default GMM is adapted to the example image while in aGMM-clean the default GMM is adapted to the ground truth image. As observed, the oracle aGMM-clean yields the best denoising performance. aGMM-example outperforms the benchmark EPLL (generic GMM) denoising algorithm both visually and objectively. For example, on average, it is 0.28 dB better in the Flower scenario, 0.78 dB better in the Face scenario, and 1.57 dB better in the Text scenario.

Since the default generic GMM is learned from thousands of natural image patches, it serves as a good prior if the testing image is also a natural image. In other words, if the testing image is quite different from a natural image, the generic GMM will perform worse to assist in denoising. However, if we have an example image for EM adaptation, the adapted GMM will produce significantly better results than the non-adapted GMM. To validate this, we further consider denoising a text image when we add excessive noise to the test image. In Figure 4.11 (a) and (b), we show the clean text image and its noisy version ($\sigma = 80$). In (c), we show the example text image with bold-faced words, which is similar but still differs much from (a). From Figure 4.11 (d) to (i), we show all the denoising results. Among all of them, the oracle case (f), which adapts the generic GMM using the ground-truth image, yields the highest PSNR. The other three adaptation-based methods (g)-(i) also perform well, and outperform the baseline methods BM3D and EPLL both visually and objectively. For example, aGMM-example and aGMM-EPLL outperform EPLL by 1.56 dB and 0.77 dB, respectively. The extraordinary result in (g) indicates that in the scenario of text image denoising, even a less similar example image can help adapt the generic GMM much and improve the denoising performance significantly.

4.5.4 Complexity Analysis

Our current implementation is in MATLAB (single thread) and we use an Intel Core i7-3770 CPU with 8 GB RAM. The runtime is about 66 seconds to denoise an image of size 256 × 256, where the EM adaptation part takes about 14 seconds while the MAP denoising part takes about 52 seconds. It is worth pointing out that the simplication in (4.28) in Section IV-D has significantly improved the computational efficiency for EM adaptation.

4.6 Conclusion

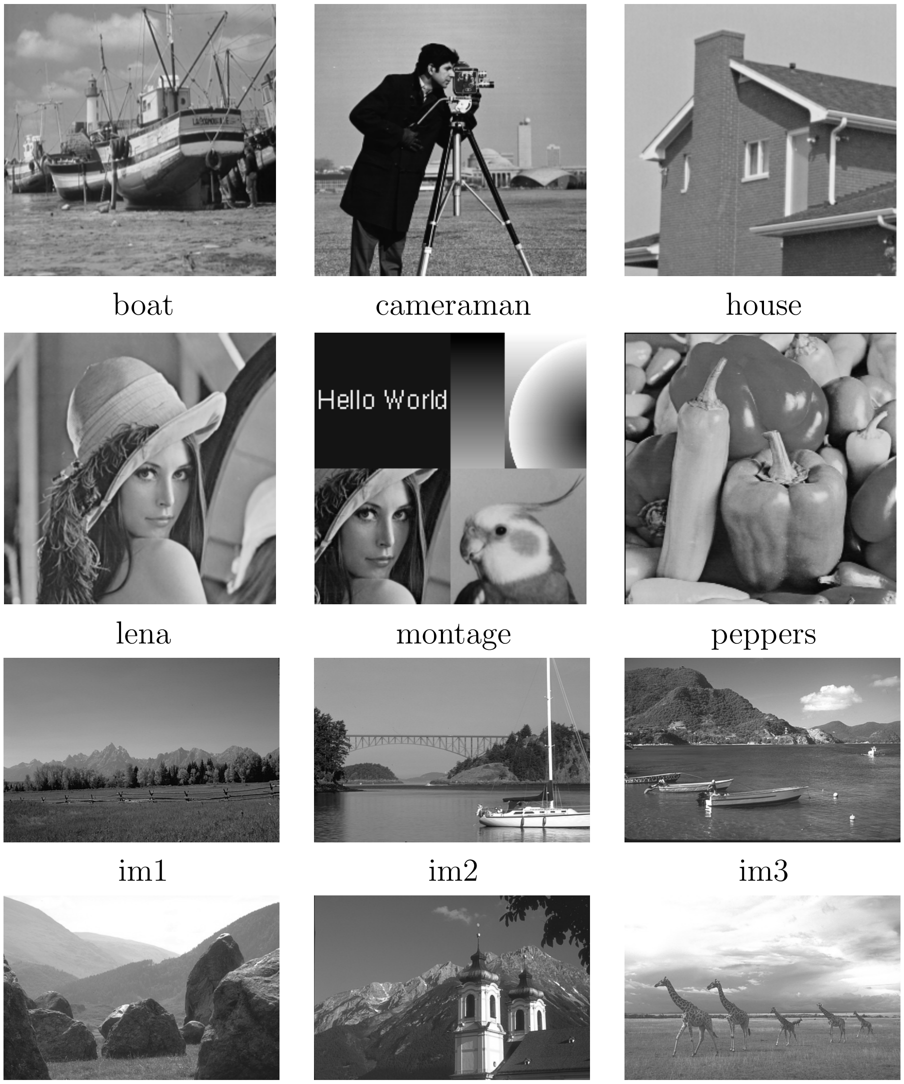
We proposed an EM adaptation method to learn effective image priors. The proposed algorithm is rigorously derived from the Bayesian hyper-prior perspective and is further simplified to reduce the computational complexity. In the absence of the latent clean image, we proposed modifications of the algorithm and analyzed how some internal parameters can be automatically estimated. The adapted prior from the EM adaptation better captures the prior distribution of the image of interest and is consistently better than the un-adapted generic one. In the context of image denoising, experimental results demonstrate its superiority over some existing denoising algorithms such as EPLL and BM3D. Future work includes its extended work on video denoising and other restoration tasks such as deblurring and inpainting.

4.7 Acknowledgement

This chapter, in part, is a reprint of a published conference paper and a submitted journal paper:

S. H. Chan, E. Luo, and T. Q. Nguyen, "Adaptive Patch-based Image Denoising by EM-adaptation," in *Proc. IEEE Global Conf. Signal and Information Process. (GlobalSIP'15)*, Dec. 2015.

E. Luo, S. H. Chan, and T. Q. Nguyen, "Adaptive Image Denoising by Mixture Adaptation," submitted to *IEEE Trans. Image Process. (TIP'16)*, Jan. 2016.



boat cameraman house

lena montage peppers

im1 im2 im3

im4 im5 im6

Figure 4.5: Test images for single image denoising. [Top] standard images of size 256 × 256; [Bottom] natural images of size 481 × 321 [2]

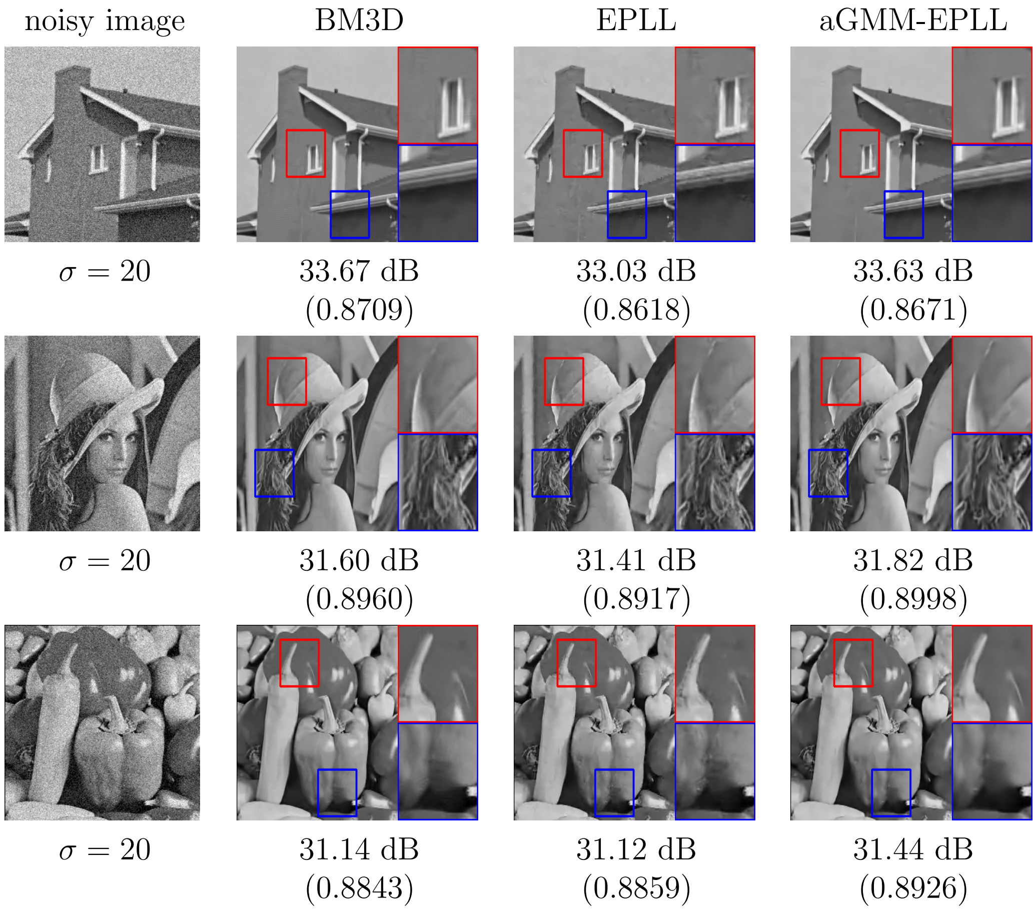


Figure 4.6: Single image denoising by using the denoised image for EM adaptation: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The testing images are standard images of size 256 × 256.

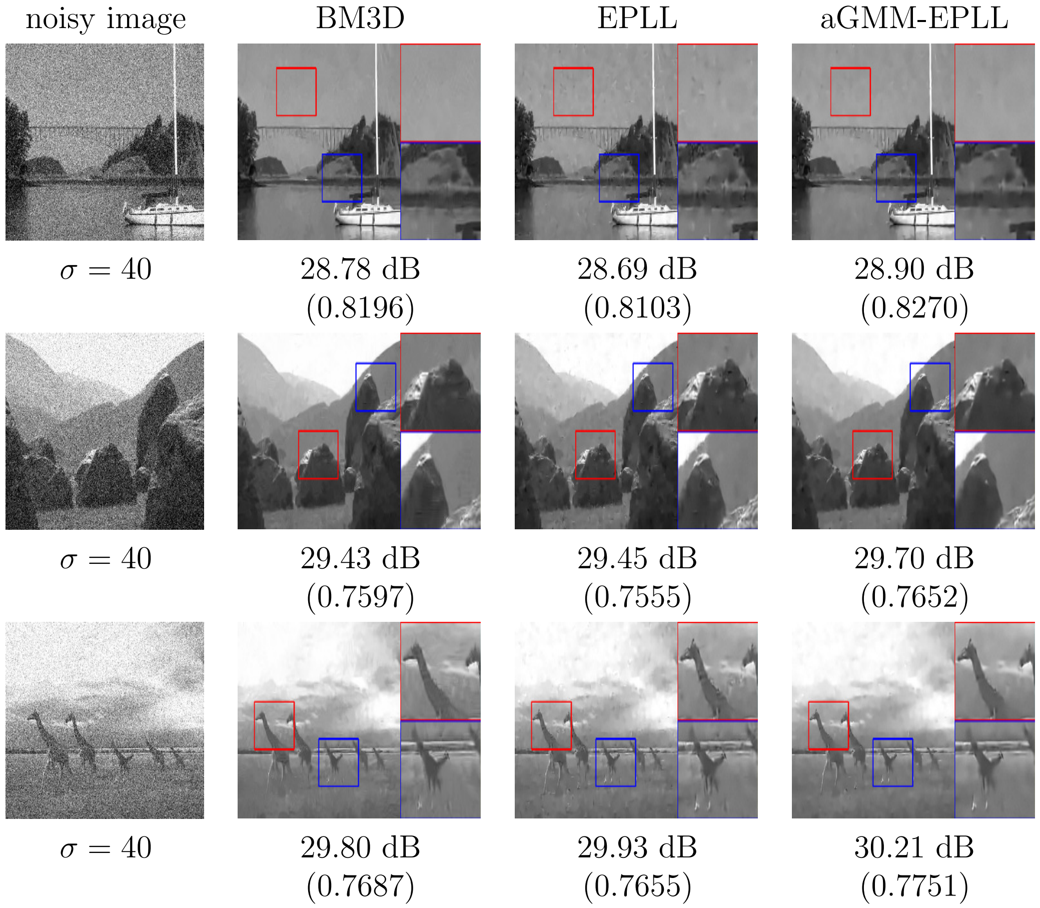


Figure 4.7: Single image denoising by using the denoised image for EM adaptation: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The testing images are natural images of size 481 × 321.

Table 4.2: PSNR results for standard images of size 256 × 256. The PSNR value for each noise level is averaged over 8 independent trials to reduce the bias due to a particular noise realization.

		BM3D	aGMM-BM3D	EPLL	aGMM-EPLL	aGMM-clean
Airplane	$\sigma = 20$	30.44	**30.77**	30.57	**30.87**	31.28
	$\sigma = 40$	26.45	**27.09**	27.00	**27.16**	27.48
	$\sigma = 60$	**25.15**	25.09	25.14	**25.24**	25.50
	$\sigma = 80$	**23.85**	23.72	23.74	**23.83**	24.00
	$\sigma = 100$	**22.82**	22.60	22.61	**22.66**	22.80
Boat	$\sigma = 20$	29.69	**29.90**	29.83	**30.00**	30.39
	$\sigma = 40$	26.09	**26.57**	26.46	**26.60**	26.86
	$\sigma = 60$	24.58	**24.65**	24.69	**24.77**	25.01
	$\sigma = 80$	**23.40**	23.36	23.41	**23.46**	23.69
	$\sigma = 100$	**22.64**	22.56	22.58	**22.61**	22.76
Cameraman	$\sigma = 20$	30.28	**30.33**	30.21	**30.38**	31.09
	$\sigma = 40$	26.78	**27.29**	26.96	**27.25**	27.76
	$\sigma = 60$	25.35	**25.42**	25.24	**25.52**	26.07
	$\sigma = 80$	**24.05**	24.04	23.90	**24.14**	24.66
	$\sigma = 100$	**23.05**	22.88	22.79	**22.94**	23.41
House	$\sigma = 20$	33.67	**33.81**	33.03	**33.63**	34.33
	$\sigma = 40$	30.49	**30.85**	29.94	**30.64**	31.31
	$\sigma = 60$	**28.88**	28.73	27.97	**28.57**	29.19
	$\sigma = 80$	**27.12**	26.95	26.34	**26.87**	27.28
	$\sigma = 100$	**25.92**	25.70	25.33	**25.67**	26.01
Lena	$\sigma = 20$	31.60	**31.76**	31.41	**31.82**	32.37
	$\sigma = 40$	27.83	**28.18**	27.98	**28.25**	28.62
	$\sigma = 60$	**26.36**	26.16	26.03	**26.23**	26.51
	$\sigma = 80$	**25.05**	24.85	24.70	**24.91**	25.12
	$\sigma = 100$	**23.88**	23.76	23.58	**23.79**	23.96
Peppers	$\sigma = 20$	31.14	**31.40**	31.12	**31.44**	32.04
	$\sigma = 40$	27.42	**28.00**	27.70	**28.03**	28.43
	$\sigma = 60$	25.87	**25.98**	25.70	**26.06**	26.39
	$\sigma = 80$	24.43	**24.56**	24.25	**24.64**	24.92
	$\sigma = 100$	23.28	**23.30**	23.05	**23.39**	23.61
Average		26.59	**26.68**	26.44	**26.71**	27.09

Table 4.3: PSNR results for natural images of size 481 × 321. The PSNR value for each noise level is averaged over 8 independent trials to reduce the bias due to a particular noise realization.

		BM3D	aGMM-BM3D	EPLL	aGMM-EPLL	aGMM-clean
Im1	$\sigma = 20$	30.96	**31.08**	31.29	**31.30**	31.65
	$\sigma = 40$	28.64	**28.76**	28.60	**28.79**	29.03
	$\sigma = 60$	**27.58**	27.45	27.16	**27.42**	27.59
	$\sigma = 80$	**26.63**	26.39	26.17	**26.38**	26.57
	$\sigma = 100$	**25.93**	25.82	25.52	**25.83**	25.82
Im2	$\sigma = 20$	32.03	**32.10**	31.98	**32.18**	32.59
	$\sigma = 40$	28.91	**28.98**	28.74	**28.99**	29.28
	$\sigma = 60$	**27.44**	27.23	27.00	**27.33**	27.53
	$\sigma = 80$	**26.35**	26.23	25.87	**26.21**	26.34
	$\sigma = 100$	**25.33**	25.28	24.88	**25.24**	25.36
Im3	$\sigma = 20$	29.85	**30.02**	30.10	**30.14**	30.42
	$\sigma = 40$	27.03	**27.30**	27.26	**27.35**	27.57
	$\sigma = 60$	25.78	**25.82**	25.77	**25.94**	26.13
	$\sigma = 80$	**24.97**	24.90	24.75	**24.96**	25.16
	$\sigma = 100$	**24.05**	23.94	23.84	**23.95**	24.21
Im4	$\sigma = 20$	31.77	**31.81**	**31.98**	31.97	32.27
	$\sigma = 40$	29.38	**29.57**	29.35	**29.64**	29.84
	$\sigma = 60$	**28.35**	28.23	27.95	**28.30**	28.41
	$\sigma = 80$	**27.54**	27.34	26.97	**27.39**	27.46
	$\sigma = 100$	**26.70**	26.57	26.18	**26.61**	26.65
Im5	$\sigma = 20$	31.09	**31.42**	31.39	**31.53**	31.86
	$\sigma = 40$	27.98	**28.22**	28.18	**28.28**	28.55
	$\sigma = 60$	**26.55**	26.53	26.54	**26.65**	26.84
	$\sigma = 80$	**25.49**	25.40	25.29	**25.44**	25.60
	$\sigma = 100$	**24.90**	24.76	24.65	**24.81**	24.97
Im6	$\sigma = 20$	32.30	**32.48**	32.54	**32.64**	32.94
	$\sigma = 40$	29.84	**30.13**	29.83	**30.17**	30.39
	$\sigma = 60$	28.60	**28.69**	28.28	**28.70**	28.79
	$\sigma = 80$	**27.66**	27.60	27.19	**27.61**	27.66
	$\sigma = 100$	**26.78**	26.65	26.28	**26.65**	26.71
Average		27.88	**27.89**	27.72	**27.95**	28.14

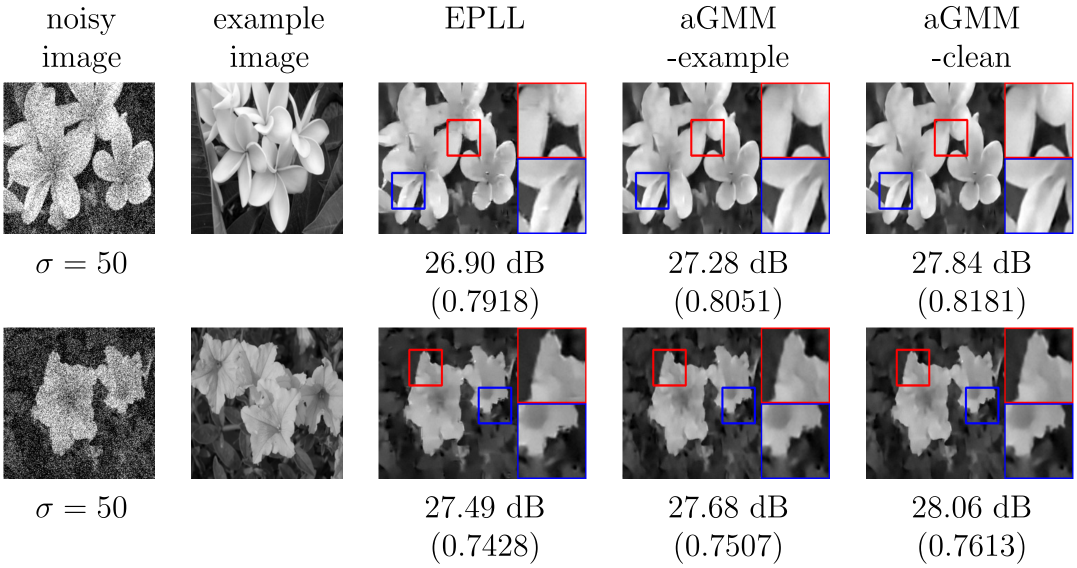


Figure 4.8: External image denoising by using an example image for EM adaptation: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The flower images are from the 102flowers dataset [3].

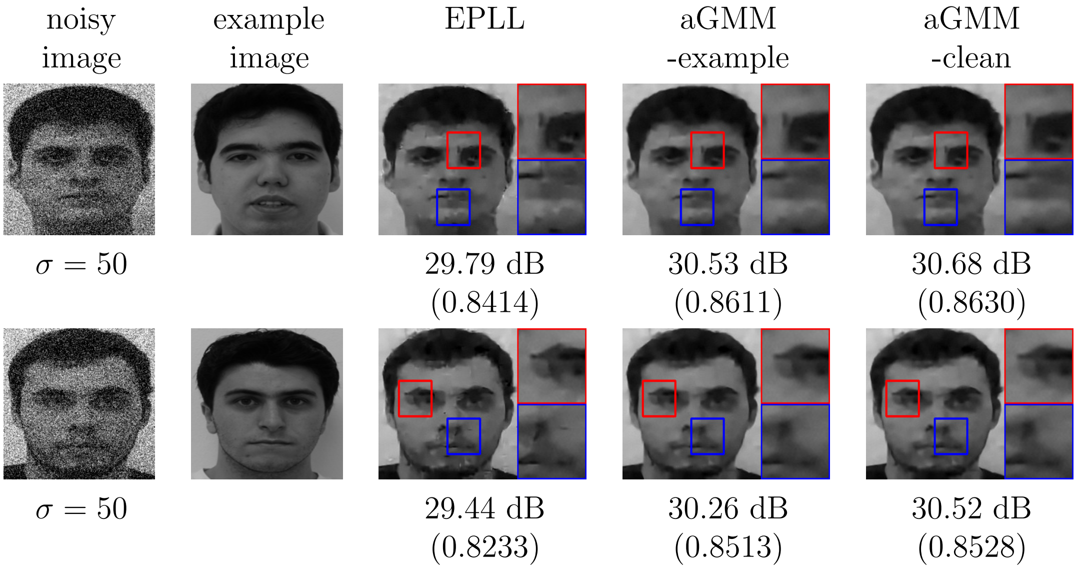


Figure 4.9: External image denoising by using an example image for EM adaptation: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The face images are from the FEI face dataset [4].

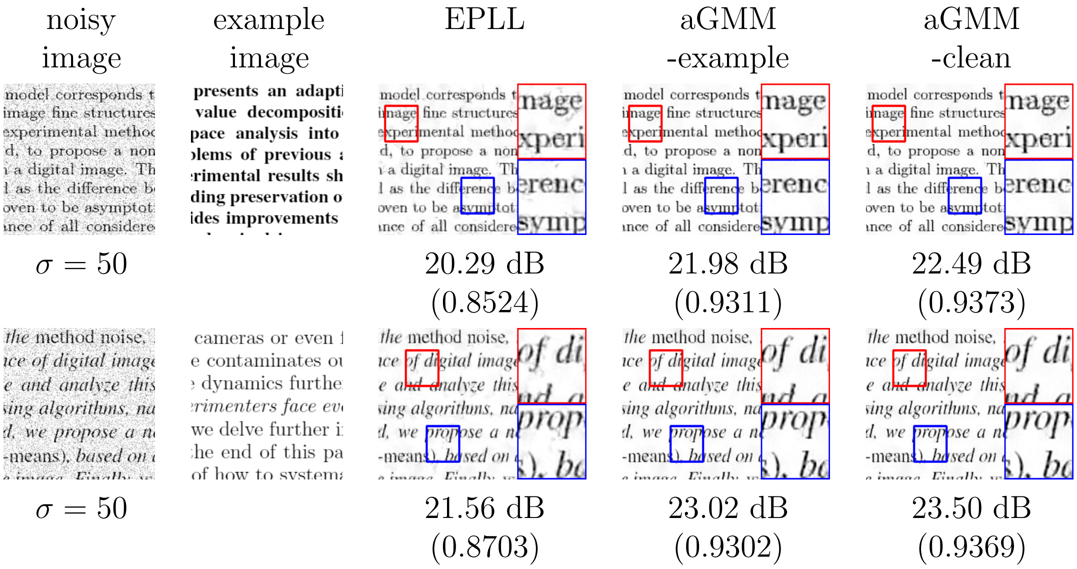
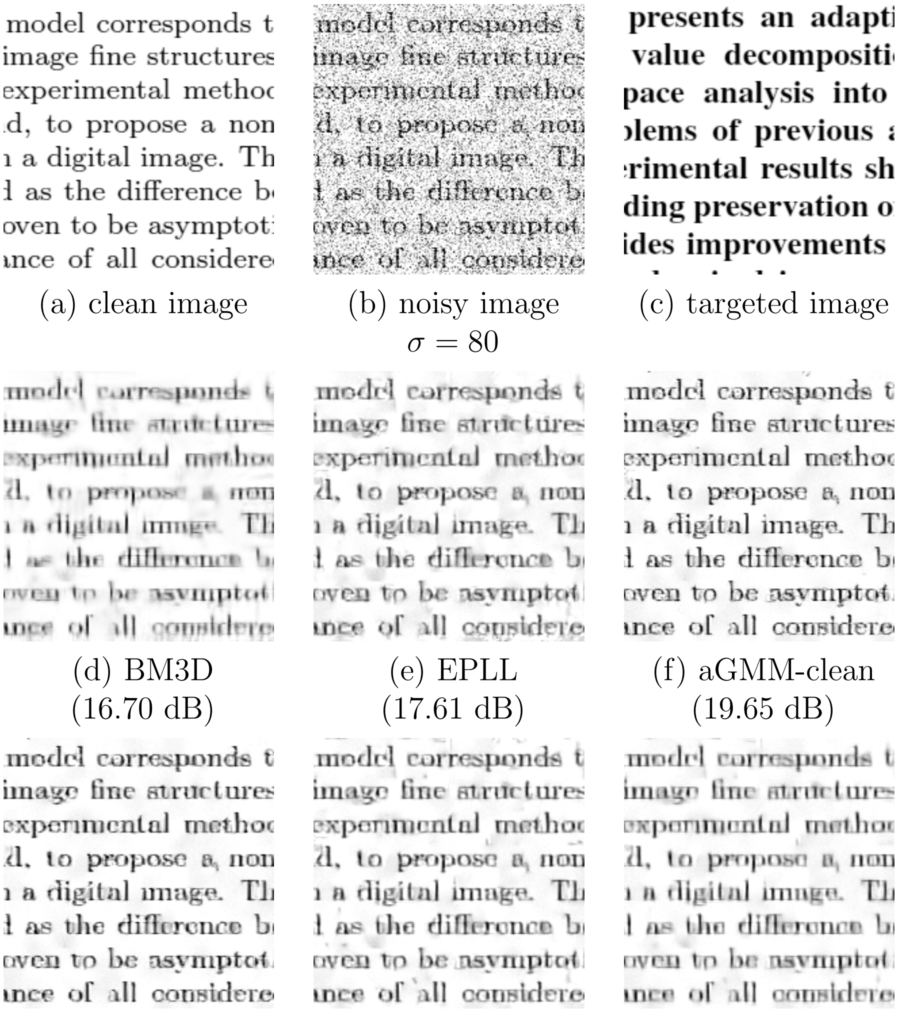


Figure 4.10: External image denoising by using an example image for EM adaptation: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The text images are cropped from randomly chosen documents.



(a) clean image

(b) noisy image $\sigma = 80$

(c) targeted image

(d) BM3D (16.70 dB)

(e) EPLL (17.61 dB)

(f) aGMM-clean (19.65 dB)

(g) aGMM-example (19.17 dB)

(h) aGMM-EPLL (18.38 dB)

(i) aGMM-BM3D (17.84 dB)

Figure 4.11: Text image denoising

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Image denoising is a long-lasting problem and it is also a test bed for a wide range of inverse problems in image processing. By advancing prior modeling we are effectively advancing our capability of solving more challenging recovery problems. In this thesis, we summarize our work on how to obtain effective priors from external image databases. For three different denoising applications with external databases, we explore effective priors, and propose statistical and adaptive patch-based image denoising algorithms.

The thesis begins with the introduction on patch-based image denoising and the discussion of internal prior (denoising) and external prior (denoising). Internal denoising based on single noisy images will soon reach the performance limit. External denoising is an alternative solution and has much room for improvement. In Chapter 2, 3 and 4, we show how to explore effective priors from external databases and develop adaptive denoising algorithms under different external settings.

In Chapter 2, we consider multiview image denoising. For each noisy view, the external database consists of its adjacent noisy views. We explore the non-local prior and present an adaptive non-local means denoising method, in which similar patches are carefully chosen from the database according to the local statistics of the estimates while dissimilar patches are discarded in order to achieve a trade-off between bias and variance.

In Chapter 3, we consider target-oriented image denoising. To facilitate the denoising of a noisy image of interest, we propose to exploit a targeted database instead of a generic database. To maximally utilize the targeted database, we show how to design an optimal linear denoising filter by exploring both the group sparsity prior and the localized Bayesian prior. The proposed method finds patches from the targeted database, which are then utilized to learn the basis matrix and the spectral coefficients of the optimal denoising filter.

In Chapter 4, we consider image denoising when only a generic database (as opposed to a targeted database) is available. We propose to take a generic prior learned from a generic database and then adapt it to the image of interest to create a specific prior. We rigorously derive the proposed Expectation-Maximization (EM) adaptation algorithm from a full Bayesian hyper-prior perspective. The proposed algorithm is further modified so that it also works when the image of interest is noisy.

In all Chapter 2, 3 and 4, we presented experimental results for a wide range of existing algorithms and our proposed methods. The comparative analyses show that our proposed methods have superior performance over existing methods both visually and quantitatively.

5.2 Future Work

There are a few suggestions for future work.

- In applying any patch-based image denoising algorithm, the denoising performance is intimately related to the capability of finding reference patches. Typically, one searches for similar patches by measuring patch similarity between a noisy patch and any other candidate patch based on some predefined metric. In Chapter 2, we proposed a robust distance metric for the specific multiview denoising setting. Yet in most denoising applications, the commonly used metric is the ℓ_2 distance (*i.e.*, Euclidean distance). However, this metric has a drawback in that the Euclidean distance does not necessarily capture the true patch similarity. In other words, two patches could be quite different, but their Euclidean distance might still be small. To improve the patch matching, two directions are worth exploring: (1) metric learning [87]. Basically we want to learn a similarity metric such that when a noisy query patch searches for its KNNs with this metric in a database, the rankings of the returned patches are the same as those when a clean query patch is used for patch searching. (2) deep learning [88–90]. The idea is to learn a deep encoder function which maps two noisy patches into an embedding space such that their relative distance could well "approximate" the relative distance between their clean versions.

- In Chapter 3, finding a good external database for the noisy image of interest is significant for the success of the proposed denoising algorithm. For an arbitrary noisy image to search for a targeted database, some image retrieval techniques in the computer vision area can be explored. In the case when it is hard to build a good targeted database or the returned external database is less satisfactory, one suggestion is to adaptively combine external denoising with internal denoising.

- In Chapter 4, the prior adaptation is applied to image denoising but it can be extended to the application of video denoising without much change. The proposed EM adaptation algorithm is rigorously derived for denoising, but it could be further modified so that it is also tailored for other inverse problems such as deblurring, inpainting and super-resolution.

Appendix A

Proofs of Chapter 3

A.1 Proof of Lemma 1

Proof. First, by writing $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{\eta}$ we get

$$\mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q} - \boldsymbol{p}\right\|_2^2\right] = \mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T(\boldsymbol{p} + \boldsymbol{n}) - \boldsymbol{p}\right\|_2^2\right].$$

Since $\boldsymbol{\eta}$ is i.i.d. Gaussian, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T(\boldsymbol{p} + \boldsymbol{\eta}) - \boldsymbol{p}\right\|_2^2\right] \\
=&\mathbb{E}\left[\left\|(\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T - \boldsymbol{I})\boldsymbol{p} + \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{\eta}\right\|_2^2\right] \\
=& \left\|\boldsymbol{U}(\boldsymbol{\Lambda} - \boldsymbol{I})\boldsymbol{U}^T\boldsymbol{p}\right\|_2^2 + \mathbb{E}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{\eta}\right\|_2^2\right] \\
=& \left\|\boldsymbol{U}(\boldsymbol{\Lambda} - \boldsymbol{I})\boldsymbol{U}^T\boldsymbol{p}\right\|_2^2 + \sigma^2\mathrm{Tr}\left(\boldsymbol{\Lambda}^2\right) \\
=& \sum_{i=1}^{n}\left[(1-\lambda_i)^2(\boldsymbol{u}_i^T\boldsymbol{p})^2 + \sigma^2\lambda_i^2\right],
\end{aligned}$$

which shows the desired result. □

A.2 Proof of Lemma 2

Proof. From (3.3), the optimization to be solved is

$$\begin{array}{ll} \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_d,\lambda_1,\ldots,\lambda_d}{\text{minimize}} & \sum_{i=1}^{d} \left[(1-\lambda_i)^2(\boldsymbol{u}_i^T\boldsymbol{p})^2 + \sigma^2\lambda_i^2\right] \\ \text{subject to} & \boldsymbol{u}_i^T\boldsymbol{u}_i = 1, \quad \boldsymbol{u}_i^T\boldsymbol{u}_j = 0. \end{array}$$

Since each term in the sum of the objective function is non-negative, we can consider the minimization over each individual term separately. This gives

$$\begin{array}{ll} \underset{\boldsymbol{u}_i,\lambda_i}{\text{minimize}} & (1-\lambda_i)^2(\boldsymbol{u}_i^T\boldsymbol{p})^2 + \sigma^2\lambda_i^2 \\ \text{subject to} & \boldsymbol{u}_i^T\boldsymbol{u}_i = 1. \end{array}$$

The Lagrangian function of the above equality-constrained problem is

$$\mathcal{L}(\boldsymbol{u}_i,\lambda_i,\beta) = (1-\lambda_i)^2(\boldsymbol{u}_i^T\boldsymbol{p})^2 + \sigma^2\lambda_i^2 + \beta(1-\boldsymbol{u}_i^T\boldsymbol{u}_i),$$

where β is the Lagrange multiplier. Differentiating $\mathcal{L}$ with respect to $\boldsymbol{u}_i$, λ_i and β yields

$$\frac{\partial\mathcal{L}}{\partial\lambda_i} = -2(1-\lambda_i)(\boldsymbol{u}_i^T\boldsymbol{p})^2 + 2\sigma^2\lambda_i \tag{A.1}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{u}_i} = 2(1-\lambda_i)^2(\boldsymbol{u}_i^T\boldsymbol{p})\boldsymbol{p} - 2\beta\boldsymbol{u}_i \tag{A.2}$$

$$\frac{\partial\mathcal{L}}{\partial\beta} = 1-\boldsymbol{u}_i^T\boldsymbol{u}_i. \tag{A.3}$$

Setting $\partial\mathcal{L}/\partial\lambda_i = 0$ yields $\lambda_i = (\boldsymbol{u}_i^T\boldsymbol{p})^2/\left((\boldsymbol{u}_i^T\boldsymbol{p})^2+\sigma^2\right)$. Substituting this λ_i into (A.2) and setting $\partial\mathcal{L}/\partial\boldsymbol{u}_i = 0$ yields

$$\frac{2\sigma^4(\boldsymbol{u}_i^T\boldsymbol{p})\boldsymbol{p}}{\left((\boldsymbol{u}_i^T\boldsymbol{p})^2+\sigma^2\right)^2} - 2\beta\boldsymbol{u}_i = 0. \tag{A.4}$$

There are two solutions. The first one is the trivial one: $\boldsymbol{u}_i$ = any unit vector orthogonal to $\boldsymbol{p}$, and $\beta = 0$. In this case, $\boldsymbol{u}_i^T\boldsymbol{p} = 0$ and $\beta\boldsymbol{u}_i = 0$ so that the left hand side of (A.4) is 0. The non-trivial solution is

$$\boldsymbol{u}_i = \boldsymbol{p}/\|\boldsymbol{p}\|, \quad \text{and} \quad \beta = \frac{\sigma^4\|\boldsymbol{p}\|^2}{(\|\boldsymbol{p}\|^2+\sigma^2)^2}, \tag{A.5}$$

which can be proved easily by substituting (A.5) into (A.4). Therefore, the denoising result is

$$\widehat{\boldsymbol{p}} = \boldsymbol{U}\left(\text{diag}\left\{\frac{\|\boldsymbol{p}\|^2}{\|\boldsymbol{p}\|^2+\sigma^2}, 0, \ldots, 0\right\}\right)\boldsymbol{U}^T\boldsymbol{q}.$$

□

A.3 Proof of Lemma 3

Proof. Let $\boldsymbol{u}_i$ be the ith column of $\boldsymbol{U}$. Then, (3.7) becomes

$$\begin{array}{ll} \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_d}{\text{minimize}} & \sum_{i=1}^d \|\boldsymbol{u}_i^T\boldsymbol{P}\|_2 \\ \text{subject to} & \boldsymbol{u}_i^T\boldsymbol{u}_i = 1, \quad \boldsymbol{u}_i^T\boldsymbol{u}_j = 0. \end{array} \tag{A.6}$$

Since each term in the sum of (A.6) is non-negative, we can consider each individual term

$$\begin{array}{ll} \underset{\boldsymbol{u}_i}{\text{minimize}} & \|\boldsymbol{u}_i^T\boldsymbol{P}\|_2 \\ \text{subject to} & \boldsymbol{u}_i^T\boldsymbol{u}_i = 1, \end{array}$$

which is equivalent to

$$\begin{array}{ll} \underset{\boldsymbol{u}_i}{\text{minimize}} & \|\boldsymbol{u}_i^T\boldsymbol{P}\|_2^2 \\ \text{subject to} & \boldsymbol{u}_i^T\boldsymbol{u}_i = 1. \end{array} \tag{A.7}$$

The constrained problem (A.7) can be solved by considering the Lagrange function,

$$\mathcal{L}(\boldsymbol{u}_i, \beta) = \|\boldsymbol{u}_i^T\boldsymbol{P}\|_2^2 + \beta(1-\boldsymbol{u}_i^T\boldsymbol{u}_i). \tag{A.8}$$

Taking derivatives $\frac{\partial\mathcal{L}}{\partial\boldsymbol{u}_i} = 0$ and $\frac{\partial\mathcal{L}}{\partial\beta} = 0$ yield

$$\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{u}_i = \beta\boldsymbol{u}_i, \quad \text{and} \quad \boldsymbol{u}_i^T\boldsymbol{u}_i = 1.$$

Therefore, $\boldsymbol{u}_i$ is the eigenvector of $\boldsymbol{P}\boldsymbol{P}^T$, and β is the corresponding eigenvalue. □

A.4 Proof of Lemma 4

Proof. First, by plugging $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{\eta}$ into BMSE we get

$$\begin{aligned} \text{BMSE} &= \mathbb{E}_{\boldsymbol{p}}\left[\mathbb{E}_{\boldsymbol{q}|\boldsymbol{p}}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T(\boldsymbol{p}+\boldsymbol{\eta}) - \boldsymbol{p}\right\|_2^2 \middle| \boldsymbol{p}\right]\right] \\ &= \mathbb{E}_{\boldsymbol{p}}\left[\boldsymbol{p}^T\boldsymbol{U}\left(\boldsymbol{I}-\boldsymbol{\Lambda}\right)^2\boldsymbol{U}^T\boldsymbol{p}\right] + \sigma^2\text{Tr}\left(\boldsymbol{\Lambda}^2\right). \end{aligned}$$

Let $\boldsymbol{L} \overset{\text{def}}{=} (\boldsymbol{I}-\boldsymbol{\Lambda})^2$, and recall the fact that for any random variable $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ and any matrix $\boldsymbol{A}$, it holds that $\mathbb{E}\left[\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}\right] = \mathbb{E}[\boldsymbol{x}]^T\boldsymbol{A}\mathbb{E}[\boldsymbol{x}] + \text{Tr}\left(\boldsymbol{A}\boldsymbol{\Sigma}_x\right)$. Therefore, the above BMSE can be simplified as

$$\begin{aligned} \text{BMSE} &= \boldsymbol{\mu}^T\boldsymbol{U}\boldsymbol{L}\boldsymbol{U}^T\boldsymbol{\mu} + \text{Tr}\left(\boldsymbol{U}\boldsymbol{L}\boldsymbol{U}^T\boldsymbol{\Sigma}\right) + \sigma^2\text{Tr}\left(\boldsymbol{\Lambda}^2\right) \\ &= \text{Tr}\left(\boldsymbol{L}\boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U} + \boldsymbol{L}\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U}\right) + \sigma^2\text{Tr}\left(\boldsymbol{\Lambda}^2\right). \end{aligned} \tag{A.9}$$

Differentiating BMSE with respect to $\boldsymbol{\Lambda}$ yields

$$\begin{aligned} \frac{\partial}{\partial\boldsymbol{\Lambda}}\text{BMSE} &= \frac{\partial\boldsymbol{L}}{\partial\boldsymbol{\Lambda}}\frac{\partial}{\partial\boldsymbol{L}}\text{BMSE} \\ &= -2(\boldsymbol{I}-\boldsymbol{\Lambda})\left(\boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U}\right) + 2\sigma^2\boldsymbol{\Lambda}. \end{aligned}$$

Setting $\frac{\partial}{\partial\boldsymbol{\Lambda}}\text{BMSE} = 0$ and assuming that the diagonal terms are dominant, we have

$$\boldsymbol{\Lambda} = \frac{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U}\right\}}{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U}\right\} + \sigma^2\boldsymbol{I}}. \tag{A.10}$$

□

A.5 Proof of Lemma 5

Proof. First, we write $\boldsymbol{\Sigma}$ in (3.21) in a matrix form

$$\begin{aligned}\boldsymbol{\Sigma} &= \left(\boldsymbol{P} - \boldsymbol{\mu}\mathbf{1}^T\right)\boldsymbol{W}\left(\boldsymbol{P} - \boldsymbol{\mu}\mathbf{1}^T\right)^T \\ &= \boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T - \boldsymbol{\mu}\mathbf{1}^T\boldsymbol{W}\boldsymbol{P}^T - \boldsymbol{P}\boldsymbol{W}\mathbf{1}\boldsymbol{\mu}^T + \boldsymbol{\mu}\mathbf{1}^T\boldsymbol{W}\mathbf{1}\boldsymbol{\mu}^T.\end{aligned}$$

It is not difficult to see that $\mathbf{1}^T\boldsymbol{W}\boldsymbol{P}^T = \boldsymbol{\mu}^T, \boldsymbol{P}\boldsymbol{W}\mathbf{1} = \boldsymbol{\mu}$ and $\mathbf{1}^T\boldsymbol{W}\mathbf{1} = 1$. Therefore,

$$\begin{aligned}\boldsymbol{\Sigma} &= \boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T - \boldsymbol{\mu}\boldsymbol{\mu}^T - \boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\mu}\boldsymbol{\mu}^T \\ &= \boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T - \boldsymbol{\mu}\boldsymbol{\mu}^T,\end{aligned}$$

which gives

$$\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T = \boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T. \tag{A.11}$$

Substituting (A.11) into (A.10), we have

$$\begin{aligned}\boldsymbol{\Lambda} &= \frac{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U}\right\}}{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U}\right\} + \sigma^2\boldsymbol{I}} \\ &= \frac{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T\boldsymbol{U}\right\}}{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{P}\boldsymbol{W}\boldsymbol{P}^T\boldsymbol{U}\right\} + \sigma^2\boldsymbol{I}} \\ &= \frac{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T\boldsymbol{U}\right\}}{\text{diag}\left\{\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T\boldsymbol{U}\right\} + \sigma^2\boldsymbol{I}} = \frac{\boldsymbol{S}}{\boldsymbol{S} + \sigma^2\boldsymbol{I}},\end{aligned} \tag{A.12}$$

where the divisions are element-wise. □

A.6 Proof of Lemma 6

Proof. To prove Lemma 6, we first apply the results in (A.9) and (A.12)

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{p}}\left[\mathbb{E}_{\boldsymbol{q}|\boldsymbol{p}}\left[\left\|\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{q}-\boldsymbol{p}\right\|_2^2\middle|\boldsymbol{p}\right]\right]\\ =&\mathrm{Tr}\left((\boldsymbol{I}-\boldsymbol{\Lambda})^2\boldsymbol{U}^T\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{U}+(\boldsymbol{I}-\boldsymbol{\Lambda})^2\boldsymbol{U}^T\boldsymbol{\Sigma}\boldsymbol{U}\right)+\sigma^2\mathrm{Tr}\left(\boldsymbol{\Lambda}^2\right)\\ =&\mathrm{Tr}(\boldsymbol{I}-\boldsymbol{\Lambda})^2\boldsymbol{S}+\sigma^2\mathrm{Tr}\left(\boldsymbol{\Lambda}^2\right)\\ =&\sum_{i=1}^d\left[(1-\lambda_i)^2s_i+\sigma^2\lambda_i^2\right]\\ =&\sum_{i=1}^d\left[(s_i+\sigma^2)\left(\lambda_i-\frac{s_i}{s_i+\sigma^2}\right)^2+\frac{s_i\sigma^2}{s_i+\sigma^2}\right].\end{aligned}$$

Adding the penalty term $\gamma\|\Lambda\mathbf{1}\|_\alpha$, the minimization problem with respect to λ_i becomes

$$\underset{\lambda_i}{\text{minimize}}\ \sum_{i=1}^d\left[(s_i+\sigma^2)\left(\lambda_i-\frac{s_i}{s_i+\sigma^2}\right)^2\right]+\gamma\|\Lambda\mathbf{1}\|_\alpha, \tag{A.13}$$

where $\gamma\|\Lambda\mathbf{1}\|_\alpha=\gamma\sum_{i=1}^d|\lambda_i|$ or $\gamma\sum_{i=1}^d\mathbb{1}(\lambda_i\neq 0)$ for $\alpha=1$ or 0. We note that when $\alpha=1$ or 0, (A.13) is the standard shrinkage problem [91], in which a closed form solution exists. Following from [32], the solutions are given by

$$\lambda_i=\max\left(\frac{s_i-\gamma/2}{s_i+\sigma^2},0\right),\qquad \text{for } \alpha=1,$$

and

$$\lambda_i=\frac{s_i}{s_i+\sigma^2}\mathbb{1}\left(\frac{s_i^2}{s_i+\sigma^2}>\gamma\right),\qquad \text{for } \alpha=0.$$

Even if we do not identify the above standard shrinkage problem, we could still show the proof as follows:
In $\sum_{i=1}^d\left[(1-\lambda_i)^2s_i+\sigma^2\lambda_i^2\right]$, the derivative with respect to each λ_i is $-2(1-\lambda_i)s_i+2\sigma^2\lambda_i$, where s_i is the ith diagonal entry in $\boldsymbol{S}$.
We first consider the case for ℓ_1 regularization: $\gamma\|\boldsymbol{\Lambda}\mathbf{1}\|_1=\gamma\sum_{i=1}^d|\lambda_i|$. For $\lambda_i\neq 0$, its derivative with respect to λ_i is $\gamma\mathrm{sign}(\lambda_i)$. Therefore, setting the sum of the

derivatives to be zero yields

$$-2(1-\lambda_i)s_i + 2\sigma^2\lambda_i + \gamma \text{sign}(\lambda_i) = 0. \tag{A.14}$$

Note that we have $s_i > 0$ and $\gamma > 0$, thus λ_i has to be inbetween 0 and 1, because otherwise the equation in (A.14) does not hold. For example, if $\lambda_i > 1$, we then have $-2(1-\lambda_i)s_i > 0$, $2\sigma^2\lambda_i > 0$, and $\gamma\text{sign}(\lambda_i) > 0$, making their sum to be larger than 0.

Solving equation (A.14) yields $\lambda_i = \frac{s_i - \gamma/2}{s_i + \sigma^2}$. Lastly if we include the case when $\lambda_i = 0$, we get the final result, which has a soft-thresholding form

$$\lambda_i = \max\left(\frac{s_i - \gamma/2}{s_i + \sigma^2}, 0\right).$$

We then consider the case for ℓ_0 regularization: $\gamma\|\mathbf{\Lambda 1}\|_0 = \gamma \sum\limits_{i=1}^{d} \mathbb{1}\left(\lambda_i \neq 0\right)$, where $\mathbb{1}$ is the indicator function. The ℓ_0 regularized BMSE can be written as $\sum\limits_{i=1}^{d} [(1-\lambda_i)^2 s_i + \sigma^2\lambda_i^2] + \gamma \sum\limits_{i=1}^{n} \mathbb{1}\left(\lambda_i \neq 0\right)$. Since each term in the sum is non-negative, minimizing the overall ℓ_0 regularized BMSE is equivalent to minimizing each individual term, which gives

$$\underset{\lambda_i}{\text{minimize}} \quad f(\lambda_i) \overset{\text{def}}{=} (1-\lambda_i)^2(\boldsymbol{u}_i^T\boldsymbol{p})^2 + \sigma^2\lambda_i^2 + \gamma\mathbb{1}\left(\lambda_i \neq 0\right).$$

We consider two cases, *i.e.*, $\lambda_i = 0$ and $\lambda_i \neq 0$.
If $\lambda_i = 0$, we have $f(\lambda_i) = s_i$.
If $\lambda_i \neq 0$, we have

$$f(\lambda_i) = \left[(1-\lambda_i)^2 s_i + \sigma^2\lambda_i^2\right] + \gamma. \tag{A.15}$$

It is easy to show that equation (A.15) is minimized when $\lambda_i = \frac{s_i}{s_i+\sigma^2}$, and its corresponding minimal value is $f(\lambda_i) = \frac{\sigma^2 s_i}{s_i+\sigma^2} + \gamma$.

Therefore, if $\frac{\sigma^2 s_i}{s_i+\sigma^2} + \gamma < s_i$, *i.e.*, $\frac{s_i^2}{s_i+\sigma^2} > \gamma$, we choose $\lambda_i = \frac{s_i}{s_i+\sigma^2}$, otherwise we choose $\lambda_i = 0$. The final result has a hard-thresholding form

$$\lambda_i = \frac{s_i}{s_i+\sigma^2}\mathbb{1}\left(\frac{s_i^2}{s_i+\sigma^2} > \gamma\right).$$

□

A.7 Proof of Theorem 1

To prove Theorem 1, we first need to define some notations. We let $\lambda_i(\widetilde{\boldsymbol{P}})$ be the ith singular value of $\widetilde{\boldsymbol{P}}$, and $\lambda_i(\widetilde{\boldsymbol{Q}})$ be ith singular value of $\widetilde{\boldsymbol{Q}}$. We also let $\boldsymbol{u}_i$ be the ith singular vector of $\widetilde{\boldsymbol{P}}$, and $\boldsymbol{v}_i$ be the ith singular vector of $\widetilde{\boldsymbol{Q}}$. Then, we can prove the following two lemmas.

Lemma 7. *Let $\lambda_i(\widetilde{\boldsymbol{P}})$ and $\lambda_i(\widetilde{\boldsymbol{Q}})$ be the ith singular values of $\widetilde{\boldsymbol{P}}$ and $\widetilde{\boldsymbol{Q}}$, respectively. Then,*

$$\left(\sum_{i=1}^{k}\left(\lambda_i(\widetilde{\boldsymbol{P}})-\lambda_i(\widetilde{\boldsymbol{Q}})\right)^2\right)^{1/2} \leq \left\|\Delta\widetilde{\boldsymbol{P}}\right\|_2. \tag{A.16}$$

Proof. By Horn and Johnson, Corollary 7.3.8(a) [92] □

Lemma 8. *Let $\boldsymbol{u}_i$ be the ith singular vector of $\widetilde{\boldsymbol{P}} \in \mathbb{R}^{n\times k}$, and let $\boldsymbol{v}_i$ be the ith singular vector of $\widetilde{\boldsymbol{Q}} \in \mathbb{R}^{n\times k}$, where $k < n$. If $\lambda_i(\widetilde{\boldsymbol{P}}) \neq \lambda_j(\widetilde{\boldsymbol{P}})$ for any $i \neq j$ and $i, j = 1, \ldots, k$, then*

$$\|\boldsymbol{v}_i - \boldsymbol{u}_i\|_2 \leq \rho\left\|\Delta\widetilde{\boldsymbol{P}}\right\|_2, \tag{A.17}$$

for some constants ρ, and $i = 1, \ldots, k$.

Proof. First, by Equation (10) of [Liu et al.], we have for $i = 1, \ldots, k$

$$\boldsymbol{v}_i = \boldsymbol{u}_i + \sum_{j\neq i, j=1}^{k}\left(\frac{\boldsymbol{u}_j^T[(\Delta\widetilde{\boldsymbol{P}})\,\widetilde{\boldsymbol{P}}^T + \widetilde{\boldsymbol{P}}\,(\Delta\widetilde{\boldsymbol{P}})^T]\boldsymbol{u}_i}{\lambda_i(\widetilde{\boldsymbol{P}})^2 - \lambda_j(\widetilde{\boldsymbol{P}})^2}\right)\boldsymbol{u}_j. \tag{A.18}$$

Therefore,

$$\begin{aligned}
&\|\boldsymbol{v}_i - \boldsymbol{u}_i\|_2 \\
&= \left\| \sum_{j\neq i, j=1}^{k} \left(\frac{\boldsymbol{u}_j^T [(\Delta\widetilde{\boldsymbol{P}})\,\widetilde{\boldsymbol{P}}^T + \widetilde{\boldsymbol{P}}\,(\Delta\widetilde{\boldsymbol{P}})^T]\boldsymbol{u}_i}{\lambda_i(\widetilde{\boldsymbol{P}})^2 - \lambda_j(\widetilde{\boldsymbol{P}})^2} \right) \boldsymbol{u}_j \right\|_2 \\
&\leq \left\| (\Delta\widetilde{\boldsymbol{P}})\,\widetilde{\boldsymbol{P}}^T + \widetilde{\boldsymbol{P}}\,(\Delta\widetilde{\boldsymbol{P}})^T \right\|_2 \sum_{j\neq i} \left(\frac{1}{\lambda_i(\widetilde{\boldsymbol{P}})^2 - \lambda_j(\widetilde{\boldsymbol{P}})^2} \right) \\
&\leq 2 \left\| \Delta\widetilde{\boldsymbol{P}} \right\|_2 \left\| \widetilde{\boldsymbol{P}} \right\|_2 \sum_{j\neq i} \left(\frac{1}{\lambda_i(\widetilde{\boldsymbol{P}})^2 - \lambda_j(\widetilde{\boldsymbol{P}})^2} \right).
\end{aligned}$$

Letting

$$\rho = 2 \left\| \widetilde{\boldsymbol{P}} \right\|_2 \sum_{j\neq i, j=1}^{k} \left(\frac{1}{\lambda_i(\widetilde{\boldsymbol{P}})^2 - \lambda_j(\widetilde{\boldsymbol{P}})^2} \right),$$

completes the proof. □

Now, we want to give a precise definition of $\widehat{\boldsymbol{p}}_P$ and $\widehat{\boldsymbol{p}}_Q$:

Definition 1. *The denoised signals $\widehat{\boldsymbol{p}}_P$ and $\widehat{\boldsymbol{p}}_Q$ are defined as*

$$\widehat{\boldsymbol{p}}_P = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T\boldsymbol{q} \quad \textit{and} \quad \widehat{\boldsymbol{p}}_Q = \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\boldsymbol{q},$$

where,

$$\begin{aligned}
\boldsymbol{S} &= \operatorname{diag}\left\{ \frac{\lambda_1(\widetilde{\boldsymbol{P}})^2}{\lambda_1(\widetilde{\boldsymbol{P}})^2 + \sigma^2}, \ldots, \frac{\lambda_k(\widetilde{\boldsymbol{P}})^2}{\lambda_k(\widetilde{\boldsymbol{P}})^2 + \sigma^2}, 0, \ldots, 0 \right\}, \\
\boldsymbol{R} &= \operatorname{diag}\left\{ \frac{\lambda_1(\widetilde{\boldsymbol{Q}})^2}{\lambda_1(\widetilde{\boldsymbol{Q}})^2 + \sigma^2}, \ldots, \frac{\lambda_k(\widetilde{\boldsymbol{Q}})^2}{\lambda_k(\widetilde{\boldsymbol{Q}})^2 + \sigma^2}, 0, \ldots, 0 \right\}.
\end{aligned}$$

Lemma 9. *The operator difference $\|\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T - \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\|_F$ is bounded as*

$$\|\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T - \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\|_F \leq \gamma \left\| \Delta\widetilde{\boldsymbol{P}} \right\|_F, \tag{A.19}$$

for some constant γ.

Proof. First, we observe that

$$\begin{aligned}
&\left\|\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T - \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\right\|_F \\
&= \left\|\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T - \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^T + \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^T - \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\right\|_F \\
&= \left\|\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T - \boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^T\right\|_F + \left\|\boldsymbol{V}\boldsymbol{S}\boldsymbol{V}^T - \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\right\|_F \\
&= \left\|(\boldsymbol{U}-\boldsymbol{V})\boldsymbol{S}(\boldsymbol{U}-\boldsymbol{V})^T\right\|_F + \left\|\boldsymbol{S}-\boldsymbol{R}\right\|_F .
\end{aligned}$$

The bound on the first term can be derived using Lemma 1.

$$\begin{aligned}
&\left\|(\boldsymbol{U}-\boldsymbol{V})\boldsymbol{S}(\boldsymbol{U}-\boldsymbol{V})^T\right\|_F \\
&= \left\|\sum_{i=1}^{k}\left(\frac{\lambda_i(\widetilde{\boldsymbol{P}})^2}{\lambda_i(\widetilde{\boldsymbol{P}})^2+\sigma^2}\right)(\boldsymbol{u}_i-\boldsymbol{v}_i)(\boldsymbol{u}_i-\boldsymbol{v}_i)^T\right\|_F \\
&\le \sum_{i=1}^{k}\left(\frac{\lambda_i(\widetilde{\boldsymbol{P}})^2}{\lambda_i(\widetilde{\boldsymbol{P}})^2+\sigma^2}\right)\left\|\boldsymbol{u}_i-\boldsymbol{v}_i\right\|^2 \\
&\le \sum_{i=1}^{k}\left(\frac{\lambda_i(\widetilde{\boldsymbol{P}})^2}{\lambda_i(\widetilde{\boldsymbol{P}})^2+\sigma^2}\right)\left(\rho\left\|\Delta\widetilde{\boldsymbol{P}}\right\|_2\right).
\end{aligned}$$

The bound on the second term is

$$
\begin{aligned}
&\|\boldsymbol{S}-\boldsymbol{R}\|_F\\
&=\left(\sum_{i=1}^{k}\left(\frac{\lambda_i(\widetilde{\boldsymbol{P}})^2}{\lambda_i(\widetilde{\boldsymbol{P}})^2+\sigma^2}-\frac{\lambda_i(\widetilde{\boldsymbol{Q}})^2}{\lambda_i(\widetilde{\boldsymbol{Q}})^2+\sigma^2}\right)^2\right)^{1/2}\\
&=\left(\sum_{i=1}^{k}\left(\frac{\sigma^2(\lambda_i(\widetilde{\boldsymbol{Q}})^2-\lambda_i(\widetilde{\boldsymbol{P}})^2)}{(\lambda_i(\widetilde{\boldsymbol{P}})^2+\sigma^2)(\lambda_i(\widetilde{\boldsymbol{Q}})^2+\sigma^2)}\right)^2\right)^{1/2}\\
&\le\left(\sum_{i=1}^{k}\left(\frac{(\lambda_i(\widetilde{\boldsymbol{Q}})^2-\lambda_i(\widetilde{\boldsymbol{P}})^2)}{\sigma^2}\right)^2\right)^{1/2}\\
&=\left(\sum_{i=1}^{k}\left(\frac{(\lambda_i(\widetilde{\boldsymbol{Q}})-\lambda_i(\widetilde{\boldsymbol{P}}))(\lambda_i(\widetilde{\boldsymbol{Q}})+\lambda_i(\widetilde{\boldsymbol{P}}))}{\sigma^2}\right)^2\right)^{1/2}\\
&\le\left(\frac{1}{\sigma^2}\max_{1\le i\le k}\left|\lambda_i(\widetilde{\boldsymbol{Q}})+\lambda_i(\widetilde{\boldsymbol{P}})\right|\right)\left(\sum_{i=1}^{k}\left(\lambda_i(\widetilde{\boldsymbol{Q}})-\lambda_i(\widetilde{\boldsymbol{P}})\right)^2\right)^{1/2}\\
&\le\left(\frac{1}{\sigma^2}\max_{1\le i\le k}\left|\lambda_i(\widetilde{\boldsymbol{Q}})+\lambda_i(\widetilde{\boldsymbol{P}})\right|\right)\left\|\Delta\widetilde{\boldsymbol{P}}\right\|_2,
\end{aligned}
$$

where the last inequality is due to Lemma 2. Finally, if we let

$$\gamma=\left(\frac{1}{\sigma^2}\max_{1\le i\le k}\left|\lambda_i(\widetilde{\boldsymbol{Q}})+\lambda_i(\widetilde{\boldsymbol{P}})\right|\right)+\rho\sum_{i=1}^{k}\left(\frac{\lambda_i(\widetilde{\boldsymbol{P}})^2}{\lambda_i(\widetilde{\boldsymbol{P}})^2+\sigma^2}\right),\tag{A.20}$$

then the proof is completed. □

Proof of Theorem 1:

Proof. First, it holds that

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\widehat{\boldsymbol{p}}_P-\widehat{\boldsymbol{p}}_Q\right\|^2\right]\\
&=\mathbb{E}\left[\left\|\boldsymbol{USU}^T\boldsymbol{q}-\boldsymbol{VRV}^T\boldsymbol{q}\right\|^2\right]\\
&=\left\|(\boldsymbol{USU}^T-\boldsymbol{VRV}^T)\boldsymbol{p}\right\|_2^2+\sigma^2\left\|\boldsymbol{USU}^T-\boldsymbol{VRV}^T\right\|_F^2\\
&\le\|\boldsymbol{p}\|_2^2\left\|\boldsymbol{USU}^T-\boldsymbol{VRV}^T\right\|_F^2+\sigma^2\left\|\boldsymbol{USU}^T-\boldsymbol{VRV}^T\right\|_F^2.
\end{aligned}
$$

Then, by Lemma 9 we have

$$\|\boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T - \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T\|_F \leq \gamma \left\|\Delta\widetilde{\boldsymbol{P}}\right\|_F.$$

Therefore,

$$\mathbb{E}\left[\|\widehat{\boldsymbol{p}}_1 - \widehat{\boldsymbol{p}}_2\|^2\right] \leq \left(\gamma^2\|\boldsymbol{p}\|_2^2 + \gamma^2\sigma^2\right)\left\|\Delta\widetilde{\boldsymbol{P}}\right\|_F^2.$$

□

A.8 Proof of Theorem 2

Proof. First, we observe that

$$\Pr\left[\min_{1\leq i\leq N}\|\boldsymbol{p}_i - \boldsymbol{p}_0\|_2 > \varepsilon\right] = \prod_{i=1}^{N}\Pr\left[\|\boldsymbol{p}_i - \boldsymbol{p}_0\|_2 > \varepsilon\right]. \tag{A.21}$$

Then,

$$\Pr\left[\|\boldsymbol{p} - \boldsymbol{p}_0\|_2 > \varepsilon\right] = 1 - \int_{\{\|\boldsymbol{p}-\boldsymbol{p}_0\|_2<\varepsilon\}} f(\boldsymbol{p})d\boldsymbol{p}. \tag{A.22}$$

By Taylor expansion on $f(\boldsymbol{p})$, we have

$$f(\boldsymbol{p}) = f(\boldsymbol{p}_0) + \nabla f(\boldsymbol{p}_0)^T(\boldsymbol{p} - \boldsymbol{p}_0) + \mathcal{O}(\|\boldsymbol{p} - \boldsymbol{p}_0\|^2). \tag{A.23}$$

Assuming $\varepsilon \ll 1$, and substitute (A.23) into (A.22), we have

$$\begin{aligned}&\int_{\{\|\boldsymbol{p}-\boldsymbol{p}_0\|_2<\varepsilon\}} f(\boldsymbol{p})d\boldsymbol{p}\\ &= \int_{\{\|\boldsymbol{p}-\boldsymbol{p}_0\|_2<\varepsilon\}} f(\boldsymbol{p}_0) + \nabla f(\boldsymbol{p}_0)^T(\boldsymbol{p} - \boldsymbol{p}_0)d\boldsymbol{p}.\end{aligned} \tag{A.24}$$

The first integral can be evaluated as

$$\begin{aligned}\int_{\{\|\boldsymbol{p}-\boldsymbol{p}_0\|_2<\varepsilon\}} f(\boldsymbol{p}_0)d\boldsymbol{p} &= f(\boldsymbol{p}_0)\int_{\{\|\boldsymbol{p}-\boldsymbol{p}_0\|_2<\varepsilon\}} d\boldsymbol{p}\\ &= f(\boldsymbol{p}_0)\frac{\varepsilon^n\pi^{n/2}}{\Gamma(\frac{n}{2}+1)},\end{aligned}$$

where the last equality holds because the integral is the volume of an n-dimensional sphere.

For the second integral, we let $\boldsymbol{x} = \boldsymbol{p} - \boldsymbol{p}_0$ and $\boldsymbol{a} = \nabla f(\boldsymbol{p}_0)$. Then, by transforming the Cartesian coordinate to the polar coordinate, we have

$$\begin{aligned}
&\int_{\{\|\boldsymbol{p}-\boldsymbol{p}_0\|_2<\varepsilon\}} \nabla f(\boldsymbol{p}_0)^T(\boldsymbol{p}-\boldsymbol{p}_0)d\boldsymbol{p} \\
&= \int_{\{\|\boldsymbol{x}\|_2<\varepsilon\}} \boldsymbol{a}^T\boldsymbol{x}d\boldsymbol{x} \\
&= \int_0^{\varepsilon}\int_{\phi_{n-1}=0}^{2\pi}\int_{\phi_{n-2}=0}^{\pi}\cdots\int_{\phi_1=0}^{2\pi} \\
&\quad \begin{bmatrix} a_1, \ldots, a_n \end{bmatrix} \begin{bmatrix} r\cos\phi_1 \\ r\sin\phi_1\cos\phi_2 \\ \ldots \\ r\sin\phi_1\ldots\sin\phi_{n-2}\cos\phi_{n-1} \\ r\sin\phi_1\ldots\sin\phi_{n-2}\sin\phi_{n-1} \end{bmatrix} \\
&\quad \times r^{n-1}\sin^{n-2}\phi_1\sin^{n-3}\phi_2\ldots\sin\phi_{n-2}drd\phi_1,\ldots,d\phi_{n-1}.
\end{aligned}$$

The last integral vanishes, because each term involves an integration of $\sin\phi_i$ or $\cos\phi_i$ over $[0,\pi]$.

Therefore, substituting into (A.24) yields

$$\int_{\{\|\boldsymbol{p}-\boldsymbol{p}_0\|_2<\varepsilon\}} f(\boldsymbol{p})d\boldsymbol{p} = f(\boldsymbol{p}_0)\frac{\varepsilon^n\pi^{n/2}}{\Gamma(\frac{n}{2}+1)},$$

and hence by letting $C = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}$, we have

$$\begin{aligned}
\prod_{i=1}^{N}\Pr\left[\|\boldsymbol{p}_i-\boldsymbol{p}_0\|_2 > \varepsilon\right] &= \left(1 - C\varepsilon^n f(\boldsymbol{p}_0)\right)^N \\
&= \exp\left\{N\log\left(1 - C\varepsilon^n f(\boldsymbol{p}_0)\right)\right\} \\
&\leq \exp\left\{-CN\varepsilon^n f(\boldsymbol{p}_0)\right\}.
\end{aligned}$$

Here, the last inequality holds because $\log(1-x) < -x$ for $0 < x < 1$. To see this, we consider the function $\phi(x) = \log(1-x) + x$. It holds that $\phi'(x) = -1/(1-x) + 1 < 0$ for $0 < x < 1$. So $\phi(x) < \phi(0)$, and hence $\log(1-x) < -x$ for $0 < x < 1$. □

Appendix B

Proofs of Chapter 4

B.1 Proof of Proposition 2

Proof. The minimization problem (4.5) can be split into two subproblems and solved in an alternating fashion. Given initial guesses $\boldsymbol{x}^{(0)}$ and $\boldsymbol{v}_i^{(0)}$, the algorithm alternatingly updates a sequence of $\boldsymbol{x}^{(m)}$ and $\boldsymbol{v}_i^{(m)}$ such that

$$\boldsymbol{v}_i^{(m+1)} = \arg\min_{\boldsymbol{v}_i} \Big\{ -\log f(\boldsymbol{v}_i) + \frac{\beta^{(m)}}{2}\|\boldsymbol{P}_i\boldsymbol{x}^{(m)} - \boldsymbol{v}_i\|^2 \Big\}, \tag{B.1}$$

$$\begin{aligned}\boldsymbol{x}^{(m+1)} &= \\ &\arg\min_{\boldsymbol{x}} \Big\{ \frac{n\sigma^{-2}}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + \frac{\beta^{(m)}}{2}\sum_{i=1}^{n}\|\boldsymbol{P}_i\boldsymbol{x} - \boldsymbol{v}_i^{(m+1)}\|^2 \Big\},\end{aligned} \tag{B.2}$$

where $\beta^{(m)}$ is an increasing sequence of penalty paramters.

For subproblem (B.1), if we further assume that $f(\boldsymbol{v}_i)$ is dominated by one of the components k_i^* where

$$k_i^* \overset{\text{def}}{=} \operatorname*{argmax}_k \ \pi_k \mathcal{N}(\boldsymbol{v}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{B.3}$$

we then relax (B.1) to

$$\arg\min_{\boldsymbol{v}_i} \left\{\frac{1}{2}\|\boldsymbol{v}_i - \boldsymbol{\mu}_{k_i^*}\|^2_{\boldsymbol{\Sigma}^{-1}_{k_i^*}} + \frac{\beta^{(m)}}{2}\|\boldsymbol{P}_i\boldsymbol{x}^{(m)} - \boldsymbol{v}_i\|^2\right\}, \tag{B.4}$$

the closed-form solution of which is

$$\boldsymbol{v}_i^{(m+1)} = \left(\beta^{(m)}\boldsymbol{\Sigma}_{k_i^*} + \boldsymbol{I}\right)^{-1}\left(\boldsymbol{\mu}_{k_i^*} + \beta^{(m)}\boldsymbol{\Sigma}_{k_i^*}\boldsymbol{P}_i\boldsymbol{x}^{(m)}\right). \tag{B.5}$$

Subproblem (B.2) is also a quadratic problem and has a closed-form solution as

$$\begin{gathered}\boldsymbol{x}^{(m+1)} = \\ \left(n\sigma^{-2}\boldsymbol{I} + \beta^{(m)}\sum_{i=1}^{n}\boldsymbol{P}_i^T\boldsymbol{P}_i\right)^{-1}\left(n\sigma^{-2}\boldsymbol{y} + \beta^{(m)}\sum_{i=1}^{n}\boldsymbol{P}_i^T\boldsymbol{v}_i^{(m+1)}\right).\end{gathered} \tag{B.6}$$

The minimizations in (B.1) and (B.2) are alternatingly solved until convergence. The final (B.6) gives the denoised image.

□

B.2 Proof of Proposition 3

Proof. Similarly as in the standard EM algorithm for GMM fitting, we first compute the probability that the i-th sample belongs to the k-th Gaussian component as

$$\gamma_{ki} = \frac{\pi_k^{(m)}\mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)})}{\sum_{l=1}^{K}\pi_l^{(m)}\mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})}, \tag{B.7}$$

where $\{(\pi_k^{(m)}, \boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)})\}_{k=1}^K$ are the GMM parameters in the m-th iteration and let $n_k \overset{\text{def}}{=} \sum_{i=1}^n \gamma_{ki}$. We can then approximate $\log f(\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n)|\widetilde{\boldsymbol{\Theta}})$ in (4.6) by the

Q function as follows

$$\begin{aligned}
Q(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{\Theta}}^{(m)}) =& \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ki}\log\Big(\widetilde{\pi}_k\mathcal{N}(\widetilde{\boldsymbol{p}}_i|\widetilde{\boldsymbol{\mu}}_k,\widetilde{\boldsymbol{\Sigma}}_k)\Big)\\
\doteq& \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ki}\Big(\log\widetilde{\pi}_k - \frac{1}{2}\log|\widetilde{\boldsymbol{\Sigma}}_k|\\
&-\frac{1}{2}(\widetilde{\boldsymbol{p}}_i-\widetilde{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{p}}_i-\widetilde{\boldsymbol{\mu}}_k)\Big)\\
=& \sum_{k=1}^{K} n_k(\log\widetilde{\pi}_k - \frac{1}{2}\log|\widetilde{\boldsymbol{\Sigma}}_k|)\\
&-\frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n} \gamma_{ki}(\widetilde{\boldsymbol{p}}_i-\widetilde{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{p}}_i-\widetilde{\boldsymbol{\mu}}_k),
\end{aligned} \tag{B.8}$$

where $\doteq$ indicates that some constant terms that are irrelevant to the parameters $\widetilde{\boldsymbol{\Theta}}$ are dropped. We further define two notations

$$\bar{\boldsymbol{\mu}}_k \overset{\text{def}}{=} \frac{1}{n_k}\sum_{i=1}^{n}\gamma_{ki}\widetilde{\boldsymbol{p}}_i, \quad \boldsymbol{S}_k \overset{\text{def}}{=} \sum_{i=1}^{n}\gamma_{ki}(\widetilde{\boldsymbol{p}}_i-\bar{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i-\bar{\boldsymbol{\mu}}_k)^T. \tag{B.9}$$

Using the equality $\sum_{i=1}^{n}\gamma_{ki}(\widetilde{\boldsymbol{p}}_i-\widetilde{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{p}}_i-\widetilde{\boldsymbol{\mu}}_k) = n_k(\widetilde{\boldsymbol{\mu}}_k-\bar{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{\mu}}_k-\bar{\boldsymbol{\mu}}_k)+\mathrm{tr}(\boldsymbol{S}_k\widetilde{\boldsymbol{\Sigma}}_k^{-1})$, we can rewrite the Q function as follows

$$\begin{aligned}
Q(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{\Theta}}^{(m)}) =& \sum_{k=1}^{K}\big\{n_k(\log\widetilde{\pi}_k - \frac{1}{2}\log|\widetilde{\boldsymbol{\Sigma}}_k|)\\
&-\frac{n_k}{2}(\widetilde{\boldsymbol{\mu}}_k-\bar{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{\mu}}_k-\bar{\boldsymbol{\mu}}_k) - \frac{1}{2}\mathrm{tr}(\boldsymbol{S}_k\widetilde{\boldsymbol{\Sigma}}_k^{-1})\big\}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1,\ldots,\widetilde{\boldsymbol{p}}_n) \propto \; \exp\big(Q(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{\Theta}}^{(m)}) + \log f(\widetilde{\boldsymbol{\Theta}})\big)\\
&= f(\widetilde{\boldsymbol{\Theta}}) \prod_{k=1}^{K} \Big\{ \widetilde{\pi}_k^{n_k} \, |\widetilde{\boldsymbol{\Sigma}}_k|^{-n_k/2}\\
&\exp\big(- \frac{n_k}{2} (\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k)^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} (\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k) - \frac{1}{2} \mathrm{tr}(\boldsymbol{S}_k \widetilde{\boldsymbol{\Sigma}}_k^{-1})\big)\Big\}\\
&= \prod_{k=1}^{K} \Big\{ \widetilde{\pi}_k^{v_k+n_k-1} |\widetilde{\boldsymbol{\Sigma}}_k|^{-(\varphi_k+n_k+d+2)/2} \exp\Big(- \frac{\tau_k + n_k}{2}\\
&(\widetilde{\boldsymbol{\mu}}_k - \frac{\tau_k \boldsymbol{\vartheta}_k + n_k \bar{\boldsymbol{\mu}}_k}{\tau_k + n_k})^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} (\widetilde{\boldsymbol{\mu}}_k - \frac{\tau_k \boldsymbol{\vartheta}_k + n_k \bar{\boldsymbol{\mu}}_k}{\tau_k + n_k})\Big)\\
&\exp\Big(- \frac{1}{2} \mathrm{tr}((\boldsymbol{\Psi}_k + \boldsymbol{S}_k\\
&+ \frac{\tau_k n_k}{\tau_k + n_k} (\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)^T) \widetilde{\boldsymbol{\Sigma}}_k^{-1})\Big)\Big\}.
\end{aligned} \tag{B.10}$$

Defining $v'_k \overset{\text{def}}{=} v_k + n_k, \varphi'_k \overset{\text{def}}{=} \varphi_k + n_k, \tau'_k \overset{\text{def}}{=} \tau_k + n_k, \boldsymbol{\vartheta}'_k \overset{\text{def}}{=} \frac{\tau_k \boldsymbol{\vartheta}_k + n_k \bar{\boldsymbol{\mu}}_k}{\tau_k + n_k}$, and $\boldsymbol{\Psi}'_k \overset{\text{def}}{=} \boldsymbol{\Psi}_k + \boldsymbol{S}_k + \frac{\tau_k n_k}{\tau_k + n_k} (\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)^T$, we will get

$$\begin{aligned}
&f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1,\ldots,\widetilde{\boldsymbol{p}}_n) \propto \textstyle\prod_{k=1}^{K} \Big\{ \widetilde{\pi}_k^{v'_k - 1} |\widetilde{\boldsymbol{\Sigma}}_k|^{-(\varphi'_k + d + 2)/2}\\
&\exp\big(- \tfrac{\tau'_k}{2} \left(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k\right)^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} \left(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k\right) - \tfrac{1}{2} \mathrm{tr}(\boldsymbol{\Psi}'_k \widetilde{\boldsymbol{\Sigma}}_k^{-1})\big)\Big\},
\end{aligned}$$

which completes the proof. □

B.3 Proof of Proposition 4

Proof. We ignore some irrelevant terms and get $\log f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1,\ldots,\widetilde{\boldsymbol{p}}_n) \doteq \sum_{k=1}^{K} \{(v'_k - 1) \log \widetilde{\pi}_k - \frac{(\varphi'_k + d + 2)}{2} \log |\widetilde{\boldsymbol{\Sigma}}_k| - \frac{\tau'_k}{2} (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k)^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k) - \frac{1}{2} \mathrm{tr}(\boldsymbol{\Psi}'_k \widetilde{\boldsymbol{\Sigma}}_k^{-1})\}$. Taking derivatives with respect to $\widetilde{\pi}_k$, $\widetilde{\boldsymbol{\mu}}_k$ and $\widetilde{\boldsymbol{\Sigma}}_k$ will yield the following solutions.

- Solution to $\widetilde{\pi}_k$.

  We form the Lagrangian

$$J(\widetilde{\pi}_k, \lambda) = \sum_{k=1}^{K} (v'_k - 1) \log \widetilde{\pi}_k + \lambda \left(\sum_{k=1}^{K} \widetilde{\pi}_k - 1 \right),$$

and the optimal solution satisfies

$$\frac{\partial J}{\partial \widetilde{\pi}_k} = \frac{v'_k - 1}{\widetilde{\pi}_k} + \lambda = 0.$$

It is easy to see that $\lambda = -\sum_{k=1}^{K}(v'_k - 1)$, and thus the solution to $\widetilde{\pi}_k$ is

$$\begin{aligned}
\widetilde{\pi}_k =& \frac{v'_k - 1}{\sum_{k=1}^{K}(v'_k - 1)} \\
=& \frac{(v_k - 1) + n_k}{(\sum_{k=1}^{K} v_k - K) + n} \\
=& \frac{n}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{n_k}{n} \\
&+ \frac{\sum_{k=1}^{K} v_k - K}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{v_k - 1}{\sum_{k=1}^{K} v_k - K}.
\end{aligned} \tag{B.11}$$

- Solution to $\widetilde{\boldsymbol{\mu}}_k$.

  We let

  $$\frac{\partial L}{\partial \widetilde{\boldsymbol{\mu}}_k} = -\frac{\tau'_k}{2}\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k) = 0, \tag{B.12}$$

  the solution of which is

  $$\begin{aligned}
  \widetilde{\boldsymbol{\mu}}_k =& \frac{\tau_k \boldsymbol{\vartheta}_k + n_k \bar{\boldsymbol{\mu}}_k}{\tau_k + n_k} \\
  =& \frac{1}{\tau_k + n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i + \frac{\tau_k}{\tau_k + n_k} \boldsymbol{\vartheta}_k.
  \end{aligned} \tag{B.13}$$

- Solution to $\widetilde{\boldsymbol{\Sigma}}_k$.

  We let

  $$\begin{aligned}
  \frac{\partial L}{\partial \widetilde{\boldsymbol{\Sigma}}_k} =& -\frac{\varphi'_k + d + 2}{2}\widetilde{\boldsymbol{\Sigma}}_k^{-1} + \frac{1}{2}\widetilde{\boldsymbol{\Sigma}}_k^{-1}\boldsymbol{\Psi}'_k\widetilde{\boldsymbol{\Sigma}}_k^{-1} \\
  &+ \frac{\tau'_k}{2}\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k)(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1} \\
  =& 0,
  \end{aligned}$$

which yields

$$(\varphi'_k + d + 2)\widetilde{\boldsymbol{\Sigma}}_k = \boldsymbol{\Psi}'_k + \tau'_k(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k)(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k)^T, \tag{B.14}$$

the solution of which is

$$\begin{aligned}
\widetilde{\boldsymbol{\Sigma}}_k =& \frac{\boldsymbol{\Psi}'_k + \tau'_k(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k)(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}'_k)^T}{\varphi'_k + d + 2} \\
=& \frac{\boldsymbol{\Psi}_k + \tau_k(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k)(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k)^T}{\varphi_k + d + 2 + n_k} \\
+& \frac{n_k(\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k)^T + \boldsymbol{S}_k}{\varphi_k + d + 2 + n_k} \\
=& \frac{n_k}{\varphi_k + d + 2 + n_k}\frac{1}{n_k}\sum_{i=1}^{n}\gamma_{ki}(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T \\
+& \frac{1}{\varphi_k + d + 2 + n_k}\left(\boldsymbol{\Psi}_k + \tau_k(\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)^T\right).
\end{aligned} \tag{B.15}$$

□

B.4 Proof of Proposition 7

Proof. Our first attempt is to expand the first term in (4.24).

$$\begin{aligned}
&\alpha_k\frac{1}{n_k}\sum_{i=1}^{n}\gamma_{ki}(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T \\
&= \alpha_k\frac{1}{n_k}\sum_{i=1}^{n}\gamma_{ki}(\widetilde{\boldsymbol{p}}_i\widetilde{\boldsymbol{p}}_i^T - \widetilde{\boldsymbol{p}}_i\widetilde{\boldsymbol{\mu}}_k^T - \widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{p}}_i^T + \widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{\mu}}_k^T) \\
&\triangleq \alpha_k\frac{1}{n_k}\sum_{i=1}^{n}\gamma_{ki}\widetilde{\boldsymbol{p}}_i\widetilde{\boldsymbol{p}}_i^T - (\widetilde{\boldsymbol{\mu}}_k - (1-\alpha_k)\boldsymbol{\mu}_k)\widetilde{\boldsymbol{\mu}}_k^T \\
&\quad - \widetilde{\boldsymbol{\mu}}_k(\widetilde{\boldsymbol{\mu}}_k - (1-\alpha_k)\boldsymbol{\mu}_k)^T + \alpha_k\widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{\mu}}_k^T \\
&= \alpha_k\frac{1}{n_k}\sum_{i=1}^{n}\gamma_{ki}\widetilde{\boldsymbol{p}}_i\widetilde{\boldsymbol{p}}_i^T - 2\widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{\mu}}_k^T \\
&\quad + (1-\alpha_k)(\boldsymbol{\mu}_k\widetilde{\boldsymbol{\mu}}_k^T + \widetilde{\boldsymbol{\mu}}_k\boldsymbol{\mu}_k^T) + \alpha_k\widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{\mu}}_k^T,
\end{aligned} \tag{B.16}$$

where $\triangleq$ holds because $\alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i = \widetilde{\boldsymbol{\mu}}_k - (1-\alpha_k)\boldsymbol{\mu}_k$ from (4.23).

We then expand the second term in (4.24)

$$\begin{aligned}
&(1-\alpha_k)\left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T\right) \\
&= (1-\alpha_k)(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T + \widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{\mu}}_k^T) \\
&\quad - (1-\alpha_k)(\boldsymbol{\mu}_k\widetilde{\boldsymbol{\mu}}_k^T + \widetilde{\boldsymbol{\mu}}_k\boldsymbol{\mu}_k^T).
\end{aligned} \tag{B.17}$$

Combining (B.16) and (B.17), we get a simplified (4.24) as

$$\begin{aligned}
\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T - \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T \\
+ (1-\alpha_k)(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T).
\end{aligned} \tag{B.18}$$

□

Bibliography

[1] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, “RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images,” *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[2] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. IEEE Intl. Conf. Computer Vision (ICCV'01)*, vol. 2, pp. 416–423, 2001.

[3] M. E. Nilsback and A. Zisserman, “Automated flower classification over a large number of classes,” in *Proc. Indian Conf. Computer Vision, Graphics and Image Process. (ICVGIP'08)*, pp. 722–729, Dec. 2008.

[4] C. E. Thomaz and G. A. Giraldi, “A new ranking method for principal components analysis and its application to face image analysis,” *Image and Vision Computing*, vol. 28, no. 6, pp. 902 – 913, 2010.

[5] R. Gonzalez and R. Woods, *Digital Image Processing*, Englewood Cliffs, NJ: Prentice-Hall, 2007.

[6] F. J. Anscombe, “The transformation of Poisson, binomial and negative-binomial data,” *Biometrika*, vol. 35, no. 3-4, pp. 246–254, 1948.

[7] S. Greenberg and D. Kogan, “Improved structure-adaptive anisotropic filter,” *Pattern Recognition Letters*, vol. 27, no. 1, pp. 59–65, 2006.

[8] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. IEEE Intl. Conf. Computer Vision (ICCV'98)*, pp. 839–846, Jan. 1998.

[9] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Trans. Image. Process.*, vol. 16, pp. 349–366, 2007.

[10] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*, vol. 2, MIT press Cambridge, MA, 1949.

[11] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3D transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “BM3D image denoising with shape-adaptive principal component analysis,” in *Signal Process. with Adaptive Sparse Structured Representations (SPARS’09)*, pp. 1–6, Apr. 2009.

[13] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli, “Image denoising using scale mixtures of gaussians in the wavelet domain,” *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[14] R. Eslami and H. Radha, “The contourlet transform for image denoising using cycle spinning,” in *Proc. of Asilomar Conf. on Signals, Systems and Computers*, vol. 2, pp. 1982–1986, Nov. 2003.

[15] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[16] R. Yan, L. Shao, and Y. Liu, “Nonlocal hierarchical dictionary learning using wavelets for image denoising,” *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4689–4698, Dec. 2013.

[17] A. Buades, B. Coll, and J. Morel, “A review of image denoising algorithms, with a new one,” *SIAM Multiscale Model and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[18] C. Kervrann and J. Boulanger, “Local adaptivity to variable smoothness for exemplar-based image regularization and representation,” *Intl. J. Computer Vision*, vol. 79, no. 1, pp. 45–69, 2008.

[19] K. Dabov, A. Foi, and K. Egiazarian, “Video denoising by sparse 3D transform-domain collaborative filtering,” in *Proc. 15th Euro. Signal Process. Conf.*, vol. 1, pp. 145–149, Sep. 2007.

[20] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “A nonlocal and shape-adaptive transform-domain collaborative filtering,” in *Proc. Intl. Workshop Local and Non-Local Approx. Image Process. (LNLA’08)*, pp. 1–8, Aug. 2008.

[21] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR ’09)*, pp. 2272–2279, Sep. 2009.

[22] L. Zhang, W. Dong, D. Zhang, and G. Shi, “Two-stage image denoising by principal component analysis with local pixel grouping,” *Pattern Recognition*, vol. 43, pp. 1531–1549, Apr. 2010.

[23] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620 – 1630, Apr. 2013.

[24] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image denoising using the higher order singular value decomposition," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 35, no. 4, pp. 849 – 862, Apr. 2013.

[25] M. Bertero and P. Boccacci, *Introduction to inverse problems in imaging*, CRC press, 2010.

[26] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'11)*, pp. 977–984, Jun. 2011.

[27] I. Mosseri, M. Zontak, and M. Irani, "Combining the power of internal and external denoising," in *Proc. Intl. Conf. Computational Photography (ICCP'13)*, pp. 1–9, Apr. 2013.

[28] H.C. Burger, C.J. Schuler, and S. Harmeling, "Learning how to combine internal and external denoising methods," *Pattern Recognition*, pp. 121–130, 2013.

[29] L. Zhang, S. Vaddadi, H. Jin, and S. Nayar, "Multiple view image denoising," in *Proc. IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR'09)*, pp. 1542–1549, Jun. 2009.

[30] S. Chan, D. Vo, and T. Nguyen, "Subpixel motion estimation without interpolation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 722–725, 2010.

[31] T. Brox, A. Bruhn, N. Papenberg, and J.Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. European Conference on Computer Vision (ECCV)*, pp. 25–36, 2004.

[32] S.H. Chan, R. Khoshabeh, K.B. Gibson, P.E. Gill, and T.Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

[33] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, "From local kernel to nonlocal multiple-model image denoising," *International Journal on Computer Vision*, vol. 86, pp. 1–32, 2010.

[34] Y. Lou, P. Favaro, S. Soatto, and A. Bertozzi, "Nonlocal similarity image filtering," in *Proc. International Conference on Image Analysis and Processing*, pp. 62–71, 2009.

[35] T. Thaipanich, B. Oh, P. Wu, and C. Kuo, “Adaptive nonlocal means algorithm for image denoising,” in *Proc. IEEE International Conference on Consumer Electronics (ICCE)*, pp. 417–418, Jan. 2010.

[36] T. Buades, Y. Lou, J. Morel, and Z. Tang, “A note on multi-image denoising,” in *Proc. IEEE Intl. Workshop on Local and Non-Local Approx. in Image Process. (LNLA’09)*, pp. 1–15, Aug. 2009.

[37] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, “Video denoising using separable 4-D nonlocal spatiotemporal transforms,” in *Proc. SPIE*, vol. 7870, 2011.

[38] C. Liu and W. Freeman, “A high-quality video denoising algorithm based on reliable motion estimation,” in *Proc. Euporpean Conference on Computer Vision (ECCV)*, pp. 706–719, 2010.

[39] J. Boulanger, C. Kervrann, and P. Bouthemy, “Space-time adaptation for patch-based image sequence restoration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1096–1102, Jun. 2007.

[40] R. Dugad and N. Ahuja, “Video denoising by combining Kalman and Wiener estimates,” in *Proc. IEEE Intl. Conf. on Image Process. (ICIP’99)*, vol. 4, pp. 152–156, 1999.

[41] C. Ercole, A. Foi, V. Katkovnik, and K. Egiazarian, “Spatio-temporal pointwise adaptive denoising in video: 3D non parametric approach,” in *Proc. Intl. Workshop Video Process. and Quality Metrics for Consumer Electronics*, 2005.

[42] V. Zlokolica and W. Philips, “Motion and detail adaptive denoising in video,” in *Proc. SPIE*, vol. 5298, pp. 403–412, 2004.

[43] N. Rajpoot, Z. Yao, and R. Wilson, “Adaptive wavelet restoration of noisy video sequences,” in *Proc. IEEE Intl. Conf. on Image Process. (ICIP’04)*, pp. 957–960, Oct. 2004.

[44] D. Rusanovskyy and K. Egiazarian, “Video denoising algorithm in sliding 3D DCT domain,” in *Proc. Advanced Concepts for Intelligent Vision Systems*, 2005.

[45] V. Katkovnik, “On adaptive local polynomial approximation with varying bandwidth,” in *Proc. IEEE Intl. Conf. on Acoustics, Speech and Signal Process. (ICASSP’98)*, vol. 4, pp. 2321–2324, May 1998.

[46] C. Liu, *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*, Ph.D. thesis, Massachusetts Institute of Technology, 2009.

[47] A. Buades, B. Coll, and J.M. Morel, "Denoising image sequences does not require motion estimation," in *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*, pp. 70–74, Sep. 2005.

[48] P. Milanfar, "A tour of modern image filtering," *IEEE Signal Process. Magazine*, vol. 30, pp. 106–128, Jan. 2013.

[49] N. Joshi, W. Matusik, E. Adelson, and D. Kriegman, "Personal photo enhancement using example images," *ACM Trans. Graph*, vol. 29, no. 2, pp. 1–15, Apr. 2010.

[50] L. Sun and J. Hays, "Super-resolution from internet-scale scene matching," in *Proc. IEEE Intl. Conf. Computational Photography (ICCP'12)*, pp. 1–12, Apr. 2012.

[51] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'08)*, pp. 1–8, Jun. 2008.

[52] M.K. Johnson, K. Dale, S. Avidan, H. Pfister, W.T. Freeman, and W. Matusik, "CG2Real: Improving the realism of computer generated images using a large collection of photographs," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 9, pp. 1273–1285, Sep. 2011.

[53] M. Elad and D. Datsenko, "Example-based regularization deployed to super-resolution reconstruction of a single image," *The Computer Journal*, vol. 18, no. 2-3, pp. 103–121, Sep. 2007.

[54] K. Dale, M. K. Johnson, K. Sunkavalli, W. Matusik, and H. Pfister, "Image restoration using online photo collections," in *Proc. Intl. Conf. Computer Vision (ICCV'09)*, pp. 2217–2224, Sep. 2009.

[55] I. Ram, M. Elad, and I. Cohen, "Image processing using smooth ordering of its patches," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2764–2774, Jul. 2013.

[56] E. Luo, S. H. Chan, S. Pan, and T. Q. Nguyen, "Adaptive non-local means for multiview image denoising: Searching for the right patches via a statistical approach," in *Proc. IEEE Intl. Conf. Image Process. (ICIP'13)*, pp. 543–547, Sep. 2013.

[57] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: Design of dictionaries for sparse representation," *Proceedings of SPARS*, vol. 5, pp. 9–12, 2005.

[58] S. Roth and M.J. Black, "Fields of experts," *Intl. J. Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.

[59] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. IEEE Intl. Conf. Computer Vision (ICCV'11)*, pp. 479–486, Nov. 2011.

[60] G. Yu, G. Sapiro, and S. Mallat, "Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2481–2499, May 2012.

[61] P. Milanfar, "Symmetrizing smoothing filters," *SIAM J. Imaging Sci.*, vol. 6, no. 1, pp. 263–284, 2013.

[62] H. Talebi and P. Milanfar, "Global image denoising," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 755–768, Feb. 2014.

[63] S. Cotter, B. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[64] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[65] S. Roth and M. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05), 2005*, vol. 2, pp. 860–867 vol. 2, Jun. 2005.

[66] D. Zoran and Y. Weiss, "Natural images, gaussian mixtures and dead leaves," *Advances in Neural Information Process. Systems (NIPS'12)*, vol. 25, pp. 1745–1753, 2012.

[67] S. M. Kay, "Fundamentals of statistical signal processing: Detection theory," 1998.

[68] A. Buades, B. Coll, and J. M. Morel, "Non-local means denoising," [Available online] http : //www.ipol.im/pub/art/2011/bcm_nlm/, 2011.

[69] E. Luo, S. Pan, and T. Q. Nguyen, "Generalized non-local means for iterative denoising," in *Proc. 20th Euro. Signal Process. Conf. (EUSIPCO'12)*, pp. 260–264, Aug. 2012.

[70] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'14)*, pp. 2774–2781, Jun. 2014.

[71] E. Luo, S. H. Chan, and T. Q. Nguyen, "Image denoising by targeted external databases," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Process. (ICASSP '14)*, pp. 2469–2473, May 2014.

[72] E. Luo, S. H. Chan, and T. Q. Nguyen, “Adaptive image denoising by targeted databases,” *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2167–2181, Jul. 2015.

[73] H. Yue, X. Sun, J. Yang, and F. Wu, “CID: Combined image denoising in spatial and frequency domains using web images,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR’14)*, pp. 2933–2940, Jun. 2014.

[74] S. H. Chan, E. Luo, and T. Q. Nguyen, “Adaptive patch-based image denoising by EM-adaptation,” in *Proc. IEEE Global Conf. Signal and Information Process. (GlobalSIP’15)*, Dec. 2015.

[75] J. Gauvain and C. Lee, “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Trans. Speech and Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[76] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang, “Image-specific prior adaptation for denoising,” *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5469–5478, Dec. 2015.

[77] D. Geman and C. Yang, “Nonlinear image recovery with half-quadratic regularization,” *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.

[78] D. Krishnan and R. Fergus, “Fast image deconvolution using hyper-Laplacian priors,” in *Advances in Neural Information Process. Systems 22*, pp. 1033–1041. Curran Associates, Inc., 2009.

[79] C. A. Bouman, “Model-based image processing,” [Available online] https : //engineering.purdue.edu/˜bouman/publications/pdf/MBIP − book.pdf, 2015.

[80] M. R. Gupta and Y. Chen, *Theory and Use of the EM Algorithm*, Now Publishers Inc., 2011.

[81] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[82] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted Gaussian mixture models,” *Digital Signal Process.*, vol. 10, no. 1, pp. 19–41, 2000.

[83] C. M. Stein, “Estimation of the mean of a multivariate normal distribution,” *The Annals of Statistics*, vol. 9, pp. 1135–1151, 1981.

[84] S. Ramani, T. Blu, and M. Unser, “Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms,” *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.

[85] P. C. Woodland, "Speaker adaptation for continuous density HMMs: A review," in *ITRW Adaptation Methods for Speech Recognition*, pp. 11–19, Aug. 2001.

[86] M. Dixit, N. Rasiwasia, and N. Vasconcelos, "Adapted Gaussian models for image classification," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR'11)*, pp. 937–943, Jun. 2011.

[87] B. McFee and G. R. Lanckriet, "Metric learning to rank," in *Proc. the 27th Intl. Conf. Machine Learning (ICML'10)*, pp. 775–782, 2010.

[88] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546, Jun. 2005.

[89] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, vol. 2, pp. 1735–1742, 2006.

[90] S. E. Reed, Y. Zhang, Y. Zhang, and H. Lee, "Deep visual analogy-making," in *Advances in Neural Information Process. Systems*, pp. 1252–1260, 2015.

[91] C. Li, *An efficient algorithm for total variation regularization with applications to the single pixel camera and compressive sensing*, Ph.D. thesis, Rice Univ., 2009, [Available online] http : //www.caam.rice.edu/˜optimization/L1/TVAL3/tval3_thesis.pdf.

[92] R. Horn and C. Johnson, *Matrix analysis*, Cambridge university press, 2012.