DUAL-DOMAIN IMAGE DENOISING

Claude Knaus   *Matthias Zwicker*

University of Bern

ABSTRACT

Image denoising methods have been implemented in both spatial and transform domains. Each domain has its advantages and shortcomings, which can be complemented by each other. State-of-the-art methods like *block-matching 3D filtering* (BM3D) therefore combine both domains. However, implementation of such methods is not trivial. We offer a hybrid method that is surprisingly easy to implement and yet rivals BM3D in quality.

Index Terms— image denoising, bilateral filter, wavelet shrinkage, short-time Fourier transform

1. INTRODUCTION

The classic image denoising problem is the reconstruction of an image that has been degraded by addition of Gaussian white noise. There are two main classes of image denoising methods: one operates in the spatial domain, the other in a transform domain. The bilateral filter [1] and the non-local means filter [2] are examples of methods which define the filter kernel in the spatial domain. They preserve features like edges, but have difficulties preserving low-contrast details. On the other hand, wavelet thresholding and shrinkage methods operate in a transform domain and excel in preserving details like textures, but suffer from ringing artifacts near edges.

Thus, a hybrid approach is taken by recent works. BM3D [3], *shape-adaptive BM3D* (SA-BM3D) [4], and *BM3D with shape-adaptive principal component analysis* (BM3D-SAPCA) [5], sorted by increasing denoising quality, are considered state-of-the-art. These are sophisticated methods which pay for the high quality with implementation complexity [6]. While producing numerically good results, the methods are not yet perfect [7, 8, 9]. They are based on block matching, which introduces visible artifacts in homogeneous regions, manifesting as low-frequency noise.

We propose a method that is competitive in quality with BM3D, but is much simpler to implement. We combine two popular filters for the two domains. For the spatial domain, we use the bilateral filter, and for the transform domain we use the short-time Fourier transform (STFT) [10] with wavelet shrinkage. The bilateral filter is known for its edge-preserving properties. It retains high-contrast features like edges, but cannot preserve low-contrast detail like textures without introducing noise. STFT wavelet shrinkage on the other hand results in good detail preservation, but suffers from ringing artifacts near steep edges. We combine these two classic methods to produce a new one which denoises better than when used separately. Apart from operating in different domains, the bilateral filter and the STFT shrinkage are very alike; hence, we call our method *dual-domain image denoising* (DDID). The resulting algorithm is described in section 2, followed by implementation details including MATLAB code in section 3. Results are then presented in section 4, and section 5 concludes our exposition.

2. ALGORITHM

In this section, we describe our iterative dual-domain algorithm. The goal is to estimate the original image x from a noise-contaminated image $y = x + \eta$ with a stationary variance $\sigma^2 = \text{Var}\,[\eta]$. We observe that spatial domain methods excel at denoising high-contrast images while transform domain methods excel at low-contrast images. We therefore separate the image into two layers, and denoise them separately. The bilateral filter is appropriate for this decomposition, finding applications in detail enhancement [11] and HDR compression [12]. The high-contrast layer is the bilaterally filtered image, and the low-contrast layer is the residual image. Since the high-contrast image is already denoised, it only remains to denoise the low-contrast image in the transform domain using wavelet shrinkage. The original image can thus be approximated by the sum of the two denoised layers as

$$\tilde{x} = \tilde{s} + \tilde{S}, \qquad (1)$$

where $\tilde{s}$ and $\tilde{S}$ are the denoised high- and low-contrast images.

The above procedure only denoises a single amplitude band. The bilateral filter protects large amplitudes of the signal, while wavelet shrinkage discards small amplitudes. As we will see in 2.5, only the signals with amplitudes in between are denoised. We therefore denoise all amplitudes from large to small by iterating. Starting with the noisy input y, we use the denoised result of an iteration to guide the subsequent iteration. In the following three sections, we describe the steps for a single iteration: bilateral filter in 2.1, domain transform in 2.2, and wavelet shrinkage in 2.3. Finally, we extend the algorithm to color images in 2.4 and provide a discussion in 2.5.

2.1. Spatial Domain: Bilateral Filter

In the first step, we calculate the denoised high-contrast value $\tilde{s}_p$ for a pixel p using a joint bilateral filter [13]. Our joint bilateral filter uses the guide image g to filter the noisy image y. We define the bilateral kernel over a square neighborhood window $\mathcal{N}_p$ centered around every pixel p with window radius r. Since we want to guide not only the bilateral filter but also the wavelet shrinkage, we filter both guide and noisy images in parallel and obtain the two denoised high-contrast images

$$\tilde{g}_p = \frac{\sum_{q \in \mathcal{N}_p} k_{p,q}\, g_q}{\sum_{q \in \mathcal{N}_p} k_{p,q}} \qquad (2)$$

$$\tilde{s}_p = \frac{\sum_{q \in \mathcal{N}_p} k_{p,q}\, y_q}{\sum_{q \in \mathcal{N}_p} k_{p,q}} \qquad (3)$$

where the bilateral kernel is

$$k_{p,q} = e^{-\frac{|p-q|^2}{2\sigma_s^2}}\, e^{-\frac{(g_p - g_q)^2}{\gamma_r \sigma^2}}. \qquad (4)$$

The parameters σ_s and γ_r shape the spatial and range kernels respectively.

2.2. Domain Transform

In the second step, we prepare for the wavelet shrinkage in the transform domain (section 2.3) by extracting the low-contrast signals and performing the STFT. We obtain the low-contrast signals by subtracting the bilaterally filtered values $\tilde{g}_p$ and $\tilde{s}_p$ from g_p and y_p, followed by multiplication with the range kernel of Equation 4. The STFT is a discrete Fourier transform (DFT) preceded by multiplication of the signal with a window function to avoid boundary artifacts [10]. We choose the spatial Gaussian of the bilateral kernel as the window function, and the entire step becomes a Gabor transform of the low-contrast signal. To transition to the frequency domain, we perform a non-unitary DFT. The resulting coefficients $G_{p,f}$ and $S_{p,f}$ are defined for frequencies f in the frequency window $\mathcal{F}_p$ with the same size as $\mathcal{N}_p$. Assuming that the bilateral kernel $k_{p,q}$ is noise-free, the variance $\sigma_{p,f}^2$ of the noisy Fourier coefficients and the coefficients themselves are

$$\sigma_{p,f}^2 = \sigma^2 \sum_{q \in \mathcal{N}_p} k_{p,q}^2 \qquad (5)$$

$$G_{p,f} = \sum_{q \in \mathcal{N}_p} e^{-i2\pi(q-p)\cdot f/(2r+1)} k_{p,q} \left(g_q - \tilde{g}_p\right) \qquad (6)$$

$$S_{p,f} = \sum_{q \in \mathcal{N}_p} e^{-i2\pi(q-p)\cdot f/(2r+1)} k_{p,q} \left(y_q - \tilde{s}_p\right). \qquad (7)$$

The signal modifications prior to the DFT resemble shape-adaptive methods [4, 5, 14]. These methods fit a kernel in the shape of a polygon to locally homogeneous regions. However, a bilateral filter is simpler to implement and actually more adaptive to the signal. Shape-adaptive methods also subtract the mean of the signal to keep the remaining signal balanced around 0. Our observation is that subtracting the bilaterally filtered value is a better choice.

2.3. Frequency Domain: Wavelet Shrinkage

In the last step, we shrink the noisy Fourier coefficients $S_{p,f}$. We use shrinkage factors similar to the range kernel of the bilateral filter in Equation 4. The range kernel is designed to retain the near zero-mean noise and to discard the bias inducing signal. For the wavelet shrinkage factors $K_{p,f}$, we want the opposite: we keep the signal and discard the noise, so we take the reciprocal of the normalized Euclidean distance $\frac{\sigma_f^2}{|G_{p,f}|^2}$. The return to the spatial domain is fast, as we are only interested in the value at the center pixel p. The non-unitary inverse DFT over the frequency domain $\mathcal{F}_p$ is simply the mean of all shrunk wavelet coefficients, yielding the low-contrast value

$$\tilde{S}_p = \frac{1}{|\mathcal{F}_p|} \sum_{f \in \mathcal{F}_p} K_{p,f} \, S_{p,f} \qquad (8)$$

where the wavelet shrinkage factors are

$$K_{p,f} = e^{-\frac{\gamma_f \sigma_{p,f}^2}{|G_{p,f}|^2}}. \qquad (9)$$

Like the bilateral kernel $k_{p,q}$, the shrinkage factors $K_{p,f}$ are defined using the spectral guide $G_{p,f}$, and the wavelet shrinkage parameter γ_f plays a similar role as the bilateral range parameter γ_r.

2.4. Color Images

For color images, we make two modifications to the algorithm. Firstly, we filter in YUV-space using an orthogonal transformation similar to other methods [3, 15]. While the transformation does not need to be strictly orthogonal, having row vectors with unit length has the advantage that variances do not change. Secondly, in the range kernel of the bilateral filter in Equation 4, we calculate the normalized Euclidean distance $\frac{(g_p - g_q)^2}{\sigma^2}$ as the sum of the normalized distances of all channels. However, the STFT wavelet shrinkage is performed for every channel independently.

2.5. Discussion

We can gain an intuition about the collaboration of the two domains by denoising a 1D signal. We specify the input as a rectangular function modulated with a sine wave, to which Gaussian white noise was added. Figure 2 illustrates the intermediate steps in the spatial and frequency domain for the first two iterations. The first bilateral filter step uses a large range parameter γ_r. This retains the large steps of the rectangle, but smoothens the rest of the signal. The following STFT step recovers the previously lost detail without being affected by the edges of the rectangle. The filtered signal is fed as a guide to the second iteration step, which uses a smaller range parameter. This time, the bilateral filter keeps the recovered edges from the previous STFT step. Although the first STFT step reintroduced ringing artifacts, the bilateral filter recognizes them as noise and filters out. The second STFT step reinforces the detail features in the center but does not bring back the ringing. As observed, the joint bilateral filter has the power to "heal" ringing artifacts caused by the wavelet shrinkage of the preceding iteration; this phenomenon has previously been exploited by Yu et al. [16]. Figure 3 shows the evolution of the guide signal g_p over three iterations. With every iteration, the noise decreases, while only little bias is introduced. Figure 4 shows averaged plots over 200 denoised signals. They demonstrate that DDID avoids noisy ringing artifacts typical to STFT with Wiener filter.

Note that in the algorithm, the two filtering steps look similar. In both domains we measure meaningful coefficients y_q and $S_{p,f}$, which are then weighted with factors $k_{p,q}$ and $K_{p,f}$ using the noise statistics of the coefficients. Both filtered values $\tilde{s}_p$ and $\tilde{S}_p$ are essentially dot products. The main difference is that the first step is responsible for denoising high-contrast features and is thus a condition to the second step for denoising low-contrast features.

3. IMPLEMENTATION

The implementation of our algorithm is straightforward. The equations from section 2 directly translate to MATLAB code in Listing 1. We iterate three steps, as we did not observe any improvement beyond that. For the three iterations, we use empirically found bilateral range parameters $\gamma_r = \{100, 8.7, 0.7\}$ and wavelet shrinkage parameters $\gamma_f = \{4.0, 0.4, 0.8\}$. We set the window radius to $r = 15$, and the spatial kernel of the bilateral filter is specified by $\sigma_s = 7$. In theory, the Fourier coefficients $S_{p,f}$ in Equation 8 are conjugate symmetric and $K_{p,f}$ is symmetric which makes $\tilde{S}_p$ real. In practice, due to numerical errors, $\tilde{S}_p$ may become complex, in which case we drop the imaginary part.

```
function x = DDID(y, sigma2)
    x = step(y, y, sigma2, 15, 7, 100, 4.0);
    x = step(x, y, sigma2, 15, 7, 8.7, 0.4);
    x = step(x, y, sigma2, 15, 7, 0.7, 0.8);
end

function xt = step(x, y, sigma2, r, sigma_s, gamma_r, gamma_f)

    [dx dy] = meshgrid(-r:r);
    h = exp(- (dx.^2 + dy.^2) / (2 * sigma_s^2));
    xp = padarray(x, [r r], 'symmetric');
    yp = padarray(y, [r r], 'symmetric');
    xt = zeros(size(x));

    parfor p = 1:numel(x), [i j] = ind2sub(size(x), p);

        % Spatial Domain: Bilateral Filter
        g  = xp(i:i+2*r, j:j+2*r);
        y  = yp(i:i+2*r, j:j+2*r);
        d  = g - g(1+r, 1+r);
        k  = exp(- d.^2 ./ (gamma_r * sigma2)) .* h;    % Eq. 4
        gt = sum(sum(g .* k)) / sum(k(:));              % Eq. 2
        st = sum(sum(y .* k)) / sum(k(:));              % Eq. 3

        % Fourier Domain: Wavelet Shrinkage
        V  = sigma2 .* sum(k(:).^2);                    % Eq. 5
        G  = fft2(ifftshift((g - gt) .* k));           % Eq. 6
        S  = fft2(ifftshift((y - st) .* k));           % Eq. 7
        K  = exp(- gamma_f * V ./ (G .* conj(G)));      % Eq. 9
        St = sum(sum(S .* K)) / numel(K);              % Eq. 8

        xt(p) = st + real(St);                         % Eq. 1
    end
end
```

Algorithm 1: *MATLAB code of Dual-Domain Image Denoising. This code reproduces all grayscale images in this paper.*

4. RESULTS

The implementation of our algorithm produces competitive results. Table 1 compares the peak signal-to-noise ratio (PSNR) of DDID and BM3D for all the BM3D test images[*]. We chose $\sigma = 25$ (PSNR = 20.17 dB) as the standard deviation of the noise. Numerically, BM3D and DDID show comparable denoising quality.

Figure 1[+] demonstrates that low-frequency noise present in BM3D images is absent in DDID images. Figure 5 shows another strength of DDID. The error comparison shows that DDID has smaller errors than BM3D for hair texture. BM3D on the other hand works well for edge-like structures as found in architecture or in the blue ridges in the cheeks of the mandrill.

Figure 6 studies noise induced artifacts. Random noise can generate patterns which can be confused as signal. DDID and BM3D-SAPCA retain the wavy patterns in the noise, while BM3D smoothes them away. The latter should be considered a mistake as on other occasions the pattern could indeed have been a valid signal. Recall that our method is much simpler than BM3D and especially BM3D-SAPCA, which extends BM3D by additional shape-adaptive and PCA steps.

In MATLAB, denosing a grayscale image with 256×256 pixels and a window size of 31×31 using a single core of a Intel Xeon 2.67 GHz takes 70 seconds. The bottleneck is the transition from spatial to frequency domain. If this transition was a pure Gabor transform, we could exploit sliding window techniques [17] to update the Fourier coefficients incrementally. However, since the signal is multiplied by an arbitrary range kernel, we need a per-pixel FFT with complexity $O(N^2 \log N)$. Thus, we implemented a C version using the FFTW library, which shortened the time to 40 seconds. Since the pixels are mutually independent, we achieved linear scalability using dual quad-core CPUs, reducing the time to 5 seconds. Finally,

GRAYSCALE	DDID	BM3D	COLOR	DDID	BM3D
Barbara	**30.80**	30.72	Baboon	**26.17**	25.95
Boats	29.79	**29.91**	F-16	**32.88**	32.78
Cameraman	**29.46**	29.45	House	32.69	**33.03**
Couple	29.56	**29.72**	Kodak 1	29.09	**29.13**
Finger Print	27.32	**27.70**	Kodak 2	32.29	**32.44**
Hill	29.71	**29.85**	Kodak 3	**34.55**	34.54
House	32.66	**32.86**	Kodak 12	33.46	**33.76**
Lena	**32.14**	32.08	Lake[*]	**28.85**	28.68
Man	29.62	29.62	Lena	**32.30**	32.27
Montage	**32.61**	32.37	Pepper	**31.25**	31.20
Pepper	**30.26**	30.16	Tiffany[*]	**32.49**	32.23

Table 1: *PSNR (dB) comparison between DDID and BM3D for noise with standard deviation $\sigma = 25$. Better results are in bold.*

our GPU implementation on an NVIDIA GeForce GTX 470 cut the time down to one second.

5. CONCLUSIONS

We have presented an image denoising method that is simple to implement, yet achieves competitive results with sophisticated state-of-the-art methods. Our method denoises 1D and 2D signals effectively and we expect straightforward extension to higher dimensions. Furthermore, our algorithm exhibits interesting properties, notably a hierarchy of signals, parallels between the guide and noisy images, and symmetry in the spatial and frequency domains. We believe that by studying simpler methods, we can gain new intuition to make progress. The simplicity of our method is an invitation to further exploration, ultimately leading to better understanding of image denoising.
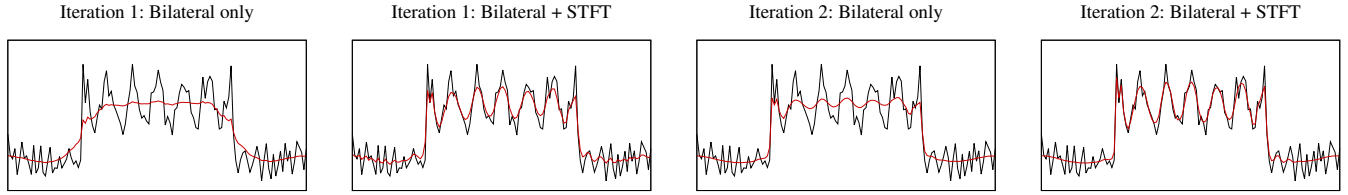
---

[*]*Lake* and *Tiffany* are not part of the BM3D test images. They are taken from the SIPI image database like most of the BM3D test images.

[+]Thanks to Alessandro Foi for allowing us to include the *Montage* image.

| Original | Noisy Image (20.17 dB) |
| **DDID** (32.61 dB) | BM3D (32.37 dB) |
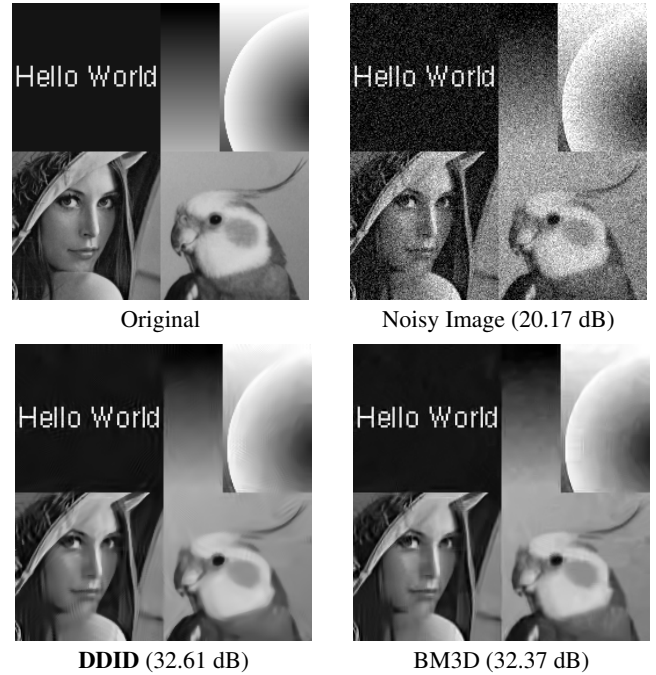
**Fig. 1**: *DDID has less low-frequency noise than BM3D.*

Fig. 2: *Intermediate steps of first two iterations of denoising. The bilateral filter and STFT shrinkage cooperate in alternation to denoise the signal. Denoised results in red are compared against the initial noisy signal in black.*
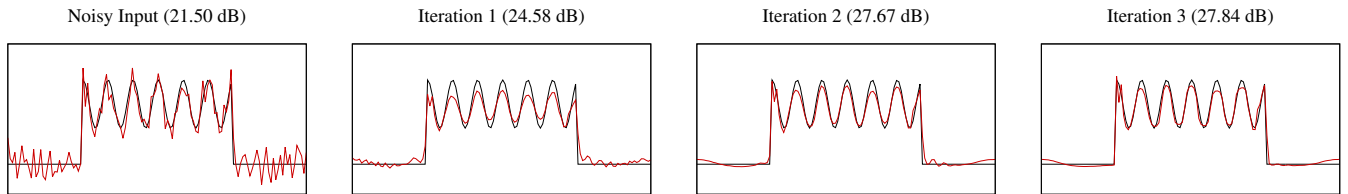
Fig. 3: *Progression of denoising. The guide signal in red improves every iteration, approximating the original signal in black.*
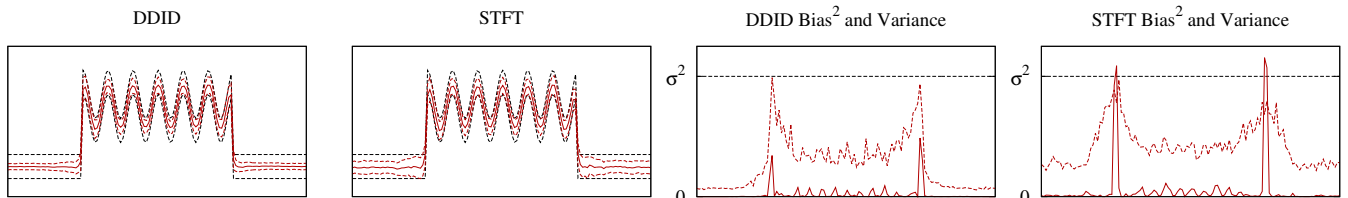
Fig. 4: *Comparison of DDID against STFT with Wiener filter, averaged over 200 signals. STFT has residual noise due to ringing artifacts, while DDID benefits from using the bilateral mask. Left: red solid and dashed lines denote the expected value with a confidence interval of one standard deviation. Right: red solid and dashed lines are the squared bias and variance. Black dashed lines are the initial noise variance.*
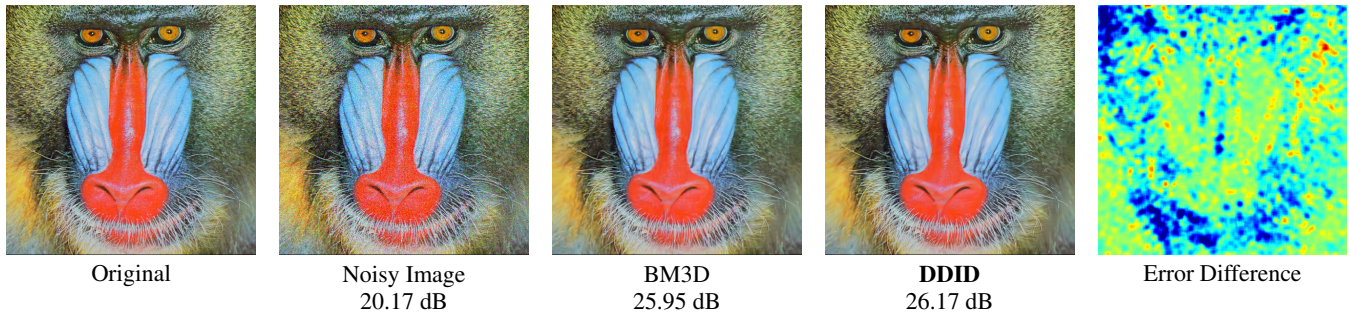
Fig. 5: *Comparison of DDID and BM3D. DDID effectively denoises hair-like structures, while BM3D is stronger for edge-like structures. Blue regions mark where DDID has lower error than BM3D, yellow and red regions mark the opposite, and green regions mark similar errors.*

Fig. 6: *Artifact comparison of DDID, BM3D, and BM3D-SAPCA. Misclassification of noise as signal due to accidently regular patterns is acceptable and is to be expected. BM3D produces a smooth result, but this would fail in other occasions where the pattern would be a signal.*

6. REFERENCES

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998, pp. 839–846.

[2] A. Buades, B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

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

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "A nonlocal and shape-adaptive transform-domain collaborative filtering," in *Proc. 2008 Int. Workshop on Local and Non-Local Approximation in Image Processing, LNLA 2008*, 2008.

[5] K. Dabov, R. 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 (SPARS'09)*, 2009.

[6] M. Lebrun, "An Analysis and Implementation of the BM3D Image Denoising Method," *Image Processing On Line*, vol. 2012, 2012.

[7] P. Chatterjee and P. Milanfar, "Is denoising dead?," *Image Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 895–911, 2010.

[8] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *CVPR*, 2011, pp. 2833–2840.

[9] A. Levin, B. Nadler, F. Durand, and W.T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *ECCV (5)*, 2012, pp. 73–86.

[10] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 25, no. 3, pp. 235 – 238, jun 1977.

[11] R. Fattal, M. Agrawala, and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 51, 2007.

[12] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, July 2002.

[13] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," in *ACM Trans. Graph.* ACM, 2004, vol. 23, pp. 664–672.

[14] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct as an overcomplete denoising tool," in *Proc. 2005 Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2005*, 2005, pp. 164–170.

[15] G. Yu and G. Sapiro, "DCT image denoising: a simple and effective image denoising algorithm," *Image Processing On Line*, vol. 2011, 2011.

[16] H. Yu, L. Zhao, and H. Wang, "Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain," *IEEE Trans. on Image Process.*, vol. 18, no. 10, pp. 2364 –2369, oct. 2009.

[17] S. Albrecht, I. Cumming, and J. Dudas, "The momentary fourier transformation derived from recursive matrix transformations," in *Digital Signal Processing Proceedings, 1997. DSP 97., 1997 13th International Conference on.* IEEE, 1997, vol. 1, pp. 337–340.