FAST AND ROBUST RECURSIVE FILTER FOR IMAGE DENOISING

Yiheng Chi and Stanley H. Chan

School of ECE and Dept of Statistics, Purdue University, West Lafayette, IN 47907.

ABSTRACT

Image denoising on mobile cameras requires low complexity, but many state-of-the-art denoising methods are computationally intensive. We present a low complexity denoising algorithm using an edge-aware recursive filter (RF). We make two contributions. First, we modify the original RF so that it is significantly more robust when estimating the gradients from noisy inputs. We extend the RF to high-order for texture and heavy noise images. Second, we introduce a SURE-based image fusion technique. We show that while individual RFs have different performance, the fused result is often better. Experimental results show that the new RF performs much faster than other denoisers while providing good quality images.

***Index Terms*—** Image denoising, recursive filter, edge-preserving, mobile camera

1. INTRODUCTION

1.1. Motivation

While image denoising algorithms over the past decade have produced many promising results [1–12], the computational complexity of the majority remains high. In order to support future imaging applications on mobile cameras and miniature robots, there is a pressing need for simple but effective algorithms. The goal of this paper is to present a new method with low complexity while offering satisfactory performance relative to state-of-the-arts.

The key element of our proposed method is the domain transform edge-aware recursive filter by Gastal and Oliveira [13, 14] (or RF in short). RF is an infinite impulse response (IIR) approximation to the standard edge-aware filters, e.g., bilateral filter [15] or non-local means [1]. The complexity of RF is low because the filter weights are recursively estimated. However, the original RF was designed for graphics applications such as stylization, abstraction, and contrast enhancement, etc. One implicit assumption is that the image to be processed must be clean so that the distance can be accurately approximated. Image denoising does not fit the framework because the input is noisy.

Modification of the RF to image denoising was first proposed by Wong and Milanfar [17]. Their denoiser, termed Turbo Denoiser, enables RF to handle noisy inputs by using a boosting procedure and a look-up table to identify important edges for boosting. When the noise is spatially varying, it was reported that Turbo Denoiser performs similarly to BM3D; Yet, when the noise is i.i.d., there is unfortunately no data for comparison. Reproducing results of Turbo Denoiser is difficult, because the look-up table is proprietary and details of many steps are not discussed.

(a) noisy $\sigma = 20/255$ (b) Our 3rd-order RF, 30.19dB, 0.279 sec / .mex (c) NLM, 29.97dB, 8.24 sec / .mex (d) BM3D, 31.65dB, 3.408 sec / .mex

Fig. 1. Comparison of image denoisers. For fair comparison, we implement all denoisers on the same MATLAB .mex platform. BM3D [3] is run using the default settings. NLM [1] uses the implementation in [16] with a search window 15×15 and patch size 7×7.

In this paper, we present an improved recursive filter for fast and robust denoising. To provide readers a quick overview, we show in Figure 1 a comparison between non-local means [1], BM3D [3] and the proposed method on i.i.d. Gaussian noise. We implement all algorithms on the same MATLAB .mex platform to ensure fairness. As shown in the figure, although BM3D generates better PSNR, its runtime is 10 times more than the proposed method.

E-mail: {chi14, stanchan}@purdue.edu. The work was supported, in part, by the U.S. National Science Foundation under Grant CCF-1718007.

1.2. Contributions

This paper has two key contributions summarized as follows.

First, we modify the original recursive filter for denoising. We extend the pixel-wise gradient estimation to patch-wise estimation to improve the robustness against noisy input. We develop high-order RF to handle heavy noisy and texture images. Both our 1st and 3rd order RF are significantly better than the original RF by Gastal and Oliveira [13].

Second, we present a SURE-based image fusion technique by taking convex combinations of the recursive filter outputs. As will be discussed, 1st-order and 3rd-order RF have different performance regimes. Our fusion technique combines the two denoisers and yields better results.

2. PROPOSED METHOD

2.1. The Original First Order RF

We start by reviewing the first order recursive filter originally proposed by Gastal and Oliveira [13]. Denoting $I[n]$ as the noisy image and $J[n]$ as the denoised image, RF computes $J[n]$ by using a recursion:

$$J[n] = (1 - a^{d[n]})I[n] + a^{d[n]}J[n-1]. \quad (1)$$

In this equation, the scalar constant $a = \exp(-\sqrt{2}/\sigma_s)$ is a user defined parameter controlling the relative emphasis of $I[n]$ and $J[n-1]$. The spatially varying power $d[n]$ is the gradient of $I[n]$ at n:

$$d[n] = 1 + \frac{\sigma_s}{\sigma_r}\Big|I[n] - I[n-1]\Big| = 1 + \frac{\sigma_s}{\sigma_r}\Big|\nabla I[n]\Big|, \quad (2)$$

where σ_s and σ_r are the spatial and range parameters, respectively. Since (2) can be pre-computed prior to the recursion, and $J[n]$ is computed independently for each row, the overall recursion in (1) is very fast to compute. For 2D signals, one can sequentially perform (1) forward and backward along two directions (totally 4 times). For color images, one repeat this sequence of operations for each color.

The reason why (1) is an edge-aware filter can be seen by expanding the recursion:

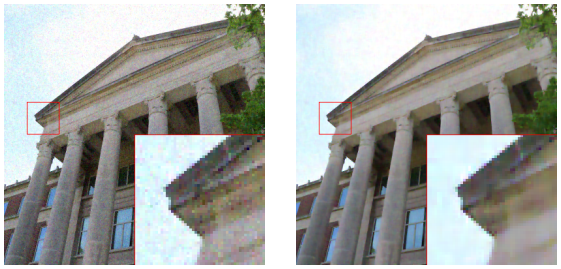
$$J[n] = \sum_{\ell=0}^{n}\left(\prod_{k=0}^{\ell} a^{d[n-k+1]}\right)(1 - a^{d[n-\ell]})I[n-\ell], \quad (3)$$

with the assumption that $d[n+1] = 0$. Substituting (2) into (3), we can show that the product inside the summation is

$$\prod_{k=0}^{\ell} a^{d[n-k+1]} = \exp\left\{-\frac{\sqrt{2}}{\sigma_s}\sum_{k=0}^{\ell}\left(1 + \frac{\sigma_s}{\sigma_r}\left|\nabla I[n-k]\right|\right)\right\},$$

which is essentially the bilateral filter weight with the range part approximated by the sum of absolute difference. Thus if there is an edge, the gradient will be large and so the exponential will be small.

The performance of the RF depends heavily on the accuracy of $d[n]$, which, in turn, depends on the quality of $I[n]$. If $I[n]$ is noisy, then (2) will become an under estimate of true gradient magnitude. When this happens, no matter how we tune the parameters σ_s and σ_r, the final denoising performance will still be degraded. Figure 2 illustrates an example.



(a) Original RF [13] 27.38dB (b) Our 1st-order RF 30.77dB

Fig. 2. Comparison between the original RF in [13] and the proposed 1st order RF. Noise level is $\sigma = 20/255$. Parameters of both methods are optimized to produce the best results.

2.2. Proposed First Order RF

If $I[n]$ is noisy, then the most straight-forward solution to improve the robustness is to adopt a pre-filtering step to generate a less noisy "reference" [12, 18]. Then we can use the reference for computing the gradient. Besides, we modify (2) by using a standard technique in NLM [1]:

$$d[n] = 1 + \frac{\sigma_s}{\sigma_r}\max\left(\Big|\nabla I[n]\Big| - \sigma, 0\right), \quad (4)$$

which is a soft-thresholding of the gradient magnitudes so that any $\Big|\nabla I[n]\Big|$ smaller than σ will become 0.

In addition to these two "standard" steps, we extend the pixel-wise distance to a patch-based distance as the latter is more robust to noise. Intuitively, this modification is analogous to the patch distance in NLM [1] relative to the pixel distance in bilateral filter [15].

The conventional definition of the patch-distance Δ between the n-th and the m-th patch is

$$\Delta(I[n], I[m]) = \sum_{p=0}^{P-1} h[p]\,\Big|I[n-p] - I[m-p]\Big|,$$

where P is the dimensionality of a patch, and $h[p]$ is the weight kernel. In RF, we modify the definition of $d[n]$ as

$$d[n] = 1 + \frac{\sigma_s}{\sigma_r}\max\left\{\sum_{p=0}^{P} h[p]\,\Big|\nabla I[n-p]\Big| - \sigma, 0\right\}. \quad (5)$$

It is not difficult to see that the complexity of computing $d[n]$ is significantly lower than $\Delta(I[n], I[m])$ because $d[n]$ is independent of m.

2.3. Proposed High-Order RF

One limitation of the 1st-order RF is that the distance $d[n]$ will become less accurate when noise level increases and when image contains texture. This is attributed to the fact that the sum of the gradients in (4) fails when there are multiple edges within the range of the summation.

In order to enable RF for high-noise and heavy texture problems, we extend the 1st-order RF to high-order RF. The idea is to generalize the recursion as

$$J[n] = b_0 I[n] + \sum_{i=1}^{Q} \alpha_i a^{d_i[n]} J[n-i], \tag{6}$$

where Q specifies the order of the RF. In our work, we find that $Q = 3$ is sufficient and higher orders have diminishing benefits. The filter coefficients $\{b_0, \alpha_1, \ldots, \alpha_Q\}$ are fixed constants. The i-th distance $d_i[n]$ is defined as

$$d_i[n] = i + \frac{\sigma_s}{\sigma_r} \sum_{p=0}^{P} h[p] \Big| I[n-p] - I[n-i-p] \Big|, \tag{7}$$

which replaces the one pixel finite difference $I[n] - I[n-1]$ in (2) by the i-th order finite difference $I[n] - I[n-i]$. Geometrically, $d_i[n]$ *skips* pixels when estimating the distance so that it is more tolerant to textures.

When using (6), we need to pay extra attention to the normalization because the transfer function (i.e., the frequency response) of (6) should have a unit gain. To solve this problem, we consider an all one signal $\mathbf{1}[n] = 1$ for all n, and define $J[n]$ as the ratio of two terms $J_{\text{num}}[n]$ and $J_{\text{den}}[n]$:

$$J[n] = \frac{J_{\text{num}}[n]}{J_{\text{den}}[n]}, \tag{8}$$

where $J_{\text{num}}[n]$ is computed using (6) with the noisy $I[n]$, and $J_{\text{den}}[n]$ is computed using $I[n] = \mathbf{1}[n]$.

The performance of the 3rd-order RF compared to the 1st-order RF is shown in Figure 3. In this example, we choose a heavy texture pattern image with high noise level $\sigma = 50/255$. We compared the PSNR between the two RFs and we also compute the SSIM score to verify the visual quality. It is clear from the result that the 3rd-order RF has higher PSNR while maintaining the SSIM score.

2.4. SURE-based Image Fusion

While the results in Figure 3 shows that 3rd-order RF has better denoising performance, the gap could be less substantial for images with less textures. In some cases, such as `cameraman`, 1st-order RF is actually better. To achieve a uniformly better performance of the two methods, we present a simple but effective image fusion technique.

Denote $J_{1\text{st}}[n]$ and $J_{3\text{rd}}[n]$ as the results of 1st-order and 3rd-order RF, respectively. Our goal is to find a weight λ such

1st-order RF
22.66dB, 0.900

3rd-order RF
23.40dB, 0.907

Fig. 3. Comparison between 1st order RF and 3rd order RF on a texture-heavy image. The noisy level is $50/255$. Note that while both methods have almost the same SSIM score, 3rd-order RF has substantially better PSNR.

that the linearly combined result

$$\widehat{J}[n] = \lambda J_{1\text{st}}[n] + (1-\lambda) J_{3\text{rd}}[n] \tag{9}$$

will have better denoising performance. Intuitively, what this linear combination does is to balance the undersmoothing and oversmoothing of the two denoisers.

To choose the weight λ, we start by considering the oracle case where the ground truth $J^*[n]$ is available. When $J^*[n]$ is available, we can compute the PSNRs of the denoised results: $\eta_{1\text{st}} \overset{\text{def}}{=} \text{PSNR}(J_{1\text{st}}[n], J^*[n])$ and $\eta_{3\text{rd}} \overset{\text{def}}{=} \text{PSNR}(J_{3\text{rd}}[n], J^*[n])$. Accordingly, we can define the weight $w_{1\text{st}}$ (and $w_{3\text{rd}}$) as

$$w_{1\text{st}} = \exp\left\{ -\frac{\left(\eta_{1\text{st}} - \max\left(\eta_{1\text{st}}, \eta_{3\text{rd}}\right)\right)^2}{2\sigma_\eta^2} \right\}, \tag{10}$$

where σ_η is a constant. The interpretation of (10) is that among the two denoisers, one of them produces a higher PSNR $\max\left(\eta_{1\text{st}}, \eta_{3\text{rd}}\right)$. If $\eta_{1\text{st}}$ is close to the higher PSNR (or if it is the higher PSNR), then it is desirable to have $w_{1\text{st}}$ large. The cutoff is controlled by σ_η, which is one standard deviation of a Gaussian. Typically, we choose $\sigma_\eta = 1$, meaning that any PSNR with more than 1dB deviation from the maximum is regarded as insignificant. Once $w_{1\text{st}}$ and $w_{3\text{rd}}$ are defined, the weight constant λ in (9) is

$$\lambda = \frac{w_{1\text{st}}}{w_{1\text{st}} + w_{3\text{rd}}}, \quad \text{and} \quad 1 - \lambda = \frac{w_{3\text{rd}}}{w_{1\text{st}} + w_{3\text{rd}}}. \tag{11}$$

In the absence of ground truth $J^*[n]$, one way to estimate the PSNR is by means of Stein's Unbiased Risk Estimator (SURE) [19]. Here, we use the Monte-Carlo SURE [20] because the recursive filter does not allow us to derive closed-form expressions. Given the SURE estimates, we can compute the corresponding PSNR estimates as

$$\text{PSNR}\Big(J_{1\text{st}}[n], J^*[n]\Big) \approx 10 \log_{10} \text{SURE}\Big(J_{1\text{st}}[n], I[n]\Big),$$

Image	BM3D	NLM	$J_{1st}[n]$	$J_{3rd}[n]$	$\widehat{J}[n]$	BM3D	NLM	$J_{1st}[n]$	$J_{3rd}[n]$	$\widehat{J}[n]$
			$\sigma = 10$					$\sigma = 20$		
Baboon512	33.12	30.11	31.33	32.41	32.41	29.05	26.74	26.86	28.41	28.42
Cameraman256	34.16	31.04	33.50	33.33	33.48	30.44	28.27	29.76	29.64	29.79
House256	36.69	34.75	35.11	35.12	35.23	33.86	32.40	32.15	32.25	32.31
Man512	33.96	32.03	33.06	33.33	33.43	30.54	29.20	29.45	29.76	29.90
Peppers256	34.69	33.03	33.90	33.82	34.10	31.23	29.97	30.32	30.20	30.50
16-image average PSNR	34.62	32.44	33.51	33.75	33.87	31.36	29.71	29.97	30.29	30.41
16-image average SSIM	0.915	0.872	0.895	0.901	0.902	0.855	0.805	0.815	0.824	0.829
Runtime /s	2.33	13.13	0.05	0.14	0.35	2.74	14.34	0.05	0.12	0.35
			$\sigma = 30$					$\sigma = 50$		
Baboon512	26.82	25.02	24.68	26.20	26.18	24.27	23.39	22.96	23.86	23.86
Cameraman256	28.69	26.54	27.59	27.54	27.81	26.03	24.46	24.45	24.76	25.02
House256	32.16	30.70	29.87	30.05	30.38	29.93	28.38	27.17	27.23	27.30
Man512	28.81	27.89	27.64	28.03	28.14	26.75	26.07	25.80	25.95	26.11
Peppers256	29.32	28.33	27.92	27.89	28.18	26.78	25.66	24.78	24.93	25.00
16-image average PSNR	29.55	28.19	27.97	28.29	28.45	27.27	26.13	25.75	25.84	26.08
16-image average SSIM	0.810	0.759	0.757	0.761	0.771	0.743	0.686	0.678	0.657	0.676
Runtime /s	2.65	14.05	0.05	0.12	0.33	3.24	13.54	0.04	0.12	0.33

Table 1. Image denoising results: individual PSNR, average PSNR, average SSIM and runtime.

where in this equation we emphasize that SURE requires the denoised image $J_{1st}[n]$ and the noisy input $I[n]$.

The complexity of Monte-Carlo SURE is approximately twice of the individual denoiser, as the Monte-Carlo needs to probe the denoiser twice in order to estimate the divergence used in SURE. As a comparison with the 1st-order RF and the 3rd-order RF, we show in Figure 4 the results of $J_{1st}[n]$, $J_{3rd}[n]$ and $\widehat{J}[n]$. In this particular example, we observe that $J_{1st}[n]$ has better performance than $J_{3rd}[n]$ as the image is mostly flat. However, using the proposed SURE-based image fusion technique, we can boost the performance by about 0.1dB over $J_{1st}[n]$ and 0.3dB over $J_{3rd}[n]$.

(a) Noisy $\sigma = 20/255$ (b) $J_{1st}[n]$, 29.41dB

(c) $J_{3rd}[n]$, 29.23dB (d) $\widehat{J}[n]$, 29.54dB

Fig. 4. Denoising results of the SURE-based image fusion.

3. EXPERIMENTAL RESULTS

We compare the proposed RF with BM3D and NLM. We choose not to compare with neural network (NN) methods [21,22] because NN typically require GPU and large memory which are not always feasible on mobile cameras. To ensure fair comparison, we implement RF in MATLAB .mex so as to minimize the gap with BM3D. For NLM, we use the implementation by G. Peyre [16] as the official NLM [23] is C++ / Linux. For testing, we choose 16 standard test images from USC SIPI [24]. All experiments are run on a laptop with Intel i7 4.4GHz / 8MB Cache / Windows 10.

The results are shown in Table 4. At low noise $\sigma = 10, 20, 30$, both PSNR and SSIM improves from $J_{1st}[n]$ to $\widehat{J}[n]$. For $\sigma = 50$, although PSNR improves, the SSIM stays the same. One reason is that $J_{3rd}[n]$ is more susceptible to artifacts as it is intrinsically a bandpass filter. This suggests that RF is more suitable for low-noise problems. For runtime, RF is significantly faster than NLM and BM3D.

4. CONCLUSION

We present an edge-aware recursive filter (RF) for fast and robust image denoising. We offer three versions of the filter: 1st-order RF for very fast denoising, 3rd-order RF for improved performance with more computing, and a fusion method that produces the most robust result but requires longer runtime. Our experimental results show that while RF does not outperform advanced denoisers such as BM3D, its very short runtime makes it possible for potential applications on mobile cameras. We anticipate that additional speed-up can be achieved by porting the algorithm to optimized platforms, e.g., Halide / C++.

5. REFERENCES

[1] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Computer Vision and Pattern Recognition*, Jun. 2005, vol. 2, pp. 60–65 vol. 2.

[2] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct 2006.

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

[4] G. Peyré, "Image processing with nonlocal spectral bases," *Multiscale Modeling & Simulation*, vol. 7, no. 2, pp. 703–730, 2008.

[5] K. Hirakawa and P. J. Wolfe, "Efficient multivariate skellam shrinkage for denoising photon-limited image data: An empirical bayes approach," in *Proc. IEEE Int. Conf. Image Process.*, Nov 2009, pp. 2961–2964.

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

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

[8] 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.

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

[10] J. Salmon, Z. Harmany, C.-A. Deledalle, and R. Willett, "Poisson noise reduction with non-local pca," *J. Mathematical Imaging and Vision*, vol. 48, no. 2, pp. 279–294, 2014.

[11] E. Luo, S. H. Chan, and T. Q. Nguyen, "Adaptive image denoising by mixture adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4489–4503, Oct. 2016.

[12] S. H. Chan, T. Zickler, and Y. M. Lu, "Understanding symmetric smoothing filters: A Gaussian mixture model perspective," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5107–5121, Nov. 2017.

[13] E. S. L. Gastal and M. M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 69:1–69:12, July 2011.

[14] E. S. L. Gastal and M. M. Oliveira, "High-order recursive filtering of non-uniformly sampled signals for image and video processing," *Computer Graphics Forum*, vol. 34, no. 2, pp. 81–93, 2015.

[15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Int, Conf. Computer Vision*, Jan 1998, pp. 839–846.

[16] G. Peyré, "Toolbox non-local means," `http://www.mathworks.com/matlabcentral/fileexchange/13619`, 2007.

[17] T. S. Wong and P. Milanfar, "Turbo denoising for mobile photographic applications," in *IEEE Int. Conf. Image Process.*, Sept 2016, pp. 988–992.

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

[19] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, 1981.

[20] S. Ramani, T. Blu, and M. Unser, "Blind optimization of algorithm parameters for signal denoising by monte-carlo SURE," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, 05 2008, pp. 905 – 908.

[21] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, July 2017.

[22] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using convolutional auto-encoders with symmetric skip connections," Available online https://arxiv.org/abs/1606.08921, 2016.

[23] A. Buades, B. Coll, and J.-M. Morel, "Non-local means denoising," *Image Processing On Line*, vol. 1, pp. 208–212, 2011.

[24] University of Southern California, "The USC-SIPI image database," `http://sipi.usc.edu/database/database.php`.