ADAPTIVE PATCH-BASED IMAGE DENOISING BY EM-ADAPTATION

Stanley H. Chan [1]*, Enming Luo*[2]*, and Truong Q. Nguyen*[2]

[1]School of ECE and Dept of Statistics, Purdue University, West Lafayette, IN 47907.
[2]University of California, San Diego, Dept of ECE, 9500 Gilman Drive, La Jolla, CA 92093.

ABSTRACT

Effective image prior is a key factor for successful image denoising. Existing learning-based priors require a large collection of images for training. Besides being computationally expensive, these training images do not necessarily correspond to the noisy image of interest. In this paper, we propose an adaptive learning procedure for learning image patch priors. The new algorithm, called the Expectation-Maximization (EM) adaptation, maps a generic prior to a targeted image to create a specific prior. EM adaptation requires significantly less amount of training data compared to the standard EM, and can be applied to pre-filtered images in the absence of clean databases. Experimental results show that the adapted prior is consistently better than the originally un-adapted prior, and has superior performance than some state-of-the-art algorithms.

Index Terms— Image denoising, Gaussian Mixture models, Expectation-Maximization, Expected Patch Log-Likelihood, EM-adaptation, BM3D

1. INTRODUCTION

1.1. Overview

We consider the classical image denoising problem of recovering an image from a noisy observation, where the noise is i.i.d. Gaussian. Mathematically, the problem is to solve an inverse problem by finding $\boldsymbol{x} \in \mathbb{R}^n$ from

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

where $\boldsymbol{y} \in \mathbb{R}^n$ is the observed image, and $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$ is a Gaussian noise vector of variance σ^2.

Despite the long history of the problem, image denoising remains an active research area in which new methods are rapidly developed. In this paper, we shall focus on a class of *patch-based* algorithms which exploit the statistics of overlapping patches of the image. These patches can be used to: form weighted averages [1–3], match and threshold using 3D transforms [4–6], denoise using group sparsity [7–9], or perform high-order singular value decomposition for denoising [10].

Patch-based algorithms rely heavily on the priors they use. General speaking, there are two types of priors. Priors learned from the single noisy image are called *internal* priors [11], and priors learned from a database of external images are called *external* priors [12–14]. 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.

Contact author: S. Chan, E-mail: stanleychan@purdue.edu. The work of E. Luo and T. Q. Nguyen is supported in part by a NSF grant CCF-1065305 and by DARPA under contract W911NF-11-C-0210.

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. This tension between the quantity of available training images and the quality of each image sets the ground for this paper.

1.2. Related Works and Contributions

In the recent literature of patch-based algorithm, there are a number of *fusion* methods aiming to combine internal and external statistics. In [12], Mosseri *et al.* proposed a patch signal-to-noise ratio as a quantitative metric to decide whether a patch should be denoised internally or externally. A similar concept was proposed by Burger *et al.* [13], where they applied a neural network approach to learn optimal weights to combine internal and external denoising results. There are also methods trying to fuse the internal and external denoising results in the frequency domain [15]. A recent paper by Sulam and Elad [16] attempted the same problem by combining the expected patch log-likelihood (EPLL) [17] with a modified K-SVD [18].

The contribution of this paper is an *adaptive* approach to adapt external priors to internal priors. Adaptive patch-based algorithms have been proposed in the past but in a different context. For example, Luo *et al.* [19] proposed a method to adaptively update the search range of non-local means. This paper is built on top of a targeted denoising algorithm proposed in [20, 21]. In [20, 21], it was shown that if the external database has a good match with the noisy image, the resulting denoising performance can be significantly better than some existing algorithms.

The method we propose in this paper is called the *EM-adaptation*. Different from the fusion methods which combine internal and external methods via a weighted average, we learn a single unified patch prior through an adaptive process which takes an external prior (called the generic prior) and maps it to a targeted noisy image to generate a specific prior. Conceptually our work could be classified as a type of knowledge transfer or transfer learning [22].

The advantages of the proposed method are two-fold. First, by adapting the generic prior to the specific image, we avoid the ad-hoc procedure of finding the fusion weights. Second, our method can be applied even if there are very few training samples. This makes the algorithm more favorable than other learning methods such as EPLL [17] and PLE [23] which fail to learn from a single image.

The rest of the paper is organized as follows. After introducing some mathematical preliminaries in Section 2, we present our proposed method in Section 3. Experimental results are shown in Section 4, and concluding remarks are given in Section 5.

2. MATHEMATICAL PRELIMINARIES

2.1. Patch Prior with Gaussian Mixtures

We formulate the denoising problem as a maximum-a-posteriori (MAP) estimation problem and we aim to solve the optimization

$$\begin{aligned}\operatorname*{argmax}_{\boldsymbol{x}} \; f(\boldsymbol{y}|\boldsymbol{x})f(\boldsymbol{x}) &= \operatorname*{argmin}_{\boldsymbol{x}} \; \{-\log f(\boldsymbol{y}|\boldsymbol{x}) - \log f(\boldsymbol{x})\} \\ &= \operatorname*{argmin}_{\boldsymbol{x}} \left\{ \frac{\sigma^{-2}}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 - \log f(\boldsymbol{x}) \right\}, \end{aligned} \tag{2}$$

where the first term in (2) is due to the Gaussian distribution of $f(\boldsymbol{y}|\boldsymbol{x})$, and $f(\boldsymbol{x})$ is the prior distribution of the latent clean image.

Finding the prior for the latent clean image is difficult because of its high dimensionality and computational intractability. Patch-based priors alleviate this problem by considering the distribution of the patches instead of the whole image. 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 $\boldsymbol{x}$, one can express the negative log of the image prior as a sum of the log patch priors, leading to the expected patch log-likelihood (EPLL) framework [17]:

$$\operatorname*{argmin}_{\boldsymbol{x}} \left\{ \frac{\sigma^{-2}}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 - \frac{1}{n}\sum_{i=1}^{n} \log f(\boldsymbol{P}_i\boldsymbol{x}) \right\}. \tag{3}$$

To define the distribution $f(\boldsymbol{P}_i\boldsymbol{x})$, in this paper we consider the Gaussian mixture model (GMM), although other mixtures of the exponential family are equally applicable. Denoting $\boldsymbol{p}_i \overset{\text{def}}{=} \boldsymbol{P}_i\boldsymbol{x}$ as the i-th patch, the distribution $f(\boldsymbol{p}_i)$ is

$$f(\boldsymbol{p}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{p}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4}$$

where π_k, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the weight, mean and covariance of the kth Gaussian component, respectively. For notational simplicity we collectively denote them as $\boldsymbol{\Theta} \overset{\text{def}}{=} \{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^{K}$. Learning the parameter $\boldsymbol{\Theta}$ can be done using the Expectation-Maximization (EM) algorithm, which we shall not discuss here. Interested readers can refer to [24] for a comprehensive tutorial.

2.2. Denoising with Gaussian Mixtures

With the GMM defined in (4), one can solve the optimization in (3) by using the *half quadratic splitting* strategy [25, 26]. The idea is to consider an equivalent problem

$$\operatorname*{argmin}_{\boldsymbol{x},\{\boldsymbol{v}_i\}} \left\{ \frac{n\sigma^{-2}}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2 + \sum_{i=1}^{n}\left(-\log f(\boldsymbol{v}_i) + \frac{\beta}{2}\|\boldsymbol{P}_i\boldsymbol{x}-\boldsymbol{v}_i\|^2 \right) \right\}, \tag{5}$$

and solve for $\boldsymbol{x}$ and $\{\boldsymbol{v}_i\}_{i=1}^{n}$ alternatingly. If we further assume that $f(\boldsymbol{v}_i)$ is dominated by one of the components k_i^* where

$$k_i^* \overset{\text{def}}{=} \operatorname*{argmax}_{k} \;\; \pi_k \mathcal{N}(\boldsymbol{v}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

then the alternating minimization in (5) can be simplified as

$$\begin{aligned} \boldsymbol{x} &= \left(n\sigma^{-2}\boldsymbol{I} + \beta\sum_{i=1}^{n}\boldsymbol{P}_i^T\boldsymbol{P}_i \right)^{-1} \left(n\sigma^{-2}\boldsymbol{y} + \beta\sum_{i=1}^{n}\boldsymbol{P}_i^T\boldsymbol{v}_i \right), \\ \boldsymbol{v}_i &= \left(\beta\boldsymbol{\Sigma}_{k_i^*} + \boldsymbol{I}\right)^{-1}\left(\boldsymbol{\mu}_{k_i^*} + \beta\boldsymbol{\Sigma}_{k_i^*}\boldsymbol{P}_i\boldsymbol{x}\right). \end{aligned} \tag{6}$$

3. EM ADAPTATION

3.1. Limitations of the EM Algorithm

While the above denoising framework produces reasonable results, learning the Gaussian mixture model has a number of issues.

1. **Adaptivity**: The GMM learned by the standard EM algorithm is a *global* structure of *all* the images in the database. The GMM is thus a *generic* prior as it is not specifically learned for a particular image. Previous works have shown that performance would improve if one uses a targeted database [20, 21]. However, how to adaptively learn a prior from an external database to a specific image remains unclear.
2. **Computational cost**: Learning a good GMM requires a large number of training samples. This is a computationally expensive task, e.g., [17] spent 30 hours on constructing the GMM.
3. **Finite samples**: When training samples are few, the learned GMM will be over-fitted, and some components will even become singular.
4. **Noise**: Most learning-based methods are designed to learn statistics from *clean* images. However, if we are only given a *noisy* image, it becomes unclear how one can learn the prior effectively.

3.2. The EM Adaptation Algorithm

Our proposed EM adaptation algorithm is a modification of the standard EM algorithm. Suppose that from an external database we have already learned a GMM with parameter $\boldsymbol{\Theta} \overset{\text{def}}{=} \{(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\}_{k=1}^{K}$. Our goal is to learn a specific GMM with parameters $\widetilde{\boldsymbol{\Theta}} \overset{\text{def}}{=} \{(\widetilde{\pi}_k, \widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)\}_{k=1}^{K}$ so that $\widetilde{\boldsymbol{\Theta}}$ is a good fit to the specific image as well as the external database.

To take a closer look at the EM adaptation algorithm, we denote $\{\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n\}$ as the collection of patches taken from the specific image. The E-step and the M-step are modified as follows.

E-Step: In the E-step of the EM adaptation, we compute the likelihood of $\widetilde{\boldsymbol{p}}_i$ conditioned on the generic parameter $\boldsymbol{\Theta}$. This returns a normalized probability

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

and we can define an intermediate parameter $n_k = \sum_{i=1}^{n}\gamma_{ki}$.

M-Step: The more interesting step of the adaptation is the M-step. Normally, without the EM adaptation, the M-step updates $(\pi_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ using only the available data, which is $\{\widetilde{\boldsymbol{p}}_1, \ldots, \widetilde{\boldsymbol{p}}_n\}$ in our case. However, with the generic prior we modify the M-step by re-weighting the contributions of the generic parameter and the new data. Taking the mean as an example, the EM adaptation updates the mean according to

$$\widetilde{\boldsymbol{\mu}}_k = \underbrace{\alpha_k \left(\frac{1}{n_k}\sum_{i=1}^{n}\gamma_{ki}\widetilde{\boldsymbol{p}}_i \right)}_{\text{new data}} + \underbrace{(1-\alpha_k)\boldsymbol{\mu}_k}_{\text{generic prior}}, \tag{8}$$

where $\alpha_k \overset{\text{def}}{=} \frac{n_k}{n_k+\rho}$ is the combination weight, with a predefined relevance factor ρ. (8) suggests that the adaptation is in fact a *soft* combination of the generic prior and the new data (as opposed to the

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 $i = 1, \ldots, n$

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^K \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_i | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}, \quad n_k = \sum_{i=1}^n \gamma_{ki}. \tag{9}$$

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

$$\widetilde{\pi}_k = \alpha_k \frac{n_k}{n} + (1 - \alpha_k)\pi_k, \tag{10}$$

$$\widetilde{\boldsymbol{\mu}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \widetilde{\boldsymbol{p}}_i + (1 - \alpha_k)\boldsymbol{\mu}_k, \tag{11}$$

$$\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} (\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T + (1 - \alpha_k) \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right). \tag{12}$$

Postprocessing: Normalize $\{\widetilde{\pi}_k\}_{k=1}^K$ so that they sum to 1, and transform $\{\widetilde{\boldsymbol{\Sigma}}_k\}_{k=1}^K$ to be positive semi-definite if it is not.

hard combination in the fusion methods). If $\alpha_k \to 1$, then more emphasis will be put on the new data. Conversely, if $\alpha_k \to 0$, then we will keep the generic parameter. The swing between the two options is automatically controlled by the likelihood sum $n_k = \sum_{i=1}^n \gamma_{ki}$.

Other steps (i.e., learning $\widetilde{\pi}_i$ and $\widetilde{\boldsymbol{\Sigma}}_i$) of the EM adaptation are summarized in Algorithm 1. In our experiments, we observe that the EM adaptation converges in very few number of iterations. In fact, the improvement becomes marginal after one single iteration.

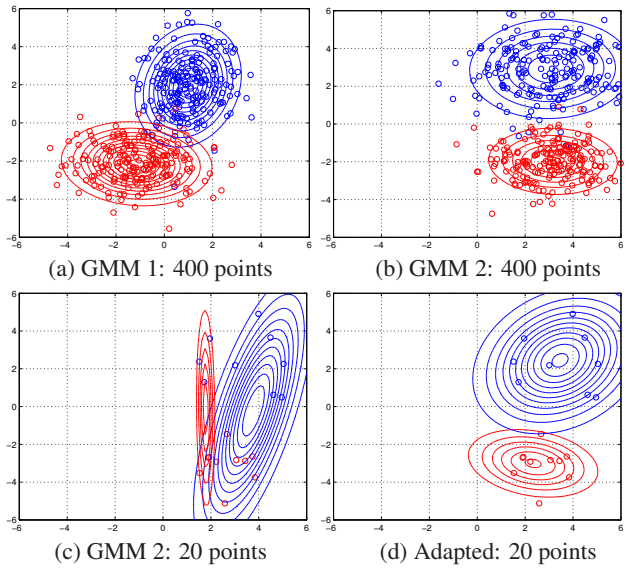
Complete Derivation: We remark that the EM adaptation algorithm presented above can be theoretically derived from a Bayesian perspective for a hidden Markov model. The weights $\widetilde{\pi}_k$ are the parameters of a Dirichlet distribution, and $(\widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\Sigma}}_k)$ are the parameters of the normal-Wishart distribution. The convergence of the EM adaptation follows from [27]. EM-adaptation has also been applied to speaker identification [28, 29] and image classification [30].

Toy Example: We illustrate the idea of the EM adaptation using a toy example. Figure 1 (a) and (b) show two individual GMMs learned from two sets of data, each containing 400 data points. It can be seen that the GMMs are estimated reasonably well. In (c), we draw a subset of 20 data points from (b) and try to learn a GMM from these 20 data points. We observe that the learned GMM is over-fitted to these 20 data points, and is very different from the GMM in (b). Using the EM adaptation, we observe in (d) that the adapted GMM is significantly better than (c), despite the fact that it only uses 20 data points. In this example we let $\rho = 1$ when computing α_k.

3.3. Adaptation to a Pre-filtered Image

At this point, careful readers would probably argue that Algorithm 1 will only work for *clean* images. We now discuss how the algorithm can be modified to learn a GMM using *noisy* images.

In the presence of noise, we adopt a pre-filtering approach similar to the two-stage denoising methods such as BM3D and EPLL. In the first stage, we apply an existing denoising algorithm to obtain a pre-filtered image. The adaptation is then applied in the second stage to the pre-filtered image. However, since a pre-filtered image is not the same as the ground truth clean image, we must quantify the residual noise remaining in the pre-filtered image.



(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 with two components, each learned from 400 data points. (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 this example we choose GMM 1 as the initialization for EM adaptation and let $\rho = 1$.

To this end, we let $\bar{\boldsymbol{x}}$ be the pre-filtered image. The distribution of the residue $\bar{\boldsymbol{x}} - \boldsymbol{x}$ is typically unknown but empirically we observe that it can be reasonably approximated by a single Gaussian. Thus, we model $(\bar{\boldsymbol{x}} - \boldsymbol{x}) \sim \mathcal{N}(\boldsymbol{0}, \widetilde{\sigma}^2 \boldsymbol{I})$, where $\widetilde{\sigma}^2$ is the estimated mean squared error. $\widetilde{\sigma}^2$ can be estimated using the Stein's Unbiased Risk Estimator [31, 32]. Due to limited space we shall leave the details to a follow up journal paper. Combining these we modify (9) as

$$\gamma_{ki} = \frac{\pi_k \mathcal{N}(\widetilde{\boldsymbol{p}}_i \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k + \widetilde{\sigma}^2 \boldsymbol{I})}{\sum_{l=1}^K \pi_l \mathcal{N}(\widetilde{\boldsymbol{p}}_l \,|\, \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l + \widetilde{\sigma}^2 \boldsymbol{I})}, \tag{13}$$

and (12) as

$$\widetilde{\boldsymbol{\Sigma}}_k = \alpha_k \frac{1}{n_k} \sum_{i=1}^n \gamma_{ki} \left((\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)(\widetilde{\boldsymbol{p}}_i - \widetilde{\boldsymbol{\mu}}_k)^T - \widetilde{\sigma}^2 \boldsymbol{I} \right) + (1 - \alpha_k) \left(\boldsymbol{\Sigma}_k + (\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)(\boldsymbol{\mu}_k - \widetilde{\boldsymbol{\mu}}_k)^T \right), \tag{14}$$

which perturbs the covariance by the noise remaining in the pre-filtered image $\bar{\boldsymbol{x}}$.

4. EXPERIMENTAL RESULTS

4.1. Experiment Settings

We compare our proposed method with BM3D [4] and EPLL [17], which are among the best performing methods for image denoising. The default GMM in EPLL is learned from 2,000,000 randomly chosen 8×8 patches. This default GMM is used as the generic prior for the proposed EM adaptation.

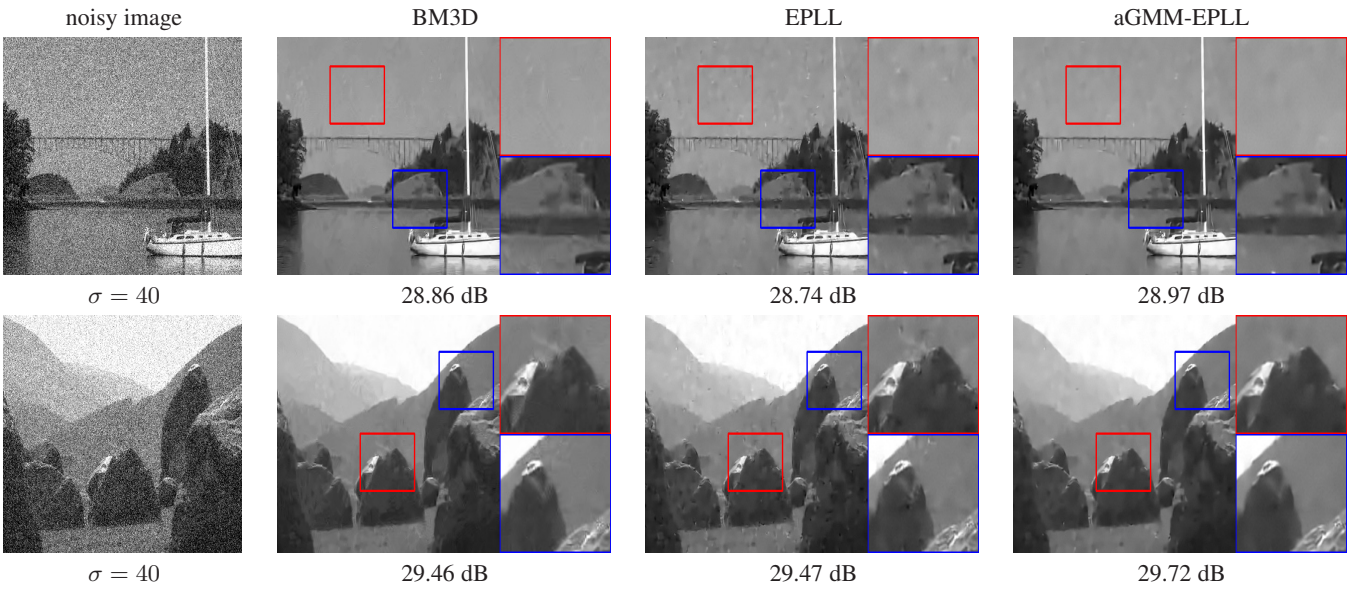


Fig. 2. Image denoising with the adapted GMM on the EPLL denoised image: Visual comparison and objective comparison. The test images are of size 481×321.

		BM3D	aGMM -BM3D	EPLL	aGMM -EPLL	aGMM -clean
Boat	$\sigma = 20$	29.59	**29.87**	29.78	**29.94**	30.47
	$\sigma = 40$	26.19	**26.66**	26.54	**26.70**	27.01
	$\sigma = 60$	24.65	**24.72**	24.74	**24.85**	25.11
	$\sigma = 80$	**23.39**	23.27	23.38	**23.40**	23.62
	$\sigma = 100$	**22.60**	22.41	22.60	**22.58**	22.77
Cameraman	$\sigma = 20$	30.24	**30.34**	30.24	**30.40**	31.33
	$\sigma = 40$	26.83	**27.35**	26.98	**27.31**	27.97
	$\sigma = 60$	25.31	**25.36**	25.30	**25.54**	26.25
	$\sigma = 80$	**23.88**	23.77	23.70	**23.88**	24.53
	$\sigma = 100$	**22.95**	22.83	22.75	**22.90**	23.48
House	$\sigma = 20$	33.65	**33.77**	32.98	**33.56**	34.58
	$\sigma = 40$	30.47	**30.84**	29.90	**30.71**	31.54
	$\sigma = 60$	**28.60**	28.36	27.64	**28.32**	29.02
	$\sigma = 80$	**27.36**	27.18	26.61	**27.18**	27.75
	$\sigma = 100$	**25.75**	25.68	25.18	**25.64**	25.96
Lena	$\sigma = 20$	31.61	**31.76**	31.44	**31.79**	32.68
	$\sigma = 40$	27.83	**28.23**	28.07	**28.36**	28.81
	$\sigma = 60$	**26.38**	26.22	26.04	**26.31**	26.59
	$\sigma = 80$	**24.93**	24.83	24.53	**24.83**	25.04
	$\sigma = 100$	**23.99**	23.89	23.63	**23.91**	24.03
Montage	$\sigma = 20$	**33.57**	33.52	32.55	**33.23**	34.63
	$\sigma = 40$	29.13	**29.39**	28.37	**29.06**	30.21
	$\sigma = 60$	26.56	**26.66**	25.97	**26.59**	27.60
	$\sigma = 80$	24.96	**24.98**	24.38	**24.95**	25.74
	$\sigma = 100$	23.72	**23.76**	23.27	**23.64**	24.48
Peppers	$\sigma = 20$	31.23	**31.50**	31.23	**31.52**	32.40
	$\sigma = 40$	27.43	**27.96**	27.72	**28.03**	28.52
	$\sigma = 60$	25.78	**25.87**	25.68	**25.99**	26.38
	$\sigma = 80$	24.27	**24.45**	24.11	**24.50**	24.70
	$\sigma = 100$	23.05	**23.28**	22.91	**23.30**	23.58
Average		26.86	**26.96**	26.61	**26.96**	27.56

Table 1. PSNR results for standard images of size 256×256.

We consider three versions of EM-adaptation: (1) An oracle adaptation by adapting the generic prior to the ground truth clean 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*. We set the parameter ρ to be 1 and experimental results show that the performance is insensitive to ρ being in the range of 1 and 10.

4.2. Denoising Standard Test Images

Figure 2 shows a visual comparison of two natural images. It can be seen that the proposed method yields the best result in terms of PSNR values. In Table 1, we report the PSNR values for different noise variances on 6 standard testing images of size 256×256. Comparing aGMM-EPLL with EPLL, we observe that the proposed aGMM-EPLL is consistently better than EPLL with an average gain of more than 0.3 dB. In some cases, aGMM-EPLL is even better than BM3D, e.g., Boat, Lena and Peppers. If we further look at the oracle aGMM-clean, the performance gain is even higher. We remark that all the adapted GMMs are learned from the pre-filtered images. If one uses the standard EM algorithm to learn a GMM from the pre-filtered image, the result will be several dBs worse.

5. CONCLUSION

We proposed an EM adaptation method to learn image priors. The proposed adaptation method allows us to use the parameters of a generic Gaussian mixture model and adapt to the image of interest. Compared to the classic EM algorithm, the proposed EM adaptation requires significantly fewer training samples and could be applied to any pre-filtered image. We also proposed modifications to compensate for the noise remaining in the pre-filtered images. EM adaptation is a computationally efficient method. Experimental results showed that the proposed method outperforms EPLL. Detailed analysis of the algorithm and automated estimation of the internal parameters will be studied in our future work.

6. REFERENCES

[1] 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.

[2] C. Kervrann and J. Boulanger, "Local adaptivity to variable smoothness for exemplar-based image regularization and representation," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 45–69, 2008.

[3] S. H. Chan, T. Zickler, and Y. M. Lu, "Monte Carlo nonlocal means: Random sampling for large-scale image filtering," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3711–3725, Aug. 2014.

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

[6] 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)*, Apr. 2009, pp. 1–6.

[7] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE Intl. Conf. Computer Vision (ICCV'09)*, Sep. 2009, pp. 2272–2279.

[8] 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.

[9] 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.

[10] 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.

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

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

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

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

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

[16] J. Sulam and M. Elad, "Expected patch log likelihood with a sparse prior," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, 2015, pp. 99–111.

[17] 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)*, Nov. 2011, pp. 479–486.

[18] 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.

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

[20] 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)*, May 2014, pp. 2469–2473.

[21] 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.

[22] S.J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[23] 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.

[24] M. R. Gupta and Y. Chen, *Theory and use of the EM algorithm*, Now Publishers Inc, 2011.

[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 Processing Systems 22*, pp. 1033–1041. Curran Associates, Inc., 2009.

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

[28] 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.

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

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

[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. on Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.