An Optimized Pixel-Wise Weighting Approach for Patch-Based Image Denoising

Jianzhou Feng*, Li Song, Xiaoming Huo, Xiaokang Yang, and Wenjun Zhang

***Abstract*—Most existing patch-based image denoising algorithms filter overlapping image patches and aggregate multiple estimates for the same pixel via weighting. Current weighting approaches always assume the restored estimates as independent random variables, which is inconsistent with the reality. In this paper, we analyze the correlation among the estimates and propose a bias-variance model to estimate the Mean Squared Error (MSE) under various weights. The new model exploits the overlapping information of the patches; it then utilizes the optimization to try to minimize the estimated MSE. Under this model, we propose a new weighting approach based on Quadratic Programming (QP), which can be embedded into various denoising algorithms. Experimental results show that the Peak Signal to Noise Ratio (PSNR) of algorithms like K-SVD and EPLL can be improved by around 0.1dB under a range of noise levels. This improvement is promising, since it is gained independent to which image model is used, especially when the gain from designing new image models becomes less and less.**

***Index Terms*—Image denoising, K-SVD, EPLL.**

EDICS Category: IMD-ANAL

I. Introduction

Image denoising is one of the most classical image processing problems; it aims to restore an image under random additive white Gaussian noise. Many state-of-the-art image denoising algorithms are based on image patches [1], [2], [3], [4], [5], [6], [7], [8], [9]. Their denoising methods can be interpreted as an iteration of a so-called *Filtering and Weighting* (F&W) process. Under the F&W process, local image patches are firstly restored through filtering, and then multiple estimates of the same pixel from overlapping restored patches are weighted to derive the final estimate. For the filtering method, sophisticated patch-based image models have been applied to generate the filters, e.g., the sparse coding model [1], [3], [5], [6], [9], the Gaussian Mixture Model [7], [8], and the non-local similarity model [2], [4]. In contrast, the weighting methods are still somewhat straightforward, either using simple averaging or deriving the weight independently based on certain transform coefficients of the corresponding image patch itself [2], [4], [9]. This form of weighting method is optimal when the estimates for weighting are independent random variables. However, the estimates can be heavily correlated due to overlapping of the patches, which violates the assumption of independence. Therefore, we may further improve the denoising performance by analyzing the correlation among the estimates using the overlapping information.

Based on the above idea, in Section II, we describe the F&W process precisely, analyze the Mean Squared Error (MSE) under various weights, and derive a bias-variance model to estimate it accurately. We also show that optimizing the weight under the proposed model yields the minimum MSE with the help of the overlapping information. In Section III, we propose a new weighting approach to solve the optimization problem under the bias-variance model via Quadratic Programming (QP). In Section IV, we introduce the proposed weighting approach into the K-SVD algorithm and the EPLL algorithm. It indicates that the Peak Signal to Noise Ratio (PSNR) of both algorithms can be improved by around 0.1dB under a range of noise levels. Finally, Section V concludes the paper.

II. The Bias-Variance Model

In this section, we first formulate the degradation model of image denoising and describe the F&W process in an analytic way. Then we propose a bias-variance model to characterize the correlation of the restored estimates under the F&W process. The new model can estimate the MSE under various weights "faithfully" by exploiting the overlapping information of the restored patches. Therefore, optimizing the weight under this model is nearly equivalent to minimizing the real MSE.

A. The Degradation Model and the F&W Process

The degradation model of image denoising can be formulated as

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{x}$ denote the (vectorized) clean image, $\mathbf{y}$ is its noisy version, and $\mathbf{n}$ represents the additive white Gaussian noise with variance σ^2.

Under the above notations, one can represent the F&W process for each pixel i as:

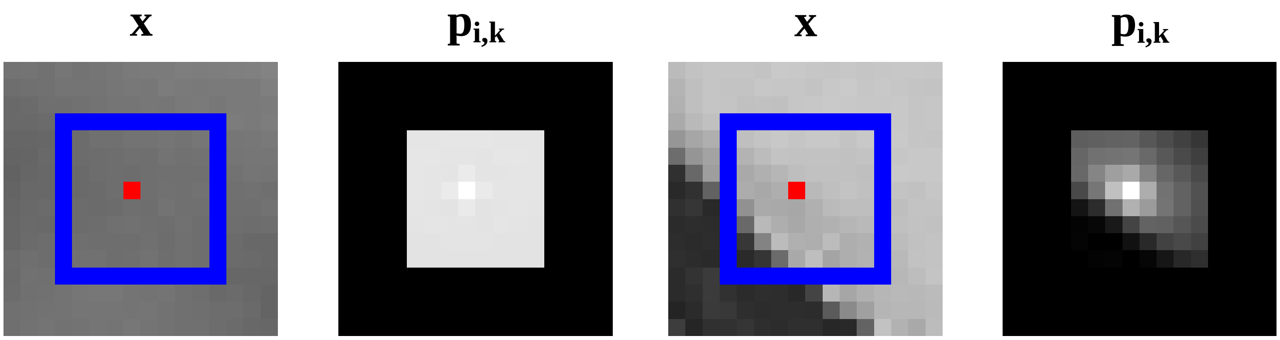
(1) *Filtering within local regions*: Suppose there are m_i local regions (also known as patches) that share pixel i. In the k-th region, x_i is estimated as

$$\hat{x}_{i,k} = \mathbf{p}_{i,k}^T \mathbf{y} + c_{i,k}, \tag{2}$$

where $(\mathbf{p}_{i,k}, c_{i,k})$ can be seen as a low-pass global filter. Various denoising algorithms compute $(\mathbf{p}_{i,k}, c_{i,k})$ in their own way, but the values are just slightly different. For example,

J. Feng, L. Song, X. Yang and W. Zhang are with Shanghai Digital Media Processing and Transmission Key Lab, Shanghai Jiaotong University, Shanghai, 200240 China, e-mail: {bravefjz, song_li, xkyang, zhangwenjun}@sjtu.edu.cn.

X. Huo is with School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332 USA, e-mail: xiaoming@isye.gatech.edu.



(a) A smooth region (b) A region contains an edge

Fig. 1. Two examples of $\mathbf{x}$ and $\mathbf{p}_{i,k}$. In the image of $\mathbf{x}$ (the left one), the red point indicates pixel i and the blue block represents the k-th local region. In the image of $\mathbf{p}_{i,k}$ (the right one), the lighter the pixel, the larger the corresponding element in $\mathbf{p}_{i,k}$.

in EPLL [7], suppose pixel i is at the l-th place of the k-th (vectorized) local region $\mathbf{R}_{i,k}\mathbf{y}$, where $\mathbf{R}_{i,k}$ is a selection matrix, then the patch is Wiener filtered to be

$$(\mathbf{\Sigma}+\sigma^2\mathbf{I})^{-1}(\mathbf{\Sigma}\mathbf{R}_{i,k}\mathbf{y}+\sigma^2\boldsymbol{\mu}), \tag{3}$$

where $\mathbf{\Sigma}$ and $\boldsymbol{\mu}$ are the parameter of a Gaussian distribution. In this case, $\mathbf{p}_{i,k}$ equals to the l-th column of

$$\left((\mathbf{\Sigma}+\sigma^2\mathbf{I})^{-1}\mathbf{\Sigma}\mathbf{R}_{i,k}\right)^T, \tag{4}$$

and $c_{i,k}$ equals to the l-th element of $\sigma^2(\mathbf{\Sigma}+\sigma^2\mathbf{I})^{-1}\boldsymbol{\mu}$.

Due to the property of $\mathbf{R}_{i,k}$, $\mathbf{p}_{i,k}$ is a sparse vector with non-zero elements only in the k-th local region, and its j-th element reflects the closeness between x_i and x_j. As illustrated in Fig. 1, the i-th element of $\mathbf{p}_{i,k}$ is always the largest, and if the local region in $\mathbf{x}$ contains two smooth areas like in Fig. 1 (b), the j-th element of $\mathbf{p}_{i,k}$ is close to 0 when pixel j is in the other area; $c_{i,k}$ is a bias term of the filter.

(2) *Weighting throughout local regions*: The m_i estimates are weighted to derive the final estimate of x_i as

$$\hat{x}_i=\sum_{k=1}^{m_i} w_{ik}\hat{x}_{i,k}, \tag{5}$$

where the weights w_{ik}'s are nonnegative and sums to one.

Denote $\mathbf{P}_i=(\mathbf{p}_{i,1},\cdots,\mathbf{p}_{i,m_i})$, $\mathbf{c}_i=(c_{i1},\cdots,c_{im_i})^T$ and $\mathbf{w}_i=(w_{i1},\cdots,w_{im_i})^T$, we have

$$\hat{x}_i=\mathbf{w}_i^T\left(\mathbf{P}_i^T\mathbf{y}+\mathbf{c}_i\right). \tag{6}$$

All the denoising algorithms in [1], [2], [3], [4], [5], [6], [7], [8], [9] fit the F&W process quite well. As for the Non-local Means algorithm [10], though it can be seem as a weighting algorithm without filtering, the weights actually reflect the closeness among pixels, which is mainly what $\mathbf{p}_{i,k}$'s do under the F&W process. Hence, NLM is more proper to be interpreted as a global filtering process with only one estimate for each pixel.

B. Two components in MSE

Under the F&W process, we assume $(\mathbf{P}_i,\mathbf{c}_i)$ in (6) is computed exactly using the original filtering method of a denoising algorithm. Therefore, $\hat{x}_i$ is formulated as a function of $\mathbf{w}_i$, and $\mathrm{MSE}(\hat{\mathbf{x}})$ is formulated as

$$\begin{aligned}\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))&=\frac{1}{M}\left(\hat{x}_i(\mathbf{w}_i)-x_i\right)^2\\&=\frac{1}{M}\left(\mathbf{w}_i^T\left(\mathbf{P}_i^T\mathbf{x}+\mathbf{c}_i\right)-x_i+\mathbf{w}_i^T\mathbf{P}_i^T\mathbf{n}\right)^2,\end{aligned} \tag{7}$$

where M is the number of pixels in $\mathbf{x}$ and $\mathbf{w}$ is denoted as the concatenation of all $\mathbf{w}_i$'s.

Since $\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))$ is a random variable depend on noise $\mathbf{n}$, we propose a bias-variance model, which estimates it by its expectation under $\mathbf{n}$. For mathematical derivation simplicity, we assume that $(\mathbf{P}_i,\mathbf{c}_i)$'s are independent to $\mathbf{n}$. Hence, the expectation can be estimated as

$$\hat{\mathbb{E}}\left[\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))\right]=\frac{1}{M}\sum_{i=1}^{M}\left(\mathrm{Bias}^2(\hat{x}_i(\mathbf{w}_i))+\mathrm{Var}(\hat{x}_i(\mathbf{w}_i))\right), \tag{8}$$

where

$$\mathrm{Bias}(\hat{x}_i(\mathbf{w}_i))=\mathbb{E}[\hat{x}_i(\mathbf{w}_i)]-x_i=\mathbf{w}_i^T(\mathbf{P}_i^T\mathbf{x}+\mathbf{c}_i)-x_i, \tag{9}$$

is the bias of $\hat{x}_i(\mathbf{w}_i)$ to x_i, and

$$\mathrm{Var}(\hat{x}_i(\mathbf{w}_i))=\mathbb{V}[\hat{x}_i(\mathbf{w}_i)]=\sigma^2\mathbf{w}_i^T\mathbf{P}_i^T\mathbf{P}_i\mathbf{w}_i \tag{10}$$

is the variance.

In reality, $(\mathbf{P}_i,\mathbf{c}_i)$'s are derived from $\mathbf{y}$, which makes them still correlated to $\mathbf{n}$, i.e., the assumption that leads to (8) may be violated. To evaluate the appropriateness of using $\hat{\mathbb{E}}\left[\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))\right]$ to approximate $\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))$, we compute their ratio

$$r(\mathbf{w})=\hat{\mathbb{E}}\left[\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))\right]/\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w})) \tag{11}$$

under various $\mathbf{w}$'s. If $r(\mathbf{w})$ is a constant under all $\mathbf{w}$'s, then we can conclude that optimizing (8) is equivalent to minimizing the true value of $\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))$. Experimental results done on several standard images under three representative denoising algorithms, K-SVD [1], EPLL [7], and BM3D [2] validate this guess. As shown in Table I, under each image and denoising algorithm combination, the values of $r(\mathbf{w})$ under an averaging $\mathbf{w}$ and a uniformly sampled $\mathbf{w}$ are really close. We only list these two values for illustration due to space limitation, the

TABLE I
$r(\mathbf{w})$ UNDER (IMAGE, DENOISING ALGORITHM) COMBINATIONS. IN EACH COMBINATION, THE LEFT ONE USE AVERAGING $\mathbf{w}$ AND THE RIGHT ONE USE A UNIFORMLY SAMPLED $\mathbf{w}$.

Image \ Alg.	K-SVD		EPLL		BM3D	
Lena	0.99	0.98	0.87	0.87	0.96	0.96
Barbara	0.99	0.99	0.93	0.93	0.85	0.85
House	0.99	0.98	0.89	0.89	0.90	0.90
Peppers	0.97	0.97	0.90	0.90	0.88	0.88

value of $r(\mathbf{w})$ under other $\mathbf{w}$'s are also quite close to the presented ones. Therefore, we denote the objective function as

$$f(\mathbf{w}) = M \cdot \hat{\mathbb{E}}\left[\mathrm{MSE}(\hat{\mathbf{x}}(\mathbf{w}))\right] = \sum_{i=1}^{M} \mathbf{w}_i^T \left(\mathbf{Q}_i + \sigma^2 \mathbf{P}_i^T \mathbf{P}_i\right) \mathbf{w}_i, \tag{12}$$

where

$$\mathbf{Q}_i = (x_i \mathbf{1} - \mathbf{P}_i^T \mathbf{x} - \mathbf{c}_i)(x_i \mathbf{1} - \mathbf{P}_i^T \mathbf{x} - \mathbf{c}_i)^T. \tag{13}$$

In current weighting approaches, $(\hat{x}_{i,k} - x_i)$'s, for $1 \leq k \leq m_i$, are assumed to be independent random variables with zero means, so that their covariance matrix is assumed to be a diagonal matrix. Among them, the most promising one computes the weight as the inverse of the sparsity of the transform coefficients [2], [9]. Though it has been validated under a shift-invariant DCT transform based denoising algorithm [9], experimental results show that the same weighting method doesn't work for K-SVD and EPLL, which use simple averaging originally. The reason is: on one side, in [9], some estimates are not so good when the block DCT basis can not represent the patches sparsely, so that the weighting strategy performs well by giving such estimates small weights; on the other side, in K-SVD and EPLL, all the estimates are comparable since the transform is more adaptive and always yields sparse representations, which makes this strategy ineffective.

While under the bias-variance model, there is no independence assumption, and the covariance matrix is derived analytically as

$$\mathrm{Cov}_i = \mathbf{Q}_i + \sigma^2 \mathbf{P}_i^T \mathbf{P}_i \tag{14}$$

as shown in (12). Such Cov_i is superior than any diagonal matrix because it retain the overlapping information of different local regions by computing $\mathbf{P}_i^T \mathbf{P}_i$. It is easy to see that each element of $\mathbf{P}_i^T \mathbf{P}_i$ is a inner product of two $\mathbf{p}_{i,k}$'s. As mentioned in Section II-A, $\mathbf{p}_{i,k}$ has zero elements outside local region k and the value of its nonzero elements is mostly dependent on how close is the corresponding x_j to x_i. Therefore, the inner product of $\mathbf{p}_{i,k_1}$ and $\mathbf{p}_{i,k_2}$ can be seen as the total squared closeness to x_i of the overlapped pixels in local region k_1 and k_2.

III. THE QP BASED WEIGHTING APPROACH

In this section, we propose a Quadratic Programming (QP) based weighting approach for optimizing $f(\mathbf{w})$. This approach contains two profiles. In Section III-A, we propose the "approximation" profile, which optimizes $f(\mathbf{w})$ with an approximation matrix $\hat{\mathbf{Q}}_i$. In Section III-B, we propose the "practical" profile, which computes the optimal weight as a linear combination of two weights, each minimizes the bias and the variance component separately, with a practically derived combination coefficient.

A. The Approximation Profile

There is unknown pixel values $\mathbf{x}$ contained in $\mathbf{Q}_i$. Hence, before optimizing $f(\mathbf{w})$, we need to approximate $\mathbf{Q}_i$ first based on $\mathbf{y}$ and $(\mathbf{P}_i, \mathbf{c}_i)$'s. For simplicity, like previous weighting methods assume Cov_i as a diagonal matrix, we assume $\mathbf{Q}_i$ here as a diagonal matrix $\hat{\mathbf{Q}}_i$, while still retain the overlapping information of patches in $\mathbf{P}_i^T \mathbf{P}_i$. The k-th diagonal entry of $\mathbf{Q}_i$ is

$$(\mathbf{Q}_i)_{kk} = (x_i - \mathbf{p}_{i,k}^T \mathbf{x} - c_{i,k})^2. \tag{15}$$

We approximate it as

$$(\hat{\mathbf{Q}}_i)_{kk} = \left(\bar{x}_i - \mathbf{p}_{i,k}^T \mathbf{y} - c_{i,k}\right)^2 + \varepsilon, \tag{16}$$

where

$$\bar{x}_i = \frac{1}{m_i} \sum_{k=1}^{m_i} \hat{x}_{i,k} = \frac{\mathbf{1}^T (\mathbf{P}_i^T \mathbf{y} + \mathbf{c}_i)}{m_i} \tag{17}$$

is the mean of all the $\hat{x}_{i,k}$'s and ε is a small parameter to ensure the entry to be positive.

Under this approximation, the optimal weight is

$$\begin{aligned} \mathbf{w}^* = \arg\min_{\mathbf{w}} & \sum_{i=1}^{M} \mathbf{w}_i^T \left(\hat{\mathbf{Q}}_i + \sigma^2 \mathbf{P}_i^T \mathbf{P}_i\right) \mathbf{w}_i, \\ \text{subject to} \quad & \mathbf{w}_i \geq 0,\ \mathbf{1}^T \mathbf{w}_i = 1, \text{for } i = 1, \cdots, M. \end{aligned} \tag{18}$$

It is easy to see that each $\mathbf{w}_i$ can be solved independently via QP. We also note that using the Lagrangian multiplier method based on $\mathbf{1}^T \mathbf{w}_i = 1$ may lead to negative elements in $\mathbf{w}_i$, which means constraint $\mathbf{w}_i \geq 0$ is non-trivial.

B. The Practical Profile

The approximation profile can be interpreted under a more general linear combination framework. We can see that $\mathbf{w}^*$ in (18) can be approximated by

$$\mathbf{w}(\lambda) = (1 - \lambda)\mathbf{u} + \lambda \mathbf{v}, \tag{19}$$

with a certain $\lambda \in [0, 1]$, where

$$\mathbf{u} = \arg\min_{\mathbf{w}} \sum_{i=1}^{M} \mathbf{w}_i^T \hat{\mathbf{Q}}_i \mathbf{w}_i \tag{20}$$

and

$$\mathbf{v} = \arg\min_{\mathbf{w}} \sum_{i=1}^{M} \mathbf{w}_i^T \mathbf{P}_i^T \mathbf{P}_i \mathbf{w}_i \tag{21}$$

under the same constraints as in (18).

In the practical profile, we compute the optimal weight as $\mathbf{w}(\lambda^*)$, where $\lambda^* \in \mathbb{R}$ is a practically learned real scalar that yields the minimum averaged MSE on a training image set. When $\hat{\mathbf{Q}}_i$ is a good approximation of $\mathbf{Q}_i$, λ^* would be within $[0, 1]$ so that both profiles can improve the denoising performance. Otherwise, λ^* may be negative and only the practical profile performs well under this case.

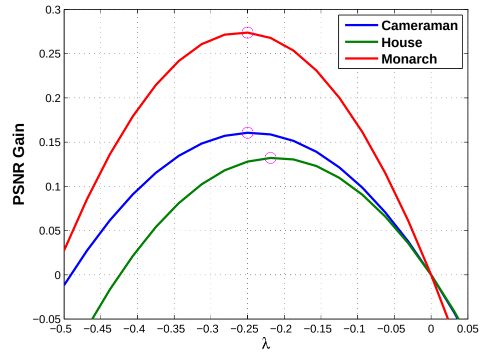


Fig. 2. The PSNR gain by using $\mathbf{w}(\lambda)$ under $\sigma = 50$ for K-SVD. Each curve represent one training image and the circle indicates the position of the optimal λ^* that leads to the maximum gain for that image.

In practice, we find that $\hat{\mathbf{Q}}_i$'s are quite likely to be a positive scalar times the identity matrix, which makes $\mathbf{u}$ to be the averaging weight. Thus we modify $\mathbf{u}$ to be the original weight of a certain denoising algorithm. This reduces the computational cost of solving (20) and makes the original algorithm as a special case under the practical profile by setting $\lambda^* = 0$.

IV. Numerical Experiments

In this section, we test the proposed weighting approach under three representative denoising algorithms: K-SVD, EPLL, and BM3D. Due to space limitation, we list the experimental results done on frequently used standard images in image denoising domain here, and list the other supportive results on our webpage.

For the K-SVD algorithm, we find the denoising performance can be improved most significantly under high noise levels using the practical profile with negative λ^*'s. For each noise level, λ^* is pre-learned from a training set with three standard images. Taking $\sigma = 50$ as an example, as shown in Fig. 2, we compute the PSNR gain of using $\mathbf{w}(\lambda)$ for $\lambda \in \mathbb{R}$, and find that $\lambda^* = -0.25$ can lead to almost the maximum PSNR gain for any of the training images. Therefore, we set $\lambda^* = -0.25$ under $\sigma = 50$ to denoise all the images. After the training process, we apply the practical profile to the other 8 standard images under σ from 30 to 50, and compare the PSNR with the original K-SVD algorithm. As shown in Table II, the averaged PSNR gain increases from about 0.1dB to 0.2dB as σ increases.

For the EPLL algorithm, we apply the proposed weighting approach for multiple times since it is an iterative algorithm. We find both of the two profiles are effective for moderate noise levels, while the approximation profile performs even better. Therefore, we choose the approximation profile to improve the EPLL algorithm. As shown in Table III, for noise level from $\sigma = 5$ to $\sigma = 20$, the averaged PSNR gain can reach around 0.1dB. The proposed weighting approach is not effective for high noise levels probably because: It is designed to minimize the MSE under only one F&W process. When it is used for multiple times, minimizing the MSE within each iteration may not be the optimal. As the noise level increases, the number of iteration also increases, which enlarges the impact of the misleading objective.

TABLE II
PSNR comparison under K-SVD. Under each noise level, the left column uses the original weight, the right column uses weight of the practical profile.

Image \ σ	30		40		50	
Peppers	28.82	28.95	27.31	27.51	26.12	26.36
Straw	24.66	24.83	22.85	23.07	21.53	21.73
Lena	30.41	30.53	28.94	29.10	27.74	27.88
Barbara	28.52	28.70	26.83	27.07	25.39	25.62
Boat	28.41	28.52	27.03	27.16	25.90	26.06
Man	28.23	28.34	27.06	27.19	26.07	26.21
Hill	28.45	28.55	27.11	27.24	26.29	26.39
Fingerprint	26.29	26.40	24.71	24.87	23.23	23.52
Average	27.97	28.10	26.48	26.65	25.28	25.47

TABLE III
PSNR comparison under EPLL. Under each noise level, the left column uses the original weight, the right column uses weight of the approximation profile.

Image \ σ	5		10		20	
Cameraman	38.18	38.32	34.01	34.12	30.36	30.44
House	39.11	39.19	35.77	35.82	33.16	33.30
Monarch	38.40	38.63	34.35	34.50	30.58	30.73
Peppers	38.03	38.12	34.54	34.65	31.24	31.35
Straw	35.53	35.65	30.89	31.01	27.04	27.16
Lena	38.69	38.73	35.60	35.67	32.70	32.79
Barbara	37.71	37.79	33.63	33.72	29.80	29.87
Boat	36.91	36.86	33.66	33.74	30.71	30.77
Fingerprint	36.60	36.70	32.18	32.29	28.38	28.48
Hill	37.11	37.08	33.45	33.50	30.42	30.46
Man	37.82	37.90	33.96	34.04	30.61	30.66
Average	37.64	37.73	33.82	33.91	30.45	30.55

For the BM3D algorithm, we find the PSNR improvement by using the proposed weighting approach is insignificant, no matter which profile is used. This is probably because BM3D has much more estimates for the same pixel compare to K-SVD and EPLL, and their correlation is also more complicated, which makes approximating the hidden covariance matrix Cov_i in (14) accurately very hard. Therefore, we need to design more sophisticated profiles for BM3D in the future.

V. Conclusion

In this paper, we propose a bias-variance model to estimate the MSE accurately by analyzing the correlation among the estimates. We then propose a new weighting approach that contains two profiles using QP. The proposed weighting approach optimize the weights by preserving the overlapping information of restored patches. Experimental results show that the PSNR gain of K-SVD and EPLL can be improved by about 0.1dB under a range of noise levels. The 0.1dB improvement is promising, since it is independent to which image model is used, especially when the gain from designing new image models becomes less and less.

This work setup a novel bias-variance model that formulates the selection of weights as an optimization problem. The proposed two profiles for solving this optimization problem can be seen as a stepping stone, and better profiling methodology may be proposed with more sophisticated techniques.

References

[1] 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.

[2] 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.

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE Int. Conf. Computer Vision*, 2009, pp. 2272–2279.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," in *PROC. WORKSHOP ON SIGNAL PROCESSING WITH ADAPTIVE SPARSE STRUCTURED REPRESENTATIONS (SPARS09*, 2009.

[5] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1641–1644.

[6] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Jun. 2011.

[7] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. IEEE Int. Conf. Computer Vision*, 2011, pp. 479–486.

[8] J. Feng, L. Song, X. Huo, X. Yang, and W. Zhang, "Image restoration via efficient gaussian mixture model learning," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013.

[9] O. G. Guleryuz, "Weighted averaging for denoising with overcomplete dictionaries," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 3020–3034, 2007.

[10] A. Buades, B. Coll, and J. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Jun. 2005.