

DA3D: Fast and Data Adaptive Dual Domain Denoising

Nicola Pierazzo, Martin Rais, Jean-Michel Morel, Gabriele Facciolo

▶ **To cite this version:**

Nicola Pierazzo, Martin Rais, Jean-Michel Morel, Gabriele Facciolo. DA3D: Fast and Data Adaptive Dual Domain Denoising. ICIP, 2015, Québec, Canada. hal-01240841

HAL Id: hal-01240841
https://enpc.hal.science/hal-01240841

Submitted on 10 Dec 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DA3D: FAST AND DATA ADAPTIVE DUAL DOMAIN DENOISING

N. Pierazzo[⋆]*, M.E. Rais*[⋆†]*, J.M. Morel*[⋆]*, and G. Facciolo*[⋆‡]

[⋆] CMLA, École Normale Supérieure de Cachan, France
[†] Dpt. Matemàtiques i Informàtica, UIB, Spain
[‡] IMAGINE/LIGM, École Nationale des Ponts et Chaussées, France

ABSTRACT

This paper presents DA3D (Data Adaptive Dual Domain Denoising), a "last step denoising" method that takes as input a noisy image and as a guide the result of any state-of-the-art denoising algorithm. The method performs frequency domain shrinkage on shape and data-adaptive patches. Unlike other dual denoising methods, DA3D doesn't process all the image samples, which allows it to use large patches (64×64 pixels). The shape and data-adaptive patches are dynamically selected, effectively concentrating the computations on areas with more details, thus accelerating the process considerably. DA3D also reduces the staircasing artifacts sometimes present in smooth parts of the guide images.The effectiveness of DA3D is confirmed by extensive experimentation. DA3D improves the result of almost all state-of-the-art methods, and this improvement requires little additional computation time.

***Index Terms*—** Image denoising, Patch-Based methods, Fourier shrinkage, Dual Denoising, Data Adaptive

1. INTRODUCTION

Image denoising is one of the fundamental image restoration challenges [1]. It consists in estimating an unknown noiseless image $\mathbf{y}$ from a noisy observation $\mathbf{x}$. We consider the classic image degradation model

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

where the observation $\mathbf{x}$ is contaminated by an additive white Gaussian noise $\mathbf{n}$ of variance σ^2.

All denoising methods assume some underlying image regularity. Depending on this assumption they can be divided, among others, into transform-domain and spatial-domain methods.

Transform domain methods work by shrinking (or thresholding) the coefficients of some transform domain [2, 3, 4]. The Wiener filter [5] is one of the first such methods operating on the Fourier transform. Donoho et al. [6] extended it to the wavelet domain.

Space-domain methods traditionally use a local notion of regularity with edge-preserving algorithms such as total variation [7], anisotropic diffusion [8], or the bilateral filter [9].

Nowadays however spatial-domain methods achieve remarkable results by exploiting the self-similarities of the image [10]. These patch-based methods are non-local as they denoise by averaging similar patches in the image. Patch-based denoising has developed into attempts to model the patch space of an image, or of a set of images. These techniques model the patch as sparse representations on dictionaries [11, 12, 13, 14, 15], using Gaussian Scale Mixtures models [16, 17], or with non-parametric approaches by sampling from a huge database of patches [18, 19, 20, 21].

Current state-of-the-art denoising methods such as BM3D [22] and NL-Bayes [23] take advantage of both space- and transform-domain approaches. They group similar image patches and jointly denoise them by collaborative filtering on a transformed domain. In addition, they proceed by applying two slightly different denoising stages, the second stage using the output of the first one as its guide.

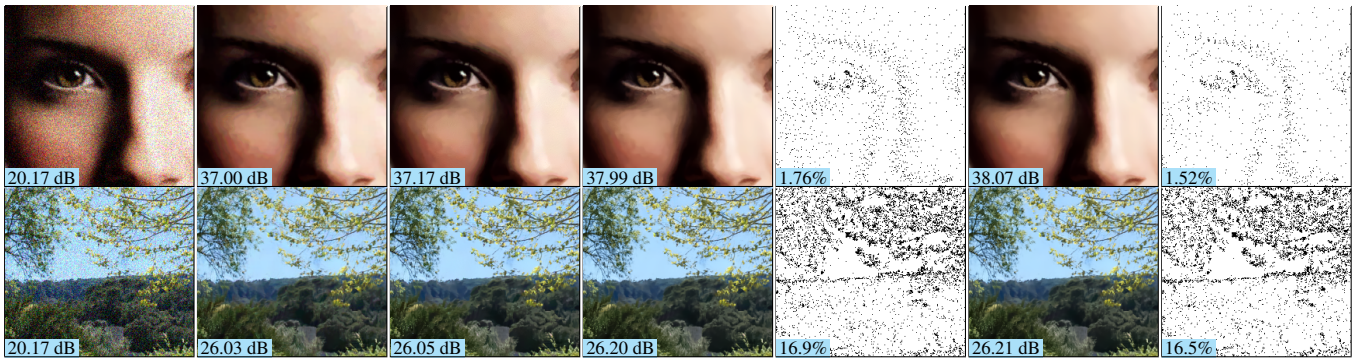
Some recently proposed methods use the result of a different algorithm as their guide for a new denoising step. Combining for instance, nonlocal principles with spectral decomposition [24], or BM3D with neural networks [25]. This allows one to mix different denoising principles, thus yielding high quality results [24, 25]. DDID [26] is an iterative algorithm that uses a guide image (from a previous iteration) to determine spatially uniform regions to which Fourier shrinkage could be applied without introducing ringing artifacts. Several methods [27, 28, 29] use a single step of DDID with a guide image produced by a different algorithm. This yields much better results than the original DDID. The reason for their success is the use of large (31×31) and shape-adaptive patches. Indeed, the Fourier shrinkage works better on large stationary blocks.

Unfortunately, DDID has a prohibitive computational cost, as it paradoxically denoises a large patch to recover a single pixel. Moreover, contrary to other methods, aggregation of these patches doesn't improve the results since it introduces blur. We show in this paper that these two problems can be solved by introducing a new patch selection, accompanied by a weighted aggregation strategy.

Contribution. This paper presents DA3D (Data Adaptive Dual Domain Denoising), a new "last step" denoising method that performs frequency domain shrinkage on shape-adaptive and data-adaptive patches. DA3D consistently improves the results of state-of-the-art methods such as BM3D or NL-Bayes with little additional computation time.

Similarly to DDID [26], DA3D uses a guide image to extract the shape-adaptive patches. But unlike it, DA3D processes only a fraction of the patches, which are aggregated instead of just taking their central pixel. The patches are dynamically selected, saving computations on uniform areas of the image to use them on areas with more details. This accelerates considerably the process, allowing to use even larger (64×64) patches, which in turn yields improved results. Some shape adaptive methods [30, 31] collectively denoise groups of similar small patches (8×8 blocks for BM3D-SAPCA [30]). DA3D does not consider groups, instead it uses much larger patches to extract more information from the underlying structure.

Aknowledgements: work partly founded by Centre National d'Etudes Spatiales (CNES, MISS Project), European Research Council (advanced grant Twelve Labours), Office of Naval research (ONR grant N00014-14-1-0023), DGA Stéréo project, ANR-DGA (project ANR-12-ASTR-0035), FUI (project Plein Phare) and Institut Universitaire de France.



(a) Noisy (b) Guide: NL-Bayes (c) NL-Bayes + DDID step (d) NL-Bayes + Sparse DDID (e) Sparse DDID samples (f) NL-Bayes + DA3D (g) DA3D samples

Fig. 1: Results of applying different post-processing steps on the same image (with noise $\sigma = 25$). The guide was produced with Non-Local Bayes. Figs. (e) and (g) show the centers of the blocks used for denoising (d) and (f) respectively (with $\tau = 2$). Image (c) was generated with 31×31 blocks, while for (d) and (f) 64×64 blocks were used. Notice in (d) and (f) the difference on che cheek and the forehead of the girl.

Our second contribution in DA3D further improves the quality of the results by adapting the processing to the underlying data. The apparition of the staircasing is well known for non-local methods [32]. To mitigate the influence of such artifacts present in the guide image, we use a first order non-linear local kernel regression [33, 34] to estimate, for each patch, an affine approximation coherent with the data within the patch. The denoising is then performed with respect to this approximation. This data-adaptive approach is another innovation enabled by the use of large patches, and it noticeably improves the quality of the results on smooth regions of the image.

Section 2 recalls the DDID postprocess. Sections 3 and 4 present the DA3D algorithm, first describing the proposed sparse aggregation, then the data-adaptive patches. The performance of DA3D is extensively validated in the experiments of section 5.

2. DUAL DOMAIN IMAGE DENOISING STEP

A DDID step is a single iteration of the DDID algorithm [26], but it can also be used as a last denoising step for other methods [29, 27]. This section, along with the pseudocode in Table 1, summarizes it.

To denoise a pixel p from the noisy image $\mathbf{y}$ the DDID step extracts a 31×31 pixel block around it (denoted y) and the corresponding block g from the guide image $\mathbf{g}$.

The blocks are processed to eliminate discontinuities that may cause artifacts in the subsequent frequency-domain denoising. To that end, the weight function k is derived from g. The weights identify the pixels of the block belonging to the same object as the center p. This weight function has the form of the bilateral filter [35, 9]

$$k(q) = \exp\left(-\frac{|g(q) - g(p)|^2}{\gamma_r \sigma^2}\right) \exp\left(-\frac{|q-p|^2}{2\sigma_s^2}\right). \quad (2)$$

The first term identifies the pixels belonging to the same structure as p, by selecting the ones with a similar color in the guide, while the second term removes the periodization discontinuities associated with the Fourier transform. The parameters σ_s and γ_r are specific of the algorithm, and σ is the standard deviation of the noise.

The weights k are then used to modify y and g in order to remove their discontinuities yielding y_m and g_m (see lines 13-14 of Table 1). In this way the "relevant" part of the blocks (similar to the central pixel) is retained by k, and its average value is assigned to the rest.

The modified block y_m is denoised by shrinkage of its Fourier coefficients using g_m as an oracle (lines 15-18 of Table 1). Since discontinuities have been removed from the blocks, filtering in the Fourier domain doesn't introduce ringing, which is a major advance made by DDID in transform thresholding methods. The value of γ_f is a parameter of the algorithm. This shrinkage assumes that the image y contains additive white Gaussian noise.

Finally, the denoised value of the central pixel is recovered by reversing the Fourier transform. This process is repeated for every pixel of the image. For more details about DDID refer to [26, 29].

For color images, k is computed by using the Euclidean distance, while the shrinkage is done independently on each channel of the YUV color space. An example of the result is shown in Fig. 1c.

3. SPARSE DDID STEP

The DDID step explained in section 2 is slow because it has to process a block for every pixel. In fact, each pixel is denoised several times, but the result is discarded every time that it is not in the center of the current block. Since that the denoising remains valid for all pixels in the "relevant" part of the block, we propose to aggregate the processed blocks to form the final result. As a result, not every block needs to be processed, thus accelerating the algorithm considerably. Note that since the processing is done with the modified block, line 13 of Table 1 must be reverted to obtain the denoised block (line 20).

In our selection-aggregation process, the image is treated by color-coherent blocks and the results are aggregated with weights deduced from the guide image. This weighted average can also be seen as the interpolation of the denoised image from a subset of processed blocks [36, 37, 38]. We found that the best aggregation weights are the squares of the weights (2).

We now describe the greedy approach used for selecting the image blocks to be processed. At each iteration a weight map w with the sum of the aggregation weights is updated. This weight map permits to identify the pixel in the image with the lowest aggregation weight, which will be selected as the center of the next block to process (line 5 of Table 1). This process iterates until the total weight for each pixel becomes larger than a threshold τ. The weight function k is always equal to 1 in the center, so the algorithm always terminates. The procedure is detailed in Table 1. This variant is faster to execute than a single DDID step, since only a small number of blocks are actually processed. This allows bigger patches to be used, that in turn gives better results in terms of denoising quality. Experimentally, good results are achieved with patches as large

DDID	Sparse	DA3D		**Input:** $\mathbf{y}$ (noisy image), $\mathbf{g}$ (guide) **Output:** denoised image
-	•	•	1	$\mathbf{w} \leftarrow 0$
•	•	•	2	$\mathbf{out} \leftarrow 0$
•	-	-	3	**for all** pixels $p \in \mathbf{y}$ **do**
-	•	•	4	**while** $\min(\mathbf{w}) < \tau$ **do**
-	•	•	5	$p \leftarrow \arg\min(\mathbf{w})$
•	•	•	6	$y \leftarrow \text{EXTRACTPATCH}(\mathbf{y}, p)$
•	•	•	7	$g \leftarrow \text{EXTRACTPATCH}(\mathbf{g}, p)$
				// regression weight, eq. 4
-	-	•	8	$K_{reg} \leftarrow \text{COMPUTEKREG}(g)$
				// regression plane, eq. 3
-	-	•	9	$P \leftarrow \arg\min_P \sum [y(q) - P(q)]^2 \cdot K_{reg}(q)$
-	-	•	10	$y \leftarrow y - P$ // subtract plane from the block
-	-	•	11	$g \leftarrow g - P$ // and from guide
•	•	•	12	$k \leftarrow \text{COMPUTEK}(g)$ // eq. 2
•	•	•	13	$y_m \leftarrow k \cdot y + (1-k)\left(\frac{\sum k(l)y(l)}{\sum k(l)}\right)$
•	•	•	14	$g_m \leftarrow k \cdot g + (1-k)\left(\frac{\sum k(l)g(l)}{\sum k(l)}\right)$
•	•	•	15	$Y \leftarrow \text{DFT}(y_m)$
•	•	•	16	$G \leftarrow \text{DFT}(g_m)$
•	•	•	17	$\sigma_f^2 \leftarrow \sigma^2 \sum k(q)^2$
				// shrinkage
•	•	•	18	$K \leftarrow \begin{cases} 1 & \text{if } f = 0 \\ \exp\left(-\frac{\gamma_f \sigma_f^2}{\lvert G(f)\rvert^2}\right) & \text{otherwise} \end{cases}$
•	•	•	19	$x_m \leftarrow \text{IDFT}(K \cdot Y)$
				// revert line 13
-	•	•	20	$x \leftarrow \left[x_m - (1-k)\left(\frac{\sum k(l)y(l)}{\sum k(l)}\right)\right]/k$
-	-	•	21	$x \leftarrow x + P$ // add plane back to the block
-	•	•	22	$aggw \leftarrow k \cdot k$ // aggregation weight
•	-	-	23	$\mathbf{out}(p) \leftarrow \text{EXTRACTCENTRALPIXEL}(x_m)$
				// accumulate patch in the correct position
-	•	•	24	$\mathbf{w} \leftarrow \text{ADDPATCHAT}(p, \mathbf{w}, aggw)$
-	•	•	25	$\mathbf{out} \leftarrow \text{ADDPATCHAT}(p, \mathbf{out}, aggw \cdot x)$
•	-	-	26	**return out**
-	•	•	27	**return out**$/\mathbf{w}$

Table 1: Pseudo-code for DDID step, Sparse DDID and DA3D. The lines used only in the DDID step are highlighted in blue. Bullets show the operations of each algorithm. Variables in **bold** denote whole images, while *italics* denote single blocks. Multiplication and division are pixel-wise.

Method	$\sigma = 5$	$\sigma = 10$	$\sigma = 25$	$\sigma = 40$	$\sigma = 80$
BM3D	**38.43**	**34.94**	**30.58**	**28.27**	24.69
-	**38.39**	**34.95**	**30.67**	**28.41**	**25.09**
SAPCA [30]	-0.04	+0.01	+0.10	+0.14	+0.39
	38.24	34.70	30.37	27.99	**24.94**
BM3D [22]	**38.24**	**34.78**	**30.54**	**28.23**	**25.03**
	+0.00	+0.08	+0.16	+0.24	+0.09
	37.95	34.55	30.34	28.05	24.68
DDID [26]	37.90	34.52	30.39	28.16	24.84
	-0.04	-0.03	+0.05	+0.11	+0.16
	37.91	34.29	29.90	27.64	24.46
G-NLM [24]	35.67	33.97	29.77	27.49	24.10
	+0.60	+0.43	+0.58	+0.63	+0.65
	38.29	34.75	30.35	28.07	24.78
LSSC [13]	**38.34**	**34.88**	**30.61**	**28.32**	**24.97**
	+0.05	+0.13	+0.27	+0.25	+0.19
MLP		34.63	30.44		
+		**34.78**	**30.61**		
BM3D [25]		+0.15	+0.17		
	38.19	34.62	30.13	27.86	24.45
NLB [23]	38.20	34.72	30.38	28.14	24.78
	+0.02	+0.10	+0.25	+0.28	+0.33
	38.12	34.62	30.30	28.11	24.83
NLDD [29]	38.09	34.60	30.29	28.09	24.77
	-0.04	-0.02	-0.01	-0.02	-0.05
	37.31	33.58	28.97	26.50	22.72
NLM [10]	37.49	33.98	29.66	27.45	24.17
	+0.18	+0.40	+0.69	+0.95	+1.44
	37.97	34.56	30.38	28.18	**24.99**
PID [28]	37.90	34.48	30.28	28.08	24.81
	-0.07	-0.08	-0.10	-0.11	-0.18
	38.31	**34.78**	30.39	28.16	**25.00**
SAIST [15]	**38.31**	**34.82**	**30.53**	**28.27**	**25.07**
	-0.01	+0.04	+0.14	+0.11	+0.07
Average	+0.06	+0.11	+0.22	+0.27	+0.31

Table 2: Average PSNR comparison between state-of-the-art methods on grayscale images. The first line of each row shows the average PSNR. The second line shows the average PSNR of DA3D using the corresponding algorithm to generate the guide. The third line shows the average improvement due to DA3D. The best result for each noise level is shown in **bold**, and the ones within a range of 0.2 dB are shown in **gray**. MLP+BM3D only works with some specific levels of noise, the other levels are left blank.

as 64×64, which is to be contrasted to patch based methods using mostly 8×8 patches. An example of the result is shown in Fig. 1d. The total number of processed blocks depends on the image complexity. The centers of the effectively processed blocks are shown in Fig. 1e. They concentrate on edges and details.

4. DATA ADAPTIVE DUAL DOMAIN DENOISING

We now address a main drawback of the weight function (2), used for the bilateral filter and for many bilateral-inspired filters, including patch based methods. This weight function selects pixels of the block with a similar value. As a result, Sparse DDID works by processing parts of the image that are piecewise constant, considering the image as composed by many "flat" layers. This model is not well adapted for images that contain gradients or shadings, as the same smooth region may be split in many thin regions. The previous method can be extended to "normalize" each patch by subtracting an estimation of the gradient around the patch center. In practice, this means estimating an affine model of the block, as proposed in [33], which can be computed using a weighted least squares regression

$$\min_P \sum [y(q) - P(q)]^2 \cdot K_{reg}(q), \tag{3}$$

where the sum is computed over the domain of y and K_{reg} is a bilateral weight function

$$K_{reg}(q) = \exp\left(-\frac{|g(q) - g(p)|^2}{\gamma_{rr}\sigma^2} - \frac{|q-p|^2}{2\sigma_{sr}^2}\right), \tag{4}$$

which selects the parts of the block that gets approximated by P. To ensure that the central pixel gets denoised, the constraint $P(p) = g(p)$ is also added. Since the weights K_{reg} should capture the overall shape of the block, they are computed using a larger range parameter than the bilateral weight function in (2). It is worth noting

Image Size	G-NLM	SAPCA	NLB	PID	BM3D	SAIST	MLP	EPLL	NLDD	DDID	BM3D+DA3D
256×256	357 s	639 s	0.48 s	188 s	1.26 s	37.8 s	16.5 s	71.7 s	1.97 s	5.26 s	**2.92 s**
512×512	3359 s	2490 s	0.80 s	725 s	4.94 s	140 s	60.7 s	272 s	7.25 s	20.4 s	**9.62 s**

Table 3: Average running time depending on image size between grayscale denoising methods. The experiments were performed on a 8-core 2.67GHz Xeon CPU. Every algorithm was tested using its official implementation. For DA3D a Python implementation was used.

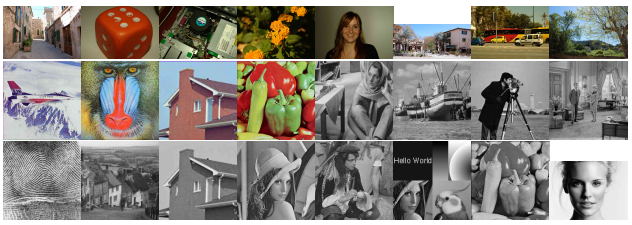
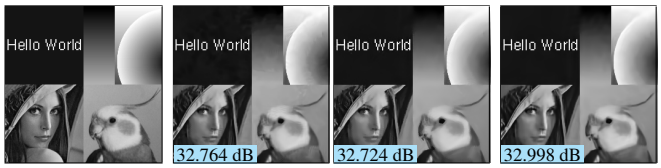
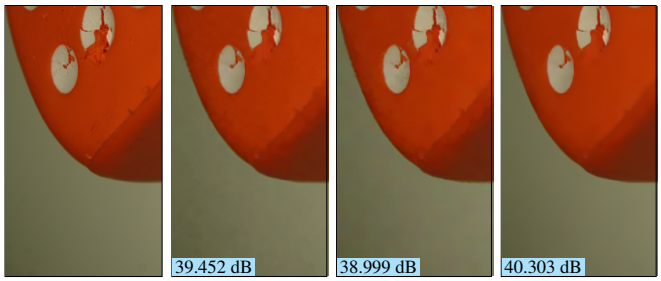


Fig. 2: Test images used in the experiments. No parameter learning or fitting was performed on this database.



(a) Original (b) BM3D-SAPCA (c) PID (d) BM3D+DA3D

Fig. 3: Denoising results for *Montage*, $\sigma = 25$. A zoom-in is advised to see the staircasing effects in (b) and (c) and their removal in (d).



(a) Original (b) BM3D (c) NL-Bayes (d) BM3D+DA3D

Fig. 4: Denoising results for *Dice*, $\sigma = 25$.

that although K_{reg} uses the guide to select the parts of the block in which to perform the estimation, the regression is performed directly on the noisy data y, thus allowing to correct any staircasing effect already present in the guide. The parameters σ_{sr} and γ_{rr} are specific of the algorithm. The case of color images is identical, since (3) is separable and can be computed independently on every channel.

Once estimated, the local plane P is subtracted from the patch, effectively removing shades and gradients. Then the standard DDID step is used to denoise the block and at the end the plane is added back. The whole procedure is detailed in Table 1 (lines 8-11, 21).

An example of the result is shown in Fig. 1f. Observe that the result present less staircasing effect and is better in terms of PSNR. In addition, less blocks are treated to denoise the gradients (Fig. 1g).

5. EXPERIMENTS

Our implementation of the DA3D algorithm (available in the support website [39]) has been tested against the set of images shown in Fig. 2. For γ_r and γ_f, the parameters of DDID [26] were kept ($\gamma_r = 0.7, \gamma_f = 0.8$), but since DA3D does not need to process all patches, the size of the patches themselves was chosen as 64×64, with $\sigma_s = 14$. The parameters $\tau = 2$, $\sigma_{sr} = 20$ and $\gamma_{rr} = 7$ were chosen experimentally on images outside the test database.

The DA3D method was applied to the results of several state-of-the-art algorithms. Each method was tested with noises of $\sigma = 5, 10, 25, 40, 80$. The results are summarized in Table 2.

DA3D improves the PSNR of every algorithm except NLDD and PID (which is to be expected, since they are based on a similar shrinkage strategy). This improvement is more marked with higher noises, which makes sense since the parameters of DDID were optimized for medium to high noise. It is worth mentioning that DA3D is even able to improve over BM3D-SAPCA, which is considered the best denoising algorithm up to date for grayscale images. Similar results are obtained using SSIM [40] as metric.

In general BM3D+DA3D (DA3D using BM3D as guide) offers one of the best performances with a reasonable computational cost. As an example the results of the best two performing methods for the image "Montage" are shown in Fig. 3, along with the result of BM3D+DA3D. The latter outperforms the other algorithms in terms of PSNR and image quality. Despite having a high PSNR value, the result of the other two algorithms present artifacts close to the edges and some staircasing (BM3D-SAPCA in particular).

The results for color images are similar to the grayscale case. As before, NLDD and PID are not improved (or just marginally improved) by DA3D. The detailed tables with PSNR and SSIM are available in the support website [39].

Fig. 4 shows the two best denoising results for the image "Dice", along with the result of BM3D+DA3D. Most of the artifacts generated by BM3D disappear with the post-processing, and at the same time the edges becomes sharper and the gradients smoother.

Running time. The time needed to run the analyzed algorithms is summarized in Table 3. Using DA3D as a post-processing method demands little additional time, while the gain is substantial (in PSNR and in visual quality). Therefore, while BM3D+DA3D is comparable to BM3D-SAPCA in terms of performance, its computation is more than 200 time faster.

6. CONCLUSIONS AND FUTURE WORK

This paper presented DA3D, a fast Data Adaptive Dual Domain Denoising algorithm for "last step" processing. It performs frequency domain shrinkage on shape and data-adaptive patches. The key innovations of this method are a sparse processing that allows bigger blocks to be used and a plane regression that greatly improves the results on gradients and smooth parts. The experiments show that DA3D can improve the results of most denoising algorithms with reasonable computational cost, achieving a performance superior to the state-of-the-art.

Future work will include optimizing the parameters of the algorithm, especially for low levels of noise, and adapting the shrinkage function to the algorithm used as guide.

7. REFERENCES

[1] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and F. C. Harris Jr., "Survey of image denoising techniques," *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," *WASE ICIE*, vol. 1, 2009.

[4] D. Gnanadurai and V. Sadasivam, "Image denoising using double density wavelet transform based adaptive thresholding technique," *IJWMIP*, vol. 03, no. 01, 2005.

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

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

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

[8] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE TPAMI*, vol. 12, no. 7, pp. 629–639, Jul 1990.

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

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

[12] 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.

[13] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," *IEEE ICCV*, pp. 2272–2279, Sept 2009.

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

[15] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE TIP*, vol. 22, no. 2, pp. 700–711, 2013.

[16] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," *IEEE ICCV*, pp. 479–486, Nov 2011.

[17] 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.

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

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

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

[21] I. Mosseri, M. Zontak, and M. Irani, "Combining the power of internal and external denoising," *IEEE ICCP*, pp. 1–9, April 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, A. Buades, and J. M. Morel, "Implementation of the "non-local bayes" (NL-bayes) image denoising algorithm," *Image Processing On Line*, 2013.

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

[25] H. C. Burger, C. Schuler, and S. Harmeling, "Learning how to combine internal and external denoising methods," in *Pattern Recognition*, vol. 8142 of *Lecture Notes in Computer Science*, pp. 121–130. Springer Berlin Heidelberg, 2013.

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

[27] C. Knaus, *Dual-domain image denoising*, Ph.D. thesis, Diss. Univ. Bern, 2013, Bern, 2013.

[28] C. Knaus and M. Zwicker, "Progressive image denoising," *IEEE TIP*, vol. 23, no. 7, pp. 3114–3125, July 2014.

[29] N. Pierazzo, M. Lebrun, M. Rais, J. M. Morel, and G. Facciolo, "Non-local dual image denoising," *IEEE ICIP*, 2014.

[30] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D Image Denoising with Shape-Adaptive Principal Component Analysis," *SPARS*, 2009.

[31] C. A. Deledalle, V. Duval, and J. Salmon, "Non-local methods with Shape-Adaptive patches (NLM-SAP)," *JMIV*, vol. 43, no. 2, pp. 103–120, May 2012.

[32] A. Buades, B. Coll, and J. M. Morel, "The staircasing effect in neighborhood filters and its solution," *IEEE TIP*, vol. 15, no. 6, pp. 1499–1505, June 2006.

[33] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE TIP*, vol. 16, no. 2, pp. 349–366, February 2007.

[34] H. Takeda, S. Farsiu, and P. Milanfar, "Higher order bilateral filters and their properties," *Electronic Imaging*, p. 64980S, 2007.

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

[36] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM SIGGRAPH*, pp. 257–266, 2002.

[37] A. B. Adams, *High-dimensional gaussian filtering for computational photography*, Stanford University, 2011.

[38] E. Gastal and M. M. Oliveira, "Adaptive manifolds for real-time high-dimensional filtering," *ACM Trans. Graph.*, vol. 31, no. 4, July 2012.

[39] Authors, "DA3D support website and supplementary material," `http://dev.ipol.im/~pierazzo/da3d`, 2015.

[40] 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.