Adaptive Image Denoising by Mixture Adaptation

Enming Luo, *Student Member, IEEE*, Stanley H. Chan, *Member, IEEE*, and Truong Q. Nguyen, *Fellow, IEEE*

***Abstract*—We propose an adaptive learning procedure to learn effective image priors. The new algorithm, called the Expectation-Maximization (EM) adaptation, takes a generic prior learned from a generic external database and adapts it to the image of interest to generate a specific prior. Different from existing methods which combine internal and external statistics in an ad-hoc way, the proposed algorithm learns a single unified prior through an adaptive process. There are two major contributions in this paper. First, we rigorously derive the EM adaptation algorithm from the Bayesian hyper-prior perspective and show that it can be further simplified to improve the computational complexity. Second, in the absence of the latent clean image, we show how EM adaptation can be modified and applied on pre-filtered images. We discuss how to estimate internal parameters and demonstrate how to improve the denoising performance by running EM adaptation iteratively. Experimental results show that the adapted prior is consistently better than the originally un-adapted prior, and is superior than some state-of-the-art algorithms.**

***Index Terms*—Image Denoising, Hyper Prior, Conjugate Prior, Gaussian Mixture Models, Expectation-Maximization (EM), Expected Patch Log-Likelihood (EPLL), EM Adapation, BM3D**

I. Introduction

A. Overview

We consider the classical image denoising problem: Given an additive iid Gaussian noise model,

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\varepsilon}, \tag{1}$$

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, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}) \in \mathbb{R}^n$ 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.

Image denoising is a long-lasting problem. Numerous denoising algorithms have been proposed in the past few decades, ranging from spatial domain methods [1–3] to transform domain methods [4–6], and from local filtering [7–9] to global optimization [10, 11]. In this paper, we will focus on the maximum-a-posteriori (MAP) approach [11, 12]. MAP is a Bayesian approach which formulates the image denoising problem by maximizing the posterior probability

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

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 [13–15]. 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 [2, 4, 11]. 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

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

where the second term in (2) is called the expected patch log likelihood (EPLL) [11].

The focus of this paper 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 [16] and external priors [9, 17–20]. 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.* [17] 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.* [18] applied a neural network approach to learn the weights to combine internal and external denoising results; Yue *et al.* [21] 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.

B. 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 paper is an extension of our previous work reported in [22]. The three new contributions of this paper are:

E. Luo and T. Nguyen are with Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093, USA. Emails: eluo@ucsd.edu and nguyent@ece.ucsd.edu

S. Chan is with School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA. Email: stanleychan@purdue.edu

This work was supported by the National Science Foundation under grant CCF-1065305. Preliminary material in this paper was presented at the 3rd IEEE Global Conference on Signal & Information Processing (GlobalSIP), Orlando, December, 2015.

This paper follows the concept of reproducible research. All the results and examples presented in the paper are reproducible using the code and images available online at http://videoprocessing.ucsd.edu/~eluo

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 [23] 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.

When this manuscript is being written, we became aware of a very recent work of Lu *et al.* [24]. In comparison with [24], 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 paper is organized as follows. Section II gives a brief review of Gaussian mixture model. Section III presents the proposed EM adaptation algorithm. Section IV discusses how the EM adaptation algorithm should be modified when the image is noisy. Experimental results are presented in Section V.

II. 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.

A. 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

$$f(\boldsymbol{p}_i \,|\, \boldsymbol{\Theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{p}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3}$$

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

$$\mathcal{N}(\boldsymbol{p}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \overset{\text{def}}{=} \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\Big(-\frac{1}{2}(\boldsymbol{p}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{p}_i - \boldsymbol{\mu}_k)\Big) \tag{4}$$

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 (3), we can specify the denoising procedure by solving the optimization problem in (2). Here, we follow [25, 26] by using the *Half Quadratic Splitting* strategy. The idea is to replace (2) with the following equivalent minimization

$$\underset{\boldsymbol{x}, \{\boldsymbol{v}_i\}_{i=1}^n}{\arg\min} \Bigg\{ \frac{1}{2\sigma^2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \Big(-\log f(\boldsymbol{v}_i) + \frac{\beta}{2}\|\boldsymbol{P}_i\boldsymbol{x} - \boldsymbol{v}_i\|^2 \Big) \Bigg\}, \tag{5}$$

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 (5) is given in the following proposition.

Proposition 1: Assuming $f(\boldsymbol{v}_i)$ is dominated by the k_i^*-th components, where $k_i^* \overset{\text{def}}{=} \underset{k}{\text{argmax}}\ \pi_k \mathcal{N}(\boldsymbol{v}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, the solution of (5) is

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

Proof: See [11]. ■

Proposition 1 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 [27]. However, we shall not elaborate on these options. Our focus is on how to obtain the GMM.

B. 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 [28] 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 [11] 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.

III. 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.

A. 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 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.

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.

B. 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 $\widetilde{\boldsymbol{\Theta}} \stackrel{\text{def}}{=} \{(\widetilde{\pi}_k, \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^K$. 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.

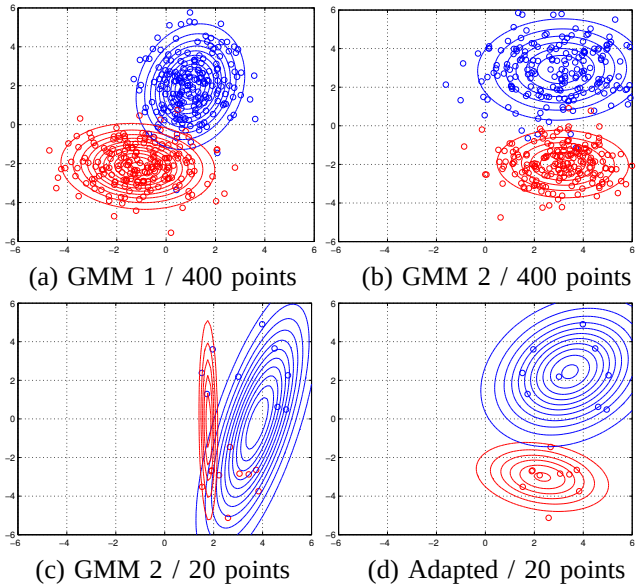
From a Bayesian inference perspective, estimation of the parameter $\widetilde{\boldsymbol{\Theta}}$ can be formulated as

$$\begin{aligned}\widetilde{\boldsymbol{\Theta}} &= \underset{\widetilde{\boldsymbol{\Theta}}}{\operatorname{argmax}}\ \log f(\widetilde{\boldsymbol{\Theta}} \,|\, \widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \\ &= \underset{\widetilde{\boldsymbol{\Theta}}}{\operatorname{argmax}} \Big(\log f(\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n \,|\, \widetilde{\boldsymbol{\Theta}}) + \log f(\widetilde{\boldsymbol{\Theta}}) \Big), \end{aligned} \tag{6}$$

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 (6) is also a MAP problem. However, the MAP for (6) is the estimation of the model parameter $\widetilde{\boldsymbol{\Theta}}$, which is different from the MAP for denoising used in (2). Although the difference seems subtle, there is a drastic different implication which we should be aware of.



(a) GMM 1 / 400 points (b) GMM 2 / 400 points

(c) GMM 2 / 20 points (d) Adapted / 20 points

Fig. 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). Note the significant improvement from (c) to (d) by using the proposed adaptation.

In (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 [29]. 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 [29].

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

For GMM, no joint conjugate prior can be found through the sufficient statistic approach [23]. 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

$$\widetilde{\pi}_1, \cdots, \widetilde{\pi}_K \ \sim\ \text{Dir}(v_1, \cdots, v_k), \tag{7}$$

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

$$(\widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k) \sim \text{NIW}(\boldsymbol{\vartheta}_k, \tau_k, \boldsymbol{\Psi}_k, \varphi_k), \quad \text{for } k = 1, \cdots, K, \quad (8)$$

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 1: 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 [23] 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 (7) and (8). By ignoring the scaling constants, it is not difficult to show that

$$f(\widetilde{\boldsymbol{\Theta}}) \propto \prod_{k=1}^{K} \Big\{ \widetilde{\pi}_k^{v_k - 1} |\widetilde{\boldsymbol{\Sigma}}_k|^{-(\varphi_k + d + 2)/2} \exp\Big(-\tfrac{\tau_k}{2} (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k)^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k) - \tfrac{1}{2} \text{tr}(\boldsymbol{\Psi}_k \widetilde{\boldsymbol{\Sigma}}_k^{-1}) \Big) \Big\}. \quad (9)$$

The importance of (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 2: Given the prior in (9), the posterior $f(\widetilde{\boldsymbol{\Theta}} | \widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n)$ is given by

$$f(\widetilde{\boldsymbol{\Theta}} \,|\, \widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \propto \prod_{k=1}^{K} \Big\{ \widetilde{\pi}_k^{v_k' - 1} |\widetilde{\boldsymbol{\Sigma}}_k|^{-(\varphi_k' + d + 2)/2} \exp\Big(-\tfrac{\tau_k'}{2} \big(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k'\big)^T \widetilde{\boldsymbol{\Sigma}}_k^{-1} \big(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k'\big) - \tfrac{1}{2} \text{tr}(\boldsymbol{\Psi}_k' \widetilde{\boldsymbol{\Sigma}}_k^{-1}) \Big) \Big\} \quad (10)$$

where

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

are the parameters for the posterior density.

Proof: See Appendix A. ∎

D. Solve for $\widetilde{\boldsymbol{\Theta}}$

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

$$\begin{aligned} \underset{\widetilde{\boldsymbol{\Theta}}}{\text{maximize}} \quad & L(\widetilde{\boldsymbol{\Theta}}) \stackrel{\text{def}}{=} \log f(\widetilde{\boldsymbol{\Theta}} | \widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \\ \text{subject to} \quad & \textstyle\sum_{k=1}^{K} \widetilde{\pi}_k = 1. \end{aligned} \quad (11)$$

The constrained problem (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 3: The optimal $(\widetilde{\pi}_k, \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)$ for (11) are

$$\widetilde{\pi}_k = \frac{n}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{n_k}{n} + \frac{\sum_{k=1}^{K} v_k - K}{(\sum_{k=1}^{K} v_k - K) + n} \cdot \frac{v_k - 1}{\sum_{k=1}^{K} v_k - K}, \quad (12)$$

$$\widetilde{\boldsymbol{\mu}}_k = \frac{1}{\tau_k + n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i + \frac{\tau_k}{\tau_k + n_k} \boldsymbol{\vartheta}_k, \quad (13)$$

$$\widetilde{\boldsymbol{\Sigma}}_k = \frac{n_k}{\varphi_k + d + 2 + n_k} \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T + \frac{1}{\varphi_k + d + 2 + n_k} \left(\boldsymbol{\Psi}_k + \tau_k (\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \quad (14)$$

Proof: See Appendix B. ∎

Remark 2: The results we showed in Proposition 3 are different from [23]. In particular, the denominator for $\widetilde{\boldsymbol{\Sigma}}_k$ in [23] 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.

E. Simplification of $\widetilde{\boldsymbol{\Theta}}$

The results in Proposition 3 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 3, we summarize the simplification results as follows.

Proposition 4: Define $\rho \stackrel{\text{def}}{=} \frac{n_k}{n}(\sum_{k=1}^{K} v_k - K) = \tau_k = \varphi_k + d + 2$. 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 (12)-(14) become

$$\widetilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k)\pi_k, \quad (15)$$

$$\widetilde{\boldsymbol{\mu}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i + (1 - \alpha_k)\boldsymbol{\mu}_k, \quad (16)$$

$$\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T + (1 - \alpha_k) \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \quad (17)$$

Remark 3: We note that Reynold *et al.* [30] 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 (17) with the work of Lu *et al.* [24], we note that in [24] the covariance is

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

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.

F. EM Adaptation Algorithm

The proposed EM adaptation algorithm is summarized in Algorithm 1. 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

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}. \tag{19}$$

M-Step: The more interesting step of the adaptation is the M-step. From (22) to (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

$$\widetilde{\boldsymbol{\mu}}_k = \underbrace{\alpha_k \left(\frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i \right)}_{\textbf{new data}} + \underbrace{(1-\alpha_k)\boldsymbol{\mu}_k}_{\textbf{generic prior}}. \tag{20}$$

The updated mean in (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 (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.

Algorithm 1 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 $i = 1, \ldots, n$

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_i | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}, \quad n_k = \sum_{i=1}^{n} \gamma_{ki}. \tag{21}$$

M-step : Compute, for $k = 1, \ldots, K$

$$\widetilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1-\alpha_k)\pi_k, \tag{22}$$

$$\widetilde{\boldsymbol{\mu}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i + (1-\alpha_k)\boldsymbol{\mu}_k, \tag{23}$$

$$\begin{aligned} \widetilde{\boldsymbol{\Sigma}}_k = \; & \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T \\ & + (1-\alpha_k) \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \end{aligned} \tag{24}$$

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.

G. Convergence

The EM adaptation shown in Algorithm 1 is an EM algorithm. Therefore, its convergence is guaranteed by the classical theory, which we state without proof as follows.

Proposition 5: 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 [28]. ∎

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 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.

IV. EM Adaptation for Denoising

The proposed Algorithm 1 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.

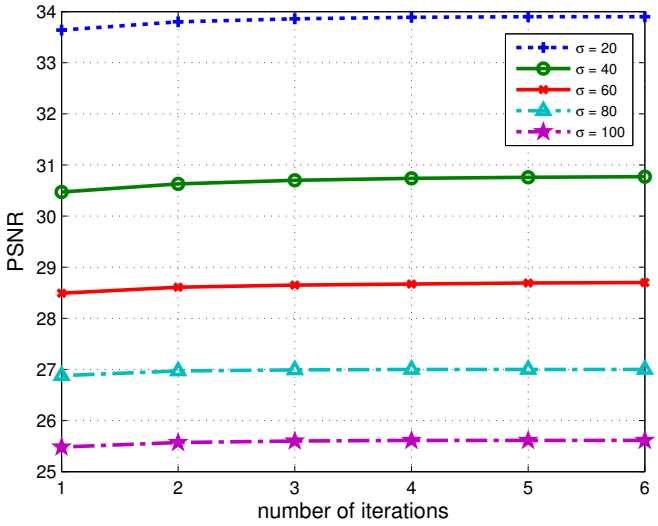


Fig. 2: Image denoising using EM adaptation: The PSNR only improves marginally after the first iteration, confirming that the EM adaptation can typically be performed in a single iteration. The testing image is House of size 256 × 256. $\sigma = 20, \ldots, 100$ indicates the noise level.

A. Adaptation to a Pre-filtered Image

To deal with the presence of noise, we adopt a two-stage approach similar to BM3D [4]. 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 $(\overline{\boldsymbol{x}} - \boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{0}, \widetilde{\sigma}^2 \boldsymbol{I})$, where $\widetilde{\sigma}^2 \stackrel{\text{def}}{=} \mathbb{E}\|\overline{\boldsymbol{x}} - \boldsymbol{x}\|^2$ is the variance of $\overline{\boldsymbol{x}}$. By incorporating the residual noise, we modify (21) as

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k + \widetilde{\sigma}^2 \boldsymbol{I})}{\sum_{l=1}^{K} \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_l \,|\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \widetilde{\sigma}^2 \boldsymbol{I})}, \tag{25}$$

and (24) as

$$\begin{aligned} \widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \left((\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T - \widetilde{\sigma}^2 \boldsymbol{I} \right) \\ + (1 - \alpha_k) \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \end{aligned} \tag{26}$$

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

B. 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) [31]. SURE provides a way for unbiased estimation of the true MSE. The analytical expression of SURE is

$$\widetilde{\sigma}^2 \approx \text{SURE} \stackrel{\text{def}}{=} \frac{1}{n}\|\boldsymbol{y} - \overline{\boldsymbol{x}}\|^2 - \sigma^2 + \frac{2\sigma^2}{n}\text{div}, \tag{27}$$

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 [32] to approximate the divergence. We shall not repeat Monte-Carlo SURE here but we summarize the steps in Algorithm 2.

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

Input: noisy image $\boldsymbol{y} \in \mathbb{R}^n$, noise variance σ^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 $\text{div} = \frac{1}{\delta}\boldsymbol{b}^T(\overline{\boldsymbol{x}}' - \overline{\boldsymbol{x}})$.
Compute $\widetilde{\sigma}^2 = \text{SURE} \stackrel{\text{def}}{=} \frac{1}{n}\|\boldsymbol{y} - \overline{\boldsymbol{x}}\|^2 - \sigma^2 + \frac{2\sigma^2}{n}\text{div}$.

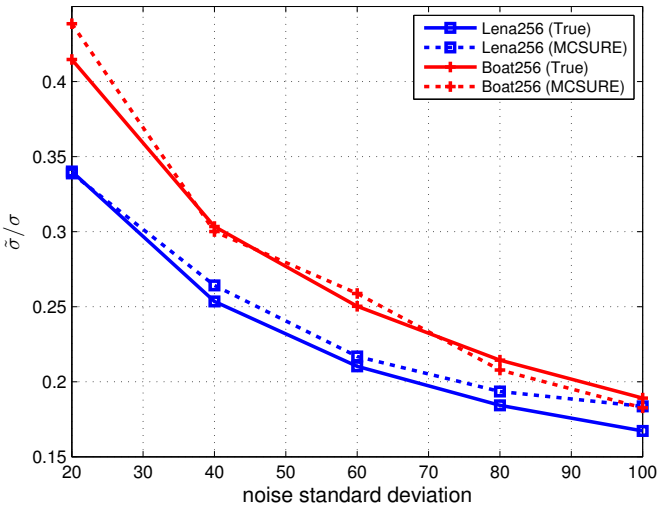


Fig. 3: Comparison between the true MSE and Monte-Carlo SURE when estimating $\widetilde{\sigma}/\sigma$ over a large range of noise levels. The pre-filtering method is EPLL.

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 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 3 is EPLL. For other methods such as BM3D, we have similar observations for different noise levels.

C. 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 and the relevance factor ρ. The factor ρ is adjusted to allow different adaptation rates. For example, in the application of speaker verification [30, 33], 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, 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.

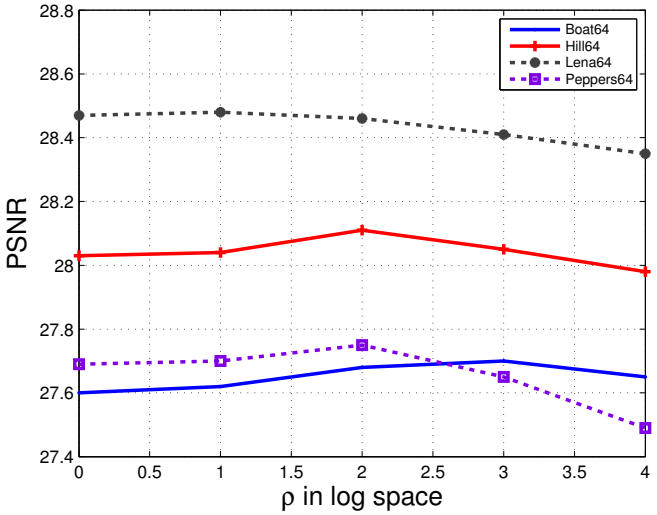


Fig. 4: The effect of ρ on denoising performance. The pre-filtered image is used for EM adaptation algorithm. The testing images are of size 64×64 with noise $\sigma = 20$.

D. 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 1, 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 [30, 33, 34]. 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 (24) but with significantly less operations. The idea is to exploit the algebraic structure of the covariance matrix.

Proposition 6: The full covariance adaptation in (24) can be simplified as

$$\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^{n} \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T - \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T + (1-\alpha_k)(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T). \quad (28)$$

Proof: See Appendix C. ■

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 1, we list the averaging runtime when computing (24) and (28) for two image sizes.

	image size	Eq. (24) (original)	Eqn. (28) (ours)	Speedup factor
runtime (sec)	(64×64)	31.34	0.30	104.5
runtime (sec)	(128×128)	136.58	1.32	103.2

Table 1: Runtime comparison between (24) and (28) for different image sizes.

V. 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.

A. Experiment Settings

For comparison, we consider two state-of-the-art methods: BM3D [4] and EPLL [11]. 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.

B. Single Image Denoising

We use 6 standard images of size 256×256, and 6 natural images of size 481×321 randomly chosen from [35] for the single image denoising experiments.

Figure 5 shows the denoising results for three standard testing images and 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.

In Table 2, we report the detailed PSNR results for different noise variances for the standard images. Two 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.

Due to limited space, the detailed PSNR results for the natural images are not shown in this paper. The additional results can be found at http://videoprocessing.ucsd.edu/~eluo.

		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 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.

C. 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 [9, 20], 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 [36] 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 [37] 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 6, 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.

D. 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 (28) in Section IV-D has significantly improved the computational efficiency for EM adaptation.

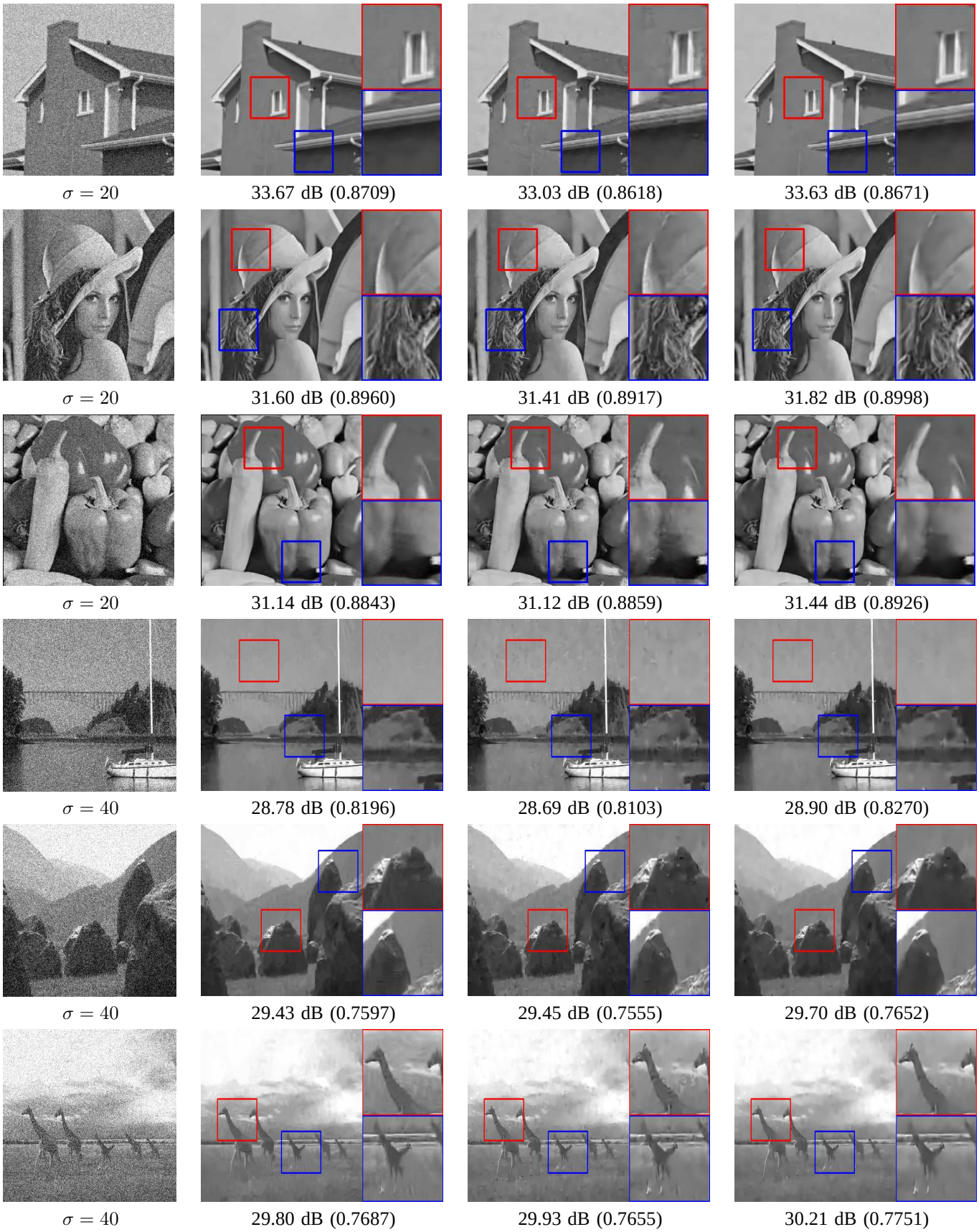


Fig. 5: Single image denoising by using the denoised image for EM adaptation: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The top three are standard images of size 256×256 while the the bottom three are natural images of size 481×321.

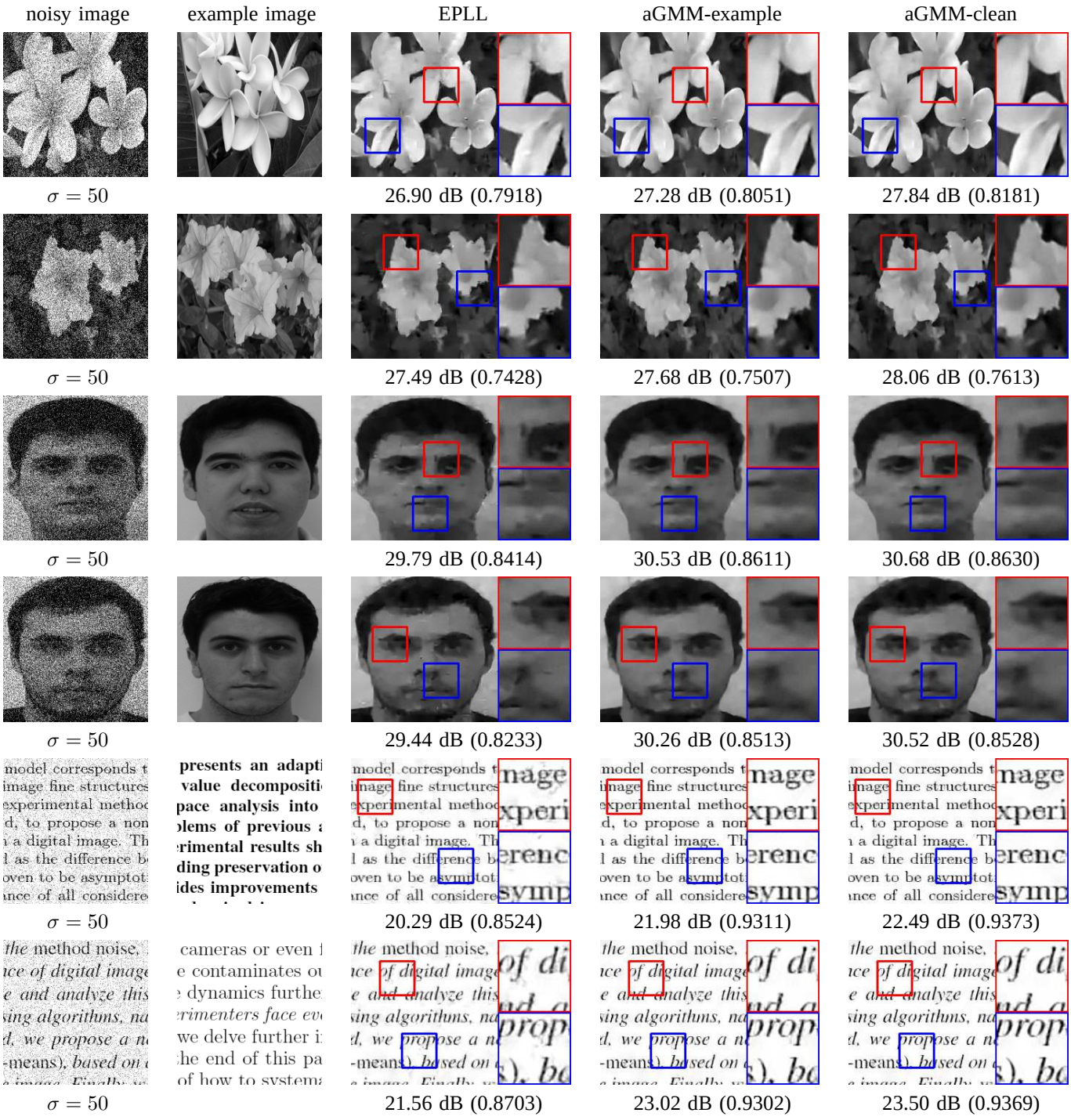


Fig. 6: External image denoising by using an example image for EM adaptation: Visual comparison and objective comparison (PSNR and SSIM in the parenthesis). The flower images are from the 102flowers dataset [36], fhe face images are from the FEI face dataset [37], and the text images are cropped from randomly chosen documents.

VI. 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.

Appendix

A. Proof of Proposition 2

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

$$\gamma_{ki} = \frac{\pi_k^{(m)}\mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_k^{(m)}, \boldsymbol{\Sigma}_k^{(m)})}{\sum_{l=1}^K \pi_l^{(m)}\mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_l^{(m)}, \boldsymbol{\Sigma}_l^{(m)})}, \tag{29}$$

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 (6) by the Q function as follows

$$\begin{aligned} Q(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{\Theta}}^{(m)}) &= \sum_{i=1}^n \sum_{k=1}^K \gamma_{ki} \log\Big(\widetilde{\pi}_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i|\widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)\Big) \\ &\doteq \sum_{i=1}^n \sum_{k=1}^K \gamma_{ki}\Big(\log\widetilde{\pi}_k - \frac{1}{2}\log|\widetilde{\boldsymbol{\Sigma}}_k| \\ &\quad - \frac{1}{2}(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)\Big) \\ &= \sum_{k=1}^K n_k(\log\widetilde{\pi}_k - \frac{1}{2}\log|\widetilde{\boldsymbol{\Sigma}}_k|) \\ &\quad - \frac{1}{2}\sum_{k=1}^K\sum_{i=1}^n \gamma_{ki}(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k), \end{aligned} \tag{30}$$

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

$$\bar{\boldsymbol{\mu}}_k \overset{\text{def}}{=} \frac{1}{n_k}\sum_{i=1}^n \gamma_{ki}\widetilde{\boldsymbol{p}}_i, \quad \boldsymbol{S}_k \overset{\text{def}}{=} \sum_{i=1}^n \gamma_{ki}(\widetilde{\boldsymbol{p}}_i - \bar{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \bar{\boldsymbol{\mu}}_k)^T. \tag{31}$$

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

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

Therefore, we have

$$\begin{aligned} & f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \propto \exp\big(Q(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{\Theta}}^{(m)}) + \log f(\widetilde{\boldsymbol{\Theta}})\big) \\ &= f(\widetilde{\boldsymbol{\Theta}}) \prod_{k=1}^K \Big\{ \widetilde{\pi}_k^{n_k} \, |\widetilde{\boldsymbol{\Sigma}}_k|^{-n_k/2} \\ & \exp\big(-\frac{n_k}{2}(\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k)^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k) - \frac{1}{2}\text{tr}(\boldsymbol{S}_k\widetilde{\boldsymbol{\Sigma}}_k^{-1})\big)\Big\} \\ &= \prod_{k=1}^K \Big\{ \widetilde{\pi}_k^{v_k + n_k - 1} |\widetilde{\boldsymbol{\Sigma}}_k|^{-(\varphi_k + n_k + d + 2)/2} \exp\Big(-\frac{\tau_k + n_k}{2} \\ & (\widetilde{\boldsymbol{\mu}}_k - \frac{\tau_k\boldsymbol{\vartheta}_k + n_k\bar{\boldsymbol{\mu}}_k}{\tau_k + n_k})^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{\mu}}_k - \frac{\tau_k\boldsymbol{\vartheta}_k + n_k\bar{\boldsymbol{\mu}}_k}{\tau_k + n_k})\Big) \\ & \exp\Big(-\frac{1}{2}\text{tr}((\boldsymbol{\Psi}_k + \boldsymbol{S}_k \\ & + \frac{\tau_k n_k}{\tau_k + n_k}(\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \bar{\boldsymbol{\mu}}_k)^T)\widetilde{\boldsymbol{\Sigma}}_k^{-1})\Big)\Big\}. \end{aligned} \tag{32}$$

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

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

which completes the proof. ∎

B. Proof of Proposition 3

Proof: We ignore some irrelevant terms and get $\log f(\widetilde{\boldsymbol{\Theta}}|\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n) \doteq \sum_{k=1}^K\{(v_k' - 1)\log\widetilde{\pi}_k - \frac{(\varphi_k' + d + 2)}{2}\log|\widetilde{\boldsymbol{\Sigma}}_k| - \frac{\tau_k'}{2}(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k')^T\widetilde{\boldsymbol{\Sigma}}_k^{-1}(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k') - \frac{1}{2}\text{tr}(\boldsymbol{\Psi}_k'\widetilde{\boldsymbol{\Sigma}}_k^{-1})\}$. 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

$$\begin{aligned}\widetilde{\pi}_k =& \frac{v_k' - 1}{\sum_{k=1}^K (v_k' - 1)} \\ =& \frac{(v_k - 1) + n_k}{(\sum_{k=1}^K v_k - K) + n} \\ =& \frac{n}{(\sum_{k=1}^K v_k - K) + n} \cdot \frac{n_k}{n} \\ &+ \frac{\sum_{k=1}^K v_k - K}{(\sum_{k=1}^K v_k - K) + n} \cdot \frac{v_k - 1}{\sum_{k=1}^K v_k - K}. \end{aligned} \quad (33)$$

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

$$\frac{\partial L}{\partial \widetilde{\boldsymbol{\mu}}_k} = -\frac{\tau_k'}{2} \widetilde{\boldsymbol{\Sigma}}_k^{-1} (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k') = 0, \quad (34)$$

the solution of which is

$$\begin{aligned}\widetilde{\boldsymbol{\mu}}_k =& \frac{\tau_k \boldsymbol{\vartheta}_k + n_k \bar{\boldsymbol{\mu}}_k}{\tau_k + n_k} \\ =& \frac{1}{\tau_k + n_k} \sum_{i=1}^n \gamma_{ki} \widetilde{\boldsymbol{p}}_i + \frac{\tau_k}{\tau_k + n_k} \boldsymbol{\vartheta}_k. \end{aligned} \quad (35)$$

- 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

$$(\varphi_k' + d + 2) \widetilde{\boldsymbol{\Sigma}}_k = \boldsymbol{\Psi}_k' + \tau_k' (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k')(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k')^T, \quad (36)$$

the solution of which is

$$\begin{aligned}\widetilde{\boldsymbol{\Sigma}}_k =& \frac{\boldsymbol{\Psi}_k' + \tau_k' (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k')(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k')^T}{\varphi_k' + d + 2} \\ =& \frac{\boldsymbol{\Psi}_k + \tau_k (\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k)(\widetilde{\boldsymbol{\mu}}_k - \boldsymbol{\vartheta}_k)^T}{\varphi_k + d + 2 + n_k} \\ &+ \frac{n_k (\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{\mu}}_k - \bar{\boldsymbol{\mu}}_k)^T + \boldsymbol{S}_k}{\varphi_k + d + 2 + n_k} \\ =& \frac{n_k}{\varphi_k + d + 2 + n_k} \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T \\ &+ \frac{1}{\varphi_k + d + 2 + n_k} \left(\boldsymbol{\Psi}_k + \tau_k (\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\vartheta}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \end{aligned} \quad (37)$$

■

C. Proof of Proposition 6

Proof: Our first attempt is to expand the first term in (24).

$$\begin{aligned}&\alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T \\ =& \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T - \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{\mu}}_k^T - \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{p}}_i^T + \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T) \\ \triangleq& \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T - (\widetilde{\boldsymbol{\mu}}_k - (1 - \alpha_k) \boldsymbol{\mu}_k) \widetilde{\boldsymbol{\mu}}_k^T \\ &- \widetilde{\boldsymbol{\mu}}_k (\widetilde{\boldsymbol{\mu}}_k - (1 - \alpha_k) \boldsymbol{\mu}_k)^T + \alpha_k \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T \\ =& \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T - 2 \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T \\ &+ (1 - \alpha_k)(\boldsymbol{\mu}_k \widetilde{\boldsymbol{\mu}}_k^T + \widetilde{\boldsymbol{\mu}}_k \boldsymbol{\mu}_k^T) + \alpha_k \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T, \end{aligned} \quad (38)$$

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 (23).

We then expand the second term in (24)

$$\begin{aligned}&(1 - \alpha_k) \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right) \\ =& (1 - \alpha_k)(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T + \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T) \\ &- (1 - \alpha_k)(\boldsymbol{\mu}_k \widetilde{\boldsymbol{\mu}}_k^T + \widetilde{\boldsymbol{\mu}}_k \boldsymbol{\mu}_k^T). \end{aligned} \quad (39)$$

Combining (38) and (39), we get a simplified (24) as

$$\begin{aligned}\widetilde{\boldsymbol{\Sigma}}_k =& \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \widetilde{\boldsymbol{p}}_i \widetilde{\boldsymbol{p}}_i^T - \widetilde{\boldsymbol{\mu}}_k \widetilde{\boldsymbol{\mu}}_k^T \\ &+ (1 - \alpha_k)(\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T). \end{aligned} \quad (40)$$

■

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.

[17] 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.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] C. A. Bouman, "Model-based image processing," [Available online] https://engineering.purdue.edu/~bouman/publications/pdf/MBIP-book.pdf, 2015.

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

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

[30] 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.

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

[32] 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.

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

[34] 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.

[35] 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.

[36] 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.

[37] 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.