A FAST AND ROBUST DEBLURRING TECHNIQUE ON HIGH NOISE ENVIRONMENT

Pulak Purkait and Bhabatosh Chanda

ECSU, Indian Statistical Institute
203, B. T. Road, Kolkata-700108

ABSTRACT

To have a unique solution of an ill-posed inverse problem, the usual way is to embed prior information in terms of regularizer or smoothness criterion. In this work, both the inverse mechanism (the relationship of blur and sharp patches) and the smoothness prior are learned simultaneously from the image itself, in multiple scales. We have shown experimentally that the proposed method outperform the existing state-of-the-art techniques on high noise environment and produce comparable result otherwise; moreover, it is almost three times faster than existing ones.

***Index Terms*—** Deblurring, patch matching, high-frequency generation, smoothness prior, self similarity.

1. INTRODUCTION

Observed images of a scene are usually degraded by blurring due to atmospheric turbulence and inappropriate camera settings. The images are farther degraded by the various noises present in the environment and the system. Therefore, it is essential to get a sharp clean image from the noisy blurred image. The relation between the noisy blurred image Y and the desired sharp image X can be represented as [1][2]

$$Y = HX + e \tag{1}$$

where Y, X and e represent lexicographically ordered column vectors of the observed image, desired sharp image and the additive noise respectively. H is the blurring matrix incorporating camera lens/CCD blurring as well as atmospheric blurring.

Image deblurring (i.e., solving (1)) is one of the classical inverse problems in image processing and it has been studied in both frequency and spatial domains extensively in last few decades. In frequency domain Wiener filter [3] is the most popular one that minimizes the mean square error. In spatial domain, the solution can be obtained by solving an unconstrained optimization problem, i.e.

$$\hat{X} = \arg\min_{X}\{\frac{1}{2}\|HX - Y\|_2^2 + \mu\Upsilon(X)\} \tag{2}$$

where μ is the regularization parameter (also known as Lagrange multiplier) that controls the emphasis between the data error term (first term) and the regularization term (second term). $\Upsilon(X)$ mostly represent the smoothness criterion by means of energy in first or second order derivative. Note that deblurring and de-noising are contradicting requirement. Deblurring method is an inverse problem and in the noisy environment, it is an extremely ill-posed problem. There is a sequence of work done based on iterative shrinkage thresholding algorithm (FISTA) [2], (MFISTA) [4] for finding out best possible regularization term $\Upsilon(X)$ and solving (2) efficiently. The TV regularization [4] [2] (L_1 norm of the gradients or high-frequency wavelet components) is the most popular one.

There are some work done on patch based image restoration with the sparsity and nonlocal assumption of patches. Recently patch based processing getting more attention and some patch based algorithms [5] are giving excellent result. The another approach [6][7] combines the power of both the approaches: frequency domain (Wiener filter) and spatial domain (patch-based). SA-DCT [8] and BM3D [6] are two of the popular algorithms which are developed by combining the power of both the approaches and are still the state-of-the-art methods for image restoration.

We observe that in very low noisy environment BM3D[6] can produce good result; however, in high noisy condition output is not satisfactory. In this work we develop a patch-based deblurring method which can produce better result in case high noise than that of state-of-the-art methods and comparable in low noise environment. We learn the image smoothness prior $\Upsilon(X)$ and also the blurred and sharp image patch pairs from the image itself. It is well established that in a natural image, an image patch reoccurs with almost certainty in the image itself either in the same scale or lower scale [9]. So it would be very natural choice to learn these correspondence and natural image prior from the image itself.

This method is motivated by the fast single frame super resolution method by Glasner et. al. [10], where they learn the low resolution and high resolution patch correspondence from the image itself and get excellent result. Here we propose a learning strategy of blurred-sharp patch pairs and also image smoothness prior from the image itself in multiple scales. The proposed algorithm uses PatchMatch[11] architecture and can produce excellent output in almost real time. Moreover, proposed approach is more robust to noise present in the image. The only tuning parameters are the size of the patch and the number of iterations in back-projection. However, those parameters are not very critical in terms of producing outputs.

The rest of the paper is organized as follows. In the next section we describe the main intuition behind the proposed algorithm, followed by a detail description. In Section 3, we show some results and finally conclude with some pros and cons of the proposed algorithm and future work direction in Section 4.

2. PROPOSED METHODOLOGY

In this section we describe the proposed method in detail. Our approach contains three main steps (i) de-noising the blurred and noisy image, (ii) generating initial sharp image based on the learning the blur and sharp patch pairs, and (iii) finally back-projecting the generated sharp image to the denoised blurred image to get the actual sharp deblurred image. The main steps of the algorithm can be summarized in the flow-chart 1 and in the algorithmic form as follows:

Algorithm 1 : Deblur an image in high noise environment

1. **Input : Noisy blurred image Y and blur kernel or the blurring operator H.**
2. **Denoise the image by wavelet hard-thresholding to generate the noise-free blurred image $\hat{Y}$.**
3. **Downsample the original noisy blurred image Y into smaller scales and blur them by H. We do this in multiple scales to generate a pair of blurred and sharp image pyramids $(H \downarrow_n Y; \downarrow_n Y)$.**
4. **Then from these pair of blurred-sharp image pyramids learn the blurred and sharp patch correspondence (x, y) and replace each blurry patch y (overlapping) of the denoised blurred image by the corresponding sharp patch x from the pyramid.**
5. **Do the aggregation of the overlapping portions of estimated sharp patches to get the sharp image $\hat{X}$.**
6. **At the end back-project the estimated high-frequency image $\hat{X}$ into the blurred and denoised image $\hat{Y}$ to get the sharped deblurred image X.**

These steps are described in detail as follows.

2.1. Denoising the noisy and blur Image

We use an adaptive wavelet hard thresholding technique [1] to denoise the noisy blurred image. The threshold is derived from a Bayesian framework, where the prior used on the wavelet coefficients is the generalized Gaussian distribution. It is called BayesShrink threshold and is obtained by minimizing a Bayesian risk with squared error.

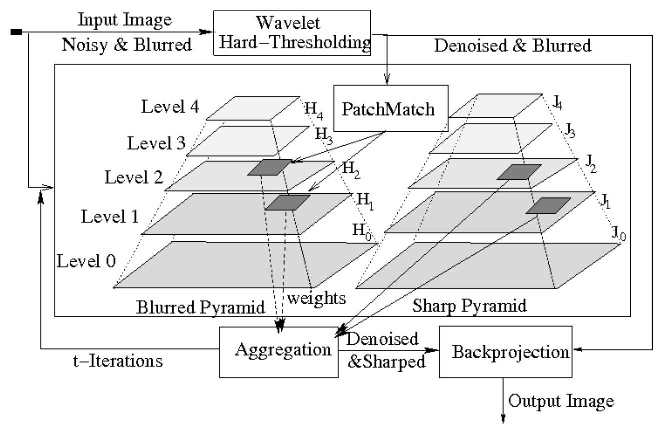

Fig. 1. Flow chart of the proposed deblurring technique.

2.2. High-frequency Generation technique

We learn the blur and sharp patch correspondence from the input image itself. As depicted in figure 1, a pair of blur and sharp pyramid $(H \downarrow_n Y; \downarrow_n Y)$ is generated by resizing the original image Y into lower resolutions (blurring and down-sampling $\downarrow_n$) and then blurred them by the blurring kernel H. Now the patches of this pair of pyramids comprise our database of patch pairs consisting of blur and sharp patches (Hx, x). Moreover, as pointed out earlier, each patch would occur again in the same image either in the same scale or in other scales almost certainly. Hence, in our case it is highly probable that each patch of the denoised, but blur image $\hat{Y}$ (after previous step) would occur in one of the images of the blurred pyramid $H \downarrow_n Y$ and the corresponding patch from the sharp pyramid would mimic the actual sharp patch. Therefore, we compare each patch (overlapping) of the denoised blur image $\hat{Y}$ with the each patch of the images in the blurred pyramid $H \downarrow_n Y$ and find out the best matching patch (nearest neighbor). Then copy the corresponding sharp patch from the sharp pyramid $\downarrow_n Y$ to the appropriate location and aggregate them to get initial sharp image $\hat{X}$. The aggregation in the overlapping region of the neighboring sharp patches is obtained by the weighted sum of the patches. Note that the comparison between patches is done based on SSD between patches and the corresponding weights are obtained by applying sigmoid over the SSD values.

Since a blurred image when downsampled essentially becomes a sharp image. i.e $\downarrow_n Y$ is a sharp image X_n (Say). In the pyramid pair, one contains blur images HX_n and the other the sharp X_n. Therefore, learning patch pairs from this pyramid pair actually minimizes the data error term of eq. (2) in an *ad hoc* way. Moreover, the regularization term $\Upsilon(\hat{X})$, which is assumed to be a smoothness prior, is satisfied as the estimated sharp patches are obtained by aggregating the actual patches from the images (of different resolution) of the sharp pyramid.

Under the high noise environment, after generating the initial sharp image, we refine this image by applying same learning mechanism for a few more iterations. However, in this case we build the pyramid pair from the output image $\hat{X}$ of the previous iteration instead of the original image Y. We have seen experimentally that 2-3 iterations are good enough for such condition. Since we learn the blur-sharp patch pairs from the image itself and all of the comparison/aggregation are done at patch level, the proposed method is more robust than the existing algorithms. We have explored this in the experimental section.

We use efficient PatchMatch [11] algorithm for finding out the best match (nearest neighbor) of a patch. This algorithm is very efficient and does the comparison in almost real time. Moreover, since most of the comparisons are done on downsampled images of the pyramid, the method is sufficiently fast. In fact, we have experimentally shown that our system is about three times faster than the state-of-the-art methods.

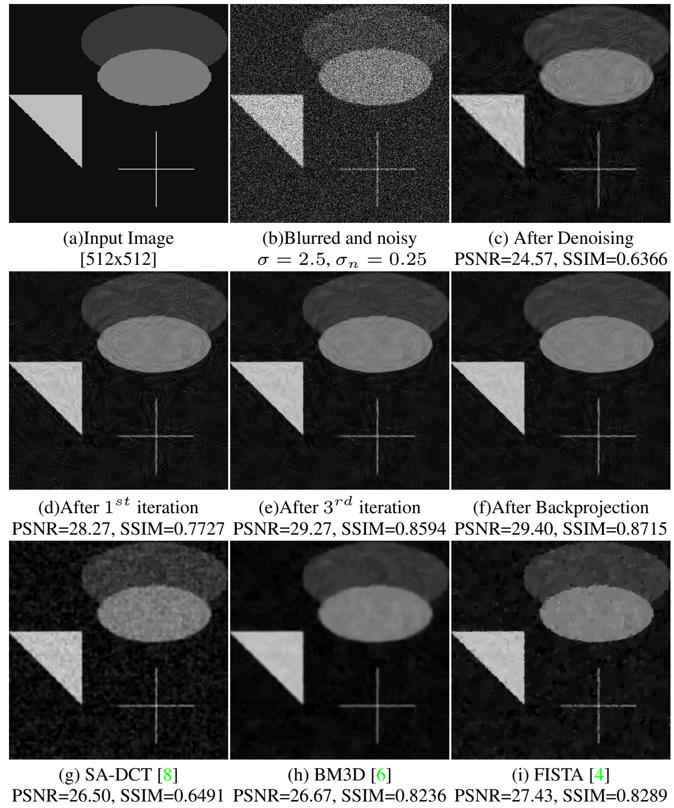

Fig. 2. Comparison of reconstruction result of a synthesized image using proposed method over some existing methods: (a)-(b) original and blurred noisy image, (c)-(f) different steps of proposed method, (g)-(i) results using some state-of-arts algorithm. **For better comparison kindly view the soft version of the paper**.

2.3. Back-projection to satisfy the constraints

The fundamental constraint of deblurring like inverse problems is that after re-degrading, the image X should become consistent with the input noisy blurred image Y as shown in equation eq. (2). In other words, after denoising the input image $\hat{Y}$ should be a blurred version of the inherent sharp image X, which can be expressed as

$$\hat{Y} = HX. \tag{3}$$

Therefore, our estimated sharp image is generated satisfying this constraint by using an iterative method as

$$X_{t+1} = X_t + \gamma H^T(\hat{Y} - HX_t) \tag{4}$$

where X_t is the estimated deblurred image after the t-th iteration and γ is an iterative constant representing the step of updation. We initialize X_0 by the sharp image $\hat{X}$ generated in the previous step. The result of the back-projection X^* is the final estimate of deblurred image. In all our experiments, we have iterated (4) until the solution converges to a predefined tolerance limit and get good results.

3. EXPERIMENTAL SECTION

The proposed deblurring technique is implemented using MATLAB 7.6 on a Linux OS with 8GB of RAM and a 3.6-GHz Intel processor. We have used mex code for some of the subroutines to speed up the process. We have chosen a small down-sampling factor ($s = 1.5$) while generating the pyramid pair and the images are re-sized into lower scale and the corresponding blurred image is generated by applying blur kernel H over the resized images.

In our first experiment, we synthesize a gray-scale image, blur it by a Gaussian kernel (std σ = 2.5) and then add Gaussian noise (std σ_n = 0.25). We apply the proposed deblurring technique and some of the state-of-the-art methods on this image for comparison. Detail results including quantitative quality comparison (psnr & ssim) are shown in figure 2. Next experiment is done on several standard images degraded with fixed blurring Gaussian kernel (std σ= 2.5) and different amount of noise (std σ_n= 0.03 to 0.27). We measure the quantitative quality in terms of psnr and ssim in each of the experiments and then plot the average values over all the images versus amount of noise as shown in figure 3. It is observed that the proposed method produces better result compared to most of the state-of-the-art methods under high noise condition; however it gives comparable result in low noise environment. In the table 1, we show the quality of results of different methods along with the proposed one for individual image under Gaussian noise with standard deviation $\sigma_n = 0.2$. The best quantitative values are shown in bold font. Clearly, in most of the cases the proposed method outperforms the others. In figure 4, we have displayed the result of one of the

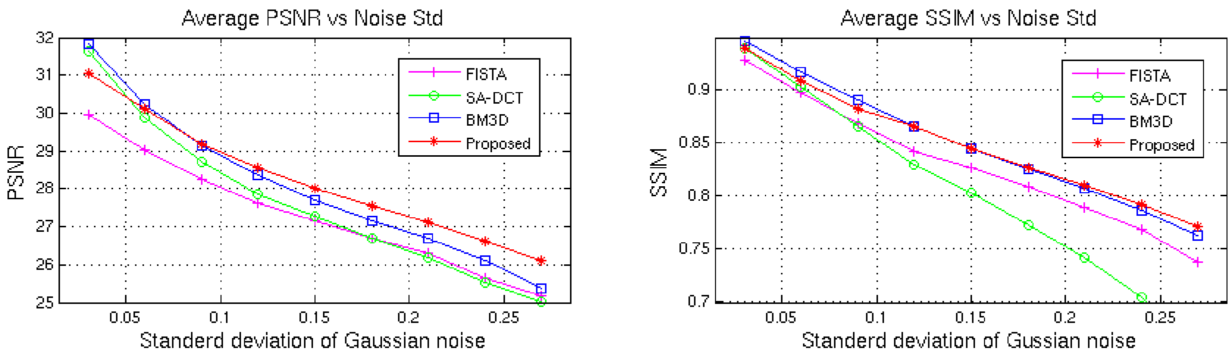

Fig. 3. Analysis of the performance of proposed method over existing methods applied on different gray images and then average PSNR and average SSIM are plotted. Details are on text.

image (lena). Table 2 depicts the average unit time required

(a)Input noisy blurred $\sigma = 2.5, \sigma_n = 0.12$ (b) BM3D [6] PSNR=27.18, SSIM=0.8253 (c) Proposed PSNR=27.53, SSIM=0.8279

Fig. 4. Comparison of reconstruction result of a synthesized image using proposed method over some existing methods for Lena image.

by different methods as in Table 1, which shows that the proposed algorithm is about 3 times faster than the state-of-the-art algorithms while generating similar or better result. We have not explored any efficient data structure to speed-up the process, hence timet can be further reduced by adopting an efficient one. While working with color images, we consider patches as 3D blocks and do all the comparison (nearest neighbor search) and aggregation (weighted sum) over the same 3D blocks. Detail results on color images, more results on gray level images and the source code are available on our project webpage http://www.isical.ac.in/ṽlrg/?q=node/12.

4. CONCLUSION

In this work, we have developed a fast and robust deblurring technique based on patch learning mechanism from the input noisy blur image itself. Robustness and efficiency issues require a much detailed and careful analysis and that is beyond the scope of the current paper. We mention these issues here to indicate that the proposed system can further be improved by paying closer attention to these issues. It is shown experimentally that while the proposed method can produce comparable or better result compared to state-of-the-art methods and it is 3 times faster than existing ones. However, the proposed method is not suitable for deblurring under asymmetric/sparse kernel. In future work we would explore those issues in detail and hopefully address other inverse methods for imaging as well.

Table 1. Psnr (dB) and ssim (dB) results of deblurred images (noise level $\sigma_n = 0.2$)

Images	FISTA	SA-DCT	BM3D	Proposed
Peppers	26.15	26.11	26.46	**26.69**
	0.8368	0.7831	**0.8458**	0.8378
Man	25.02	25.01	25.11	**25.31**
	0.7114	0.7077	0.7207	**0.7231**
Barbara	22.88	22.92	23.15	**23.20**
	0.7085	0.6881	0.7282	**0.7315**
Boat	24.69	24.78	24.95	**25.14**
	0.7590	0.7169	0.7639	**0.7758**
Baboon	20.19	20.31	20.31	**20.39**
	0.5089	0.5393	0.5286	**0.5538**
Splash	28.86	28.41	**29.29**	29.21
	0.8550	0.7705	**0.8575**	0.8490
Tiffany	27.72	27.47	28.12	**28.20**
	0.7950	0.7460	0.8118	**0.8182**
Zelda	28.32	28.47	29.10	**29.42**
	0.8101	0.7955	0.8413	**0.8550**
Airplane	24.83	24.86	24.91	**25.31**
	0.8194	0.7468	0.8138	**0.8240**
Average	25.40	25.37	25.71	**25.87**
	0.7560	0.7219	0.7679	**0.7743**

Table 2. Time comparison of proposed method over different methods.

Method	FISTA	SA-DCT	BM3D	Poposed
Time	19.7s	10.6s	11.4s	4.1s

5. REFERENCES

[1] S. Grace Chang, Bin Yu, and Martin Vetterli, "Adaptive wavelet thresholding for image denoising and compression.," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1532–1546, September 2000.

[2] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems.," *SIAM J. of Imaging Sciences*, vol. 2, no. 1, pp. 183–202, August 2006.

[3] B. Widrow and S.D. Stearns, "Adaptive signal processing," *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.*, vol. 1, 1985.

[4] P. C. Hansen, J. G. Nagy, and D. P. OLeary, "Deblurring images," *Fundamentals of Algorithms. Society for Industrial and Applied Mathematics (SIAM).*, vol. 3, no. 8, pp. 291–294, August 2006.

[5] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 457–464.

[6] 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, August 2007.

[7] A.Danielyan, V. Katkovnik, and K. Egiazarian, "Bm3d frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, April 2007.

[8] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images.," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, May 2007.

[9] M. Zontak and M. Irani, "Internal statistics of a single natural image," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 977–984.

[10] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image.," in *In Proceedings International Conference on Computer Vision*. IEEE, 2009, pp. 349–356.

[11] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24:1–24:11, July 2009.