NON-LOCAL DUAL IMAGE DENOISING

N. Pierazzo*, *M. Lebrun**, *M.E. Rais**†, *J.M. Morel**, *and G. Facciolo**

* CMLA, École Normale Supérieure de Cachan, France
† Dpt. Matemàtiques i Informàtica, UIB, Spain

ABSTRACT

The current state-of-the-art non-local algorithms for image denoising have the tendency to remove many low contrast details. Frequency-based algorithms keep these details, but on the other hand many artifacts are introduced. Recently, the Dual Domain Image Denoising (DDID) method has been proposed to address this issue. While beating the state-of-the-art, this algorithm still causes strong frequency domain artifacts. This paper reviews DDID under a different light, allowing to understand their origin. The analysis leads to the development of NLDD, a new denoising algorithm that outperforms DDID, BM3D and other state-of-the-art algorithms. NLDD is also three times faster than DDID and easily parallelizable.

Index Terms— Image denoising, Patch-Based methods, Fourier shrinkage, Dual Denoising, Non-Local Bayes

1. INTRODUCTION

Image denoising is one of the fundamental image processing challenges [1]. Image denoising methods can be divided into two main categories: frequency-based or spatial-based.

The frequency domain methods rely on an underlying image regularity assumption and work by compressing/thresholding coefficients in some frequency domain [2, 3, 4, 5]. The Wiener filter [6] is one of the first such methods. Donoho et al. [7] expand it to the wavelet domain.

Among the spatial-based methods, non-linear variational methods such as Total Variation Minimization (Rudin et al. [8, 9]) were once the state-of-the-art. Nowadays, spatial-based methods achieve remarkable results by exploiting spatial self-similarity in the image itself. Non-Local Means (NL-Means) (Buades et al. [10, 11]) and UINTA (Awate et al. [12]) are among the first methods of this kind. They denoise by averaging similar patches in the image. Patch-based denoising methods have developed into attempts to model the patch space of an image, or of a set of images. Recently, algorithms proposing sparse representations of patches using dictionaries were introduced by Elad et al. [13], Mairal et al. [14, 15, 16] and Yu et al. [17]. Modeling image properties using a Gaussian Scale Mixture (GSM) model is the basic idea of a denoising algorithm proposed by Portilla et al. [18]. Rajaei recently proposed an improvement of the BLS-GSM method [19]. In 2011, Levin and Nadler [20] tried to model the patch space using a non-parametric approach by sampling from a huge database of patches. This method was later accelerated by Pierazzo and Rais [21].

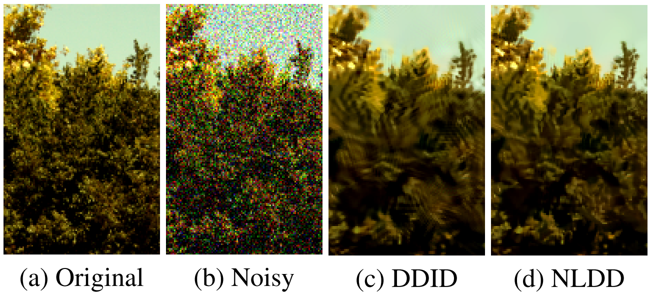
(a) Original (b) Noisy (c) DDID (d) NLDD

Fig. 1. A detail of the artifacts produced by DDID and the corresponding result of NLDD. In this example $\sigma = 30$.

A current state-of-the-art method that takes advantage of both space and frequency domain approaches is BM3D (Dabov et al. [22, 23]), which is one of the most efficient patch-based denoising methods to date. Finally NL-Bayes (Lebrun et al. [24]) is a spatial-based method that improves NL-means by considering a Gaussian probability model for each set of similar patches. Contrary to BM3D, NL-Bayes does not produce artifacts.

In 2013 a new hybrid method called *Dual-Domain Image Denoising* (DDID) was published by Knaus and Zwicker [25]. It is remarkably simple to implement, and it provides results that are generally superior in terms of PSNR to state-of-the-art methods such as BM3D and NL-Bayes. Its main drawback is that it produces typical frequency domain artifacts, as shown in Fig. 1. This is unexpected, since the method itself was developed to avoid the artifacts of frequency-based methods. The present article explains the creation of those artifacts and presents Non-Local Dual Denoising (NLDD), a faster and better performing denoising algorithm.

The organization of this paper is as follows. In section

Aknowledgements: work partially supported by Centre National d'Etudes Spatiales (MISS Project), European Research Council (Advanced Grant Twelve Labours), Office of Naval Research (under Grant N00014-97-1-0839), Direction Générale de l'Armement, Fondation Mathématique Jacques Hadamard and Agence Nationale de la Recherche (Stereo project).

978-1-4799-5751-4/14/$31.00 ©2014 IEEE

2 the Dual-Domain Image Denoising algorithm is revisited from a different perspective that gives insight about the artifacts. In section 3 NLDD is presented, addressing DDID's problems, and its results are shown in section 4.

2. DUAL DOMAIN IMAGE DENOISING

This section presents an alternative interpretation of DDID that differs from the one originally proposed by Knaus and Zwicker [25]. The original description of DDID splits the image into a low- and a high-contrast layer, which are treated respectively with a spatial and a frequency domain method. In this work instead, the spatial domain filtering is seen as a pre-processing to improve the frequency domain denoising.

DDID consists of three almost identical steps. The output of each step is used to guide the following one. Each step of the algorithm processes the noisy image y pixel-wise using the guide image g. Each pixel p is denoised using the $d \times d$ neighborhood ($d = 31$) of both the noisy and the guide image.

Denoising in the frequency domain often results in the appearance of artifacts. To prevent it each patch is pre-processed to eliminate discontinuities corresponding to object's edges and patch's boundaries. To that end, a kernel k is created from g identifying the pixels belonging to the same object as its central pixel p. This kernel is the product of a *spatial* and *range* kernels, as used in the bilateral filter [26, 27]

$$k(q) = k_s(q) \cdot k_r(q). \tag{1}$$

- The *range* kernel is used to identify the pixels belonging to the same object. The idea is that, in g, pixels belonging to the same object as the central pixel will have similar values. The kernel is

$$k_r(q) = \exp\left(-\frac{|g(q) - g(p)|^2}{\gamma_r \sigma^2}\right), \tag{2}$$

  where γ_r is a parameter of the algorithm and σ is the standard deviation of the noise.

- The *spatial* kernel, identifies the pixels close to the central one and smooths periodization discontinuities associated to the frequency domain processing. To achieve that a Gaussian kernel of standard deviation σ_s is used, where σ_s is a parameter of the algorithm:

$$k_s(q) = \exp\left(-\frac{|q - p|^2}{2\sigma_s^2}\right). \tag{3}$$

Since denoising with Fourier coefficients has problems in presence of edges (due to the Gibbs phenomenon), the goal is to make the parts of the patch not relevant to the denoising as *regular* as possible. k is used to compute the average of the "relevant" part of both the noisy and the guide patches:

$$\tilde{s} = \frac{\sum k(q)y(q)}{\sum k(q)}, \qquad \tilde{g} = \frac{\sum k(q)g(q)}{\sum k(q)}, \tag{4}$$

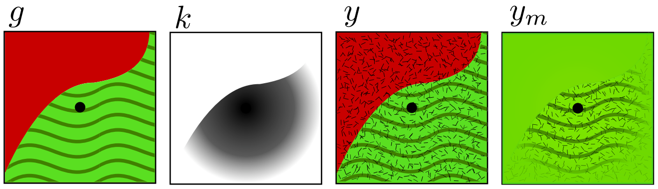

Fig. 2. Illustration of DDID's preprocessing of a patch. The kernel k is computed using the guide g. In the modified patch y_m all object discontinuities have been removed, leaving only the texture information corresponding to the object selected by the kernel k. The removed pixels are replaced by $\tilde{s}$: the average of the *meaningful* portion of the patch.

where the sums are computed over $\mathcal{N}_p$, the domain of the $d \times d$ patch centered at p. After that, the parts of the patch not taken into account by k are set to the respective average. The resulting *modified* patch is

$$y_m(q) = k(q)y(q) + (1 - k(q))\tilde{s}. \tag{5}$$

As illustrated in Fig. 2 the patch y_m is similar to y in the parts belonging to the same object as the central pixel (including the noise) and smooth in the rest. The same procedure is applied to the guide patch

$$g_m(q) = k(q)g(q) + (1 - k(q))\tilde{g}. \tag{6}$$

At this point, y_m and g_m are two patches, built in the same way, in which discontinuities have been strongly reduced and only information "relevant" to denoise the central pixel has been kept. It is therefore safe to apply the Fourier transform and to continue the process in the frequency domain

$$G(f) = \sum_{q \in \mathcal{N}_p} \exp\left(-\frac{2i\pi(q-p)f}{d}\right) g_m(q), \tag{7}$$

$$S(f) = \sum_{q \in \mathcal{N}_p} \exp\left(-\frac{2i\pi(q-p)f}{d}\right) y_m(q). \tag{8}$$

Assuming that y contains an additive white Gaussian noise of variance σ^2, the amount of noise present in y_m depends on k. In particular, for a pixel q, $y_m(q)$ contains a noise equal to $\sigma^2 k(q)$. An interesting property of the Fourier transform is that the noise in every pixel is evenly distributed over all frequencies. Thus every frequency of S has Gaussian noise with the same variance

$$\sigma_f^2 = \sigma^2 \sum_{q \in \mathcal{N}_p} k(q)^2. \tag{9}$$

The patch is then denoised by shrinking its Fourier coefficients $S(f)$ by the factor

$$K(f) = \begin{cases} 1 & \text{if } f = 0, \\ \exp\left(-\frac{\gamma_f \sigma_f^2}{|G(f)|^2}\right) & \text{otherwise,} \end{cases} \tag{10}$$

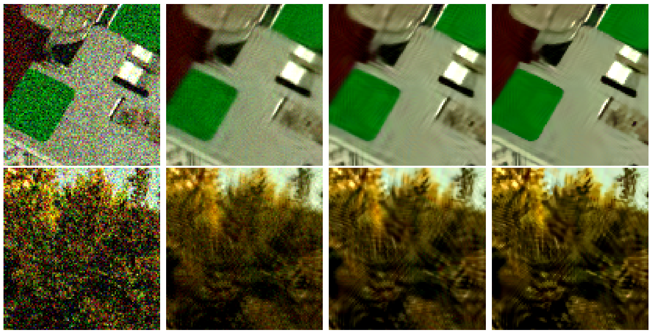

Fig. 3. Artifacts in DDID. From left to right: the noisy image (with $\sigma = 30$), the result of the first, second, and last iteration of the algorithm.

where γ_f is a parameter of the algorithm. The denoised value of the central pixel is finally recovered by reversing the Fourier transform. Inverting equation (5) is unnecessary, since $k(p) = 1$. Since the inverse Fourier transform evaluated in the center of the patch is the average of the frequencies, the central pixel's value is computed as

$$x(p) = \frac{1}{d^2} \sum_f S(f)K(f). \tag{11}$$

Equations (7-11) are slightly different from the ones presented in [25]. In fact, it can be easily proved that $G(f)$ and $S(f)$ differ from the ones presented in the original paper only at the zero frequency. This frequency is then restored after the shrinkage. In the presented version, the zero frequency is left untouched by the shrinkage, by imposing $K(0) = 1$.

For color images, the kernel k_r is computed by using the Euclidean distance in the color space, while the Fourier thresholding is done independently on each channel in the YUV color space.

Artifacts in DDID

The above description highlights that the denoising in DDID is accomplished in the frequency domain, while the spatial pre-processing is used to remove discontinuities from the image. The described procedure is applied three times with different parameters. Each time the result of the previous calculation is used as a guide, except in the first iteration where the noisy image itself is used. It is worth noting that the image is denoised in the last iteration only. The other two are only used to obtain a suitable guide.

Besides being slow to compute, the main drawback of DDID is that its results often present ringing artifacts (as seen in Fig. 1). This is surprising since removing the strong edges, as in equation (5), should prevent it. Since the guide image used in the first iteration is noisy, and the kernel in (1) is computed from it, "parasite" information is retained and propagated in the following iterations (see Fig. 3). This yields a result that contains artifacts.

3. NON-LOCAL DUAL DENOISING

Since, as concluded in the previous section, most of the artifacts of DDID come from the guide image, a method to avoid them is to feed the algorithm with a cleaner image.

Non-Local Dual Denoising uses the Non-Local Bayes [24] denoising algorithm to provide a clean guide, and then applies the last step of DDID to denoise the image (with parameters $\sigma_s = 7$, $\gamma_r = 0.7$ and $\gamma_f = 0.8$). NL-Bayes has been chosen over other state-of-the-art algorithms (such as BM3D) because it generally provides a smoother output (see [28]). BM3D has been tested too as the guide. However the results, while still improving the ones of both BM3D and DDID, were slightly worse than the ones of NLDD. A pseudo-code for NLDD is listed in Algorithm 1.

Algorithm 1 Non-Local Dual Denoising

```
function NLDD($\mathbf{y}$, $\sigma$)
    $\mathbf{g} \leftarrow$ NL-BAYES($\mathbf{y}$, $\sigma$)
    for all pixels $p \in \mathbf{y}$ do
        $y \leftarrow$ EXTRACTPATCH($\mathbf{y}$, $p$)
        $g \leftarrow$ EXTRACTPATCH($\mathbf{g}$, $p$)
        $k \leftarrow$ COMPUTEK($\mathbf{g}$, $p$)                ▷ Eq. 1
        $y_m, g_m \leftarrow$ MODIFYPATCHES($y$, $g$, $k$)      ▷ Eq. 4-6
        $S \leftarrow$ FFT($y_m$)
        $G \leftarrow$ FFT($g_m$)
        $\mathbf{x}(p) \leftarrow$ SHRINK($S$, $G$, $k$, $\sigma$)          ▷ Eq. 10-11
    end for
    return $\mathbf{x}$
end function
```

This algorithm has several advantages over DDID. Since the guide image (provided by NL-Bayes) has less artifacts than the one computed in the first two iterations of DDID, it generally provides better results, as shown in section 4. As expected, the results contain less artifacts. In addition, NLDD is faster than DDID as only one iteration is needed. Moreover, since both DDID and NL-Bayes are heavily parallelizable, NLDD could also be implemented on a GPU architecture [25]. Our multi-thread C++ implementation of DDID takes 69 seconds to denoise a 704×469 color image on a 8-core Intel Xeon 2.33 GHz. On the same machine, NLDD takes just 22 seconds, about one third of the time.[1]

4. EXPERIMENTAL RESULT

NLDD has been compared against DDID, BM3D and NL-Bayes with different amounts of noise. For the tests an heterogeneous set of noise-free images was used. All the results are evaluated using the *Peak Signal-to-Noise Ratio* (PSNR) and SSIM [29], which isan alternative metric conceived to simulate the response of the Human Visual System.

[1] The C++ code for NLDD along with a MATLAB wrapper and an online demo is available in the supplementary materials webpage [28].

978-1-4799-5751-4/14/$31.00 ©2014 IEEE

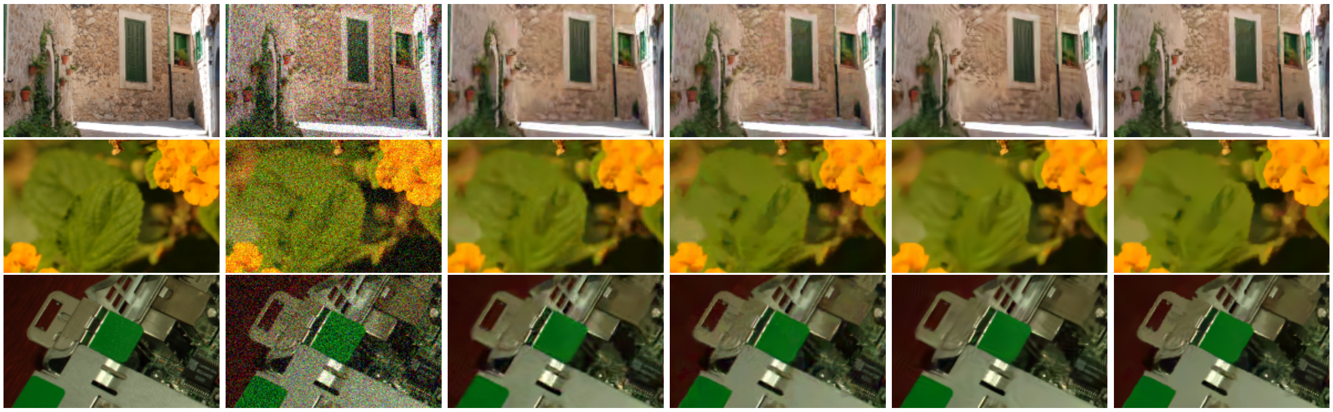

Fig. 4. Crops from the results of the images Alley, Flowers and Computer. From left to right: the original image, the noisy image ($\sigma = 30$), the outputs of BM3D, NL-Bayes, DDID, and NLDD. Full results are available in the article's website.

Image	BM3D	NL-Bayes	DDID	NLDD
Alley	29.32	29.12	29.33	**29.41**
Computer	30.66	30.68	31.00	**31.10**
Dice	38.02	37.97	38.45	**38.78**
Flowers	33.76	33.85	34.36	**34.48**
Girl	36.95	36.62	37.26	**37.33**
Traffic	28.83	29.00	29.20	**29.40**
Trees	24.62	25.02	24.85	**25.09**
Valldemossa	27.24	27.37	27.27	**27.48**
Mean	31.18	31.20	31.46	**31.64**

Table 1. Values of PSNR for $\sigma = 30$.

The results for $\sigma = 30$, where DDID performs best, are summarized in Table 1. NLDD outperforms the other algorithms in terms of PSNR. The same holds for the SSIM comparison, which is available online, along with the complete database of results [28]. The results for other levels of noise are summarized in Table 2. NLDD provides the best results for values of σ between 20 and 60. These coincide with the values of noise for which DDID has the best performance. Looking closely at Fig. 4 fewer artifacts can be noticed for NLDD. However, the values of SSIM don't reflect the magnitude of this improvement, but the details in Fig. 1 suggest that the quality of the two reconstructions is significantly different.

It is worth noticing that when the guide image is inaccurate NLDD also performs relatively poorly. For example in the image "Flowers" NL-Bayes fails to recover the texture of the leaves. As a result, these areas of the image are not fully recovered by NLDD.

PSNR

σ	BM3D	NL-Bayes	DDID	NLDD
10	36.84	**37.07**	36.92	36.92
20	33.22	33.42	33.52	**33.64**
30	31.18	31.20	31.46	**31.64**
40	29.70	29.62	29.99	**30.21**
60	27.45	28.35	27.96	**28.38**
80	26.53	**27.03**	26.50	26.89

SSIM

σ	BM3D	NL-Bayes	DDID	NLDD
10	0.9684	0.9680	**0.9699**	0.9691
20	0.9303	0.9308	0.9331	**0.9346**
30	0.8938	0.8928	0.8957	**0.8995**
40	0.8614	0.8572	0.8600	**0.8671**
60	0.8064	0.8084	0.7994	**0.8100**
80	0.7592	**0.7595**	0.7500	0.7570

Table 2. Average values of PSNR and SSIM with different noise levels.

5. CONCLUSION

In this paper the DDID denoising algorithm has been reviewed under a different light, allowing to understand the origin of most of its artifacts. This analysis has led to the development of NLDD, a new denoising algorithm that addresses the creation of these artifacts and outperforms DDID and other state-of-the-art algorithms in almost every test.

The results clearly show that NLDD is superior to the other methods in both PSNR and SSIM. However, a close inspection of the results leads to the conclusion that a robust metric for evaluating denoising algorithms is still needed. Indeed it is observed that even modern metrics such as SSIM, commonly applied in these cases, fall short in presence of isolated but blunt artifacts. Nonetheless the qualitative analysis of the results confirms the superiority of NLDD.

By construction the NLDD method is faster than DDID, but it is still slow compared to other methods. However, like DDID and NL-Bayes themselves it is heavily parallelizable and can be implemented in GPU hardware.

6. REFERENCES

[1] M.C. Motwani, M.C. Gadiya, R.C. Motwani, and F.C. Harris Jr., "Survey of image denoising techniques," in *Proceedings of GSPX*, 2004.

[2] J. L. Starck, E. J. Candès, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE TIP*, vol. 11, no. 6, 2002.

[3] H.Q. Li, S.Q. Wang, and C.Z. Deng, "New image denoising method based wavelet and curvelet transform," in *WASE ICIE*, 2009, vol. 1.

[4] D. Gnanadurai and V. Sadasivam, "Image denoising using double density wavelet transform based adaptive thresholding technique," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 03, no. 01, 2005.

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

[6] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, The MIT Press, 1964.

[7] D.L. Donoho and J.M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, 1994.

[8] L.I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, 1992.

[9] P. Getreuer, "Rudin-Osher-Fatemi total variation denoising using split Bregman," *Image Processing On Line*, vol. 2012, 2012.

[10] A. Buades, B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," *SIAM Mult. Model. Simul.*, vol. 4, no. 2, 2006.

[11] A. Buades, B. Coll, and J.M. Morel, "Non-local means denoising," *Image Processing On Line*, 2011.

[12] S.P. Awate and R.T. Whitaker, "Unsupervised, Information-Theoretic, adaptive image filtering for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, 2006.

[13] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE TIP*, vol. 15, no. 12, 2006.

[14] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE TIP*, vol. 17, no. 1, 2008.

[15] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *SIAM Mult. Model. Simul.*, vol. 7, no. 1, 2008.

[16] M. Lebrun and A. Leclaire, "An implementation and detailed analysis of the K-SVD image denoising algorithm," *Image Processing On Line*, 2012.

[17] G. Yu, G. Sapiro, and S. Mallat, "Image modeling and enhancement via structured sparse model selection," in *IEEE ICIP*, 2010.

[18] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE TIP*, 2003.

[19] B. Rajaei, "An analysis and improvement of the bls-gsm denoising method," *Image Processing On Line*, 2013.

[20] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *IEEE CVPR*, 2011.

[21] N. Pierazzo and M. Rais, "Boosting shotgun denoising by patch normalization," in *IEEE ICIP*, 2013.

[22] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3d transform-domain collaborative filtering," *IEEE TIP*, vol. 16, no. 82, 2007.

[23] M. Lebrun, "An analysis and implementation of the bm3d image denoising method," *Image Processing On Line*, 2012.

[24] M. Lebrun, A. Buades, and J.M. Morel, "Implementation of the "non-local bayes" (nl-bayes) image denoising algorithm," *Image Processing On Line*, 2013.

[25] C. Knaus and M. Zwicker, "Dual-domain image denoising," *IEEE ICIP*, 2013.

[26] L.P. Yaroslavsky, "Local adaptive image restoration and enhancement with the use of DFT and DCT in a running window," in *Proceedings of SPIE*, 1996.

[27] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *IEEE ICCV*, 1998.

[28] "NLDD supplementary material," `http://dev.ipol.im/~pierazzo/nldd`, 2014.

[29] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, 2004.