Bounded Non-Local Means for Fast and Effective Image Denoising

Federico Tombari(✉) and Luigi Di Stefano

University of Bologna, Bologna, Italy
{federico.tombari,luigi.stefano}@unibo.it
http://vision.deis.unibo.it

Abstract. Non-Local Means (NLM) is a powerful but computationally expensive image denoising algorithm, which estimates a noiseless pixel as a weighted average across a large surrounding region whereby pixels centered at more similar patches are given higher weights. In this paper, we propose a method aimed at improving the computational efficiency of NLM by quick pre-selection of dissimilar patches thanks to a rapidly computable upper bound of the weighting function. Unlike previous approaches, our technique mathematically guarantees all highly correlated patches to be accounted for while discarding dissimilar ones, this providing not only faster speed but improved denoising too.

Keywords: Non local means · Image denoising · Fast bounding method

1 Introduction and Related Work

Image denoising is a recurrent topic in image processing research due to the ubiquitous presence of noise in the image formation and acquisition process. Research on effective image denoising algorithms has brought a wealth of methods (see [2],[4] for a review). Among the several proposals, Non-Local Means (NLM) [2] has demonstrated remarkable effectiveness even in presence of high noise levels. With NLM, a noiseless pixel is estimated by averaging across image positions so that pixels centered at highly correlated patches contribute more. This conceptually simple and effective approach comes at the cost of a huge computational complexity, which is theoretically $O(n^2 r_p^2)$, n being the number of pixels in the image and r_p the size of the side of the adopted squared patch. Therefore, the search for correlated patches is limited in practice to a squared surrounding *search area* of size r_s, the computational complexity decreasing to $O(n r_s^2 r_p^2)$ accordingly.

Several algorithms aimed at speeding up NLM have been proposed in literature. In [8], a pre-selection of contributing neighboring pixels is carried out by computing each patch's average value and gradients. In [1] candidate preselection is accomplished by arranging the data in a cluster tree where each leaf node is constrained to a minimum size, so that each pixel can be weighted with a relatively large subset of similar patches. The method in [13] proposes to

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part II, LNCS 9280, pp. 183–193, 2015.
DOI: 10.1007/978-3-319-23234-8_18

estimate probabilistically the dissimilarity between two patches out of a small portion of the distance function in order to terminate the computation early for highly uncorrelated patches. On completely different grounds, [11] uses PCA to project the set of neighboring candidates onto a lower dimensional subspace, thus reducing the computational burden by computing distances in the subspace rather than at full dimensionality. A fast approximated scheme based on a multi-resolution computation of the NLM weights is proposed in [5]. Finally, in [7] the computation of the distance is carried out in the Fourier domain to achieve improved efficiency when the patch size and search area size are different enough [6].

One drawback associated with methods based on quick pre-selection of mismatching image positions such as [8],[1],[13] is that they might also eliminate useful (i.e. highly correlated) candidates, thus implying computational savings to be achieved at the expense of some deterioration of the final result: e.g., in [8], a high difference in mean intensity and gradient orientation between two patches does not guarantee that their distance is always high. In our approach, instead, we aim at selecting mismatching candidates while at the same time guaranteeing that all highly correlated image positions are included in the final averaging operation. This is made possible thanks to the deployment of an efficiently computable lower bound of the dissimilarity function used in the NLM weight formulation. Moreover, discarding candidates that are guaranteed to lie - patchwise - far away from the current pixel tends not only to speed-up the computation but also to improve the accuracy of the denoising process. Another advantage of the proposed technique is that the minimum guaranteed correlation between two patches is an explicit parameter which can be easily set by the user so as to lean towards either higher speed or higher accuracy.

The paper will introduce the NLM algorithm in Section 2.1 and illustrate the proposed technique in Section 2.2. Successively, an experimental comparison including a standard benchmark dataset is carried out in Section 3. Finally, conclusions are drawn in Section 4.

2 Weighting by a Bounding Function

2.1 Original NLM

Like many denoising approaches, the NLM algorithm computes the noiseless estimate $\tilde{I}(i)$ of pixel $I(i)$ as a weighted sum within a surrounding region centered at position i:

$$\tilde{I}(i) = \sum_{j \in s(i)} w(i,j) \cdot I(j) \tag{1}$$

Notably, the *supporting pixel set* $s(i)$ in (1) may be in principle the whole image, although a squared window of size r_s is used in practice for the sake of computational tractability.

The weighting advocated in [2] consists of a decreasing function of the dissimilarity between the patch centered at position i and that centered at j, under

the assumption that the noiseless estimate of $I(i)$ should be obtained by averaging the intensities of those locations featuring image patches similar to that centered at i:

$$w(i,j) = \frac{1}{Z(i)} exp\left(-\frac{\| I(\mathcal{N}(i)) - I(\mathcal{N}(j)) \|^2_{2,a}}{h^2}\right) \tag{2}$$

In (2), $\mathcal{N}(i)$ and $\mathcal{N}(j)$ are square image patches of size r_p centered at i and j, h is a parameter of the method, $\| \cdot \|^2_{2,a}$ is the L_2 norm weighted by a Gaussian function centered at the patch and with standard deviation a and Z is a normalization factor:

$$Z(i) = \sum_{j \in s(i)} exp\left(-\frac{\| I(\mathcal{N}(i)) - I(\mathcal{N}(j)) \|^2_{2,a}}{h^2}\right) \tag{3}$$

Successively, the authors publicly released an implementation of NLM[1] based on a definition of the weighting function slightly different to that originally proposed in [2]:

$$w'(i,j) = exp\left(-\frac{max\left(\| I(\mathcal{N}(i)) - I(\mathcal{N}(j)) \|^2_2 - 2\sigma^2, 0\right)}{h^2}\right) \tag{4}$$

with σ related to the estimated amount of noise affecting the image. The authors recommend now to use this new formulation as it obtains improved results[2].

2.2 Proposed Bounded NLM

The main idea underpinning our approach is that, while image positions exhibiting highly similar patches represent a valuable subset to rely upon to estimate a noiseless pixel, those yielding a low degree of similarity do not bring in any useful contribution in the averaging process formulated in (1); indeed, they tend to distort the final estimate and should thus better be discarded from the averaging process to improve the quality of the final image. An inherent advantage of such an approach deals with low-correlation patches being often high in number within the search area surrounding a pixel, so that if we are able to devise a method which is able to quickly detect mismatching patches, we would obtain both improved accuracy as well as significant computational savings.

Purposely, we define a *test* to be evaluated at each pixel $j \in s(i)$ *before* the actual computation of the weight $w'(i,j)$:

$$e(i,j) > \tau \tag{5}$$

In (5), function $e(i,j)$ is a lower bound of the L_2 distance between the patch centered at i and that centered at j, while τ is a parameter. Accordingly, we

[1] www.ipol.im/pub/art/2011/bcm_nlm

[2] From personal communication with A. Buades.

Algorithm 1. The Bounded-NLM denoising algorithm

```
I = image of n pixels
s = search area of radius r_s
compute b, the box-filtered squared norms of r_p-sized patches ∈ I
for i ∈ I do
  Z(i) = 0
  for j ∈ s(i) do
    if (b(i) − b(j))^2 > τ then
      w'(i, j) = 0
    else
      compute w'(i, j) as in (4)
      Z(i) = Z(i) + w'(i, j)
    end if
    w'(i, j) = w'(i, j)/Z(i)
    compute Ĩ(i) as in (1)
  end for
end for
```

define a new weighting function $w_{\tau}^{'}$ so that if test (5) holds at position j the associated weight is set to 0:

$$w_{\tau}^{'}(i,j) = \begin{cases} 0, & e(i,j) > \tau \\ w'(i,j), & otherwise \end{cases} \tag{6}$$

As $e(i,j)$ is a lower bound of the L_2 distance, all positions at which (5) holds would have yielded a patch distance higher than τ. Hence parameter τ represents the maximum dissimilarity above which a weight is set to zero and it is guaranteed that no position j closer than τ to i is going to be discarded by test (5). In particular, we choose:

$$e(i,j) = \left(\| I(\mathcal{N}(i)) \|_2^2 - \| I(\mathcal{N}(j)) \|_2^2 \right)^2 \tag{7}$$

which turns out to be a lower bound of the L_2 distance

$$e(i,j) \leq \| I(\mathcal{N}(i)) - I(\mathcal{N}(j)) \|_2^2, \forall i,j \tag{8}$$

due to the triangular inequality. Moreover, the chosen function $e(i,j)$ is very efficiently computable because the two norms appearing in (7) can be calculated once and for all at initialization time and independently of the size of the patch via fast incremental schemes such as Box Filtering [9] or Integral Images [3]. The pseudo-code of the proposed algorithm is reported in Alg. 1.

Hence, the higher the number of positions satisfying test (5) the more substantial are the computational savings due to weights being immediately just set to zero without calculating neither the dissimilarity function, i.e. the right hand side in (8), nor (4). Remarkably, the proposed method guarantees that all pixels showing a degree of dissimilarity to i smaller than τ will be included in the averaging process required to compute its noiseless estimate. This compares favorably with respect to other approaches in literature aimed at selecting

Table 1. PSNR and efficiency comparison between NLM and bNLM on a standard benchmark dataset at increasing noise levels ($\sigma = 5, \cdots, 40$). At each noise level, the same value of $\tilde{\tau}$ is used in bNLM throughout the dataset (reported on the left). Efficiency is compared in terms of measured execution times (reported in seconds in the Table).

				Barbara (512x512)				***Boat*** (512x512)				***Fingerprint*** (512x512)			
				NLM		bNLM		NLM		bNLM		NLM		bNLM	
σ	r_p	r_s	$\tilde{\tau}$	PSNR	t	PSNR	t	PSNR	t	PSNR	t	PSNR	t	PSNR	t
5	1	10	4	37.04	4.69	37.07	1.79	36.58	4.82	36.61	1.99	35.14	4.19	35.08	0.68
10	1	10	6.6	33.17	5.08	33.24	2.29	32.92	5.16	33.04	2.46	31.03	4.49	31.09	1.03
15	1	10	10	30.81	5.16	30.87	2.76	30.73	5.25	30.90	2.90	28.74	4.78	28.85	1.41
20	2	10	10	30.25	9.91	30.32	5.25	29.76	10.80	29.93	5.56	27.30	9.43	27.33	2.67
25	2	10	10	29.09	9.86	29.18	5.03	28.62	9.86	28.88	5.31	26.25	9.68	26.33	2.70
30	2	10	13	28.08	9.93	28.24	5.67	27.69	9.88	28.00	5.94	25.36	9.84	25.56	3.31
35	3	17	8	27.45	45.90	27.82	16.81	26.81	45.84	27.35	19.21	24.80	46.07	25.01	12.39
40	3	17	8	26.50	46.38	27.03	16.58	26.03	45.83	26.69	18.67	24.03	46.37	24.39	12.50
				House (256x256)				***Lena*** (512x512)				***Peppers*** (256x256)			
				NLM		bNLM		NLM		bNLM		NLM		bNLM	
σ	r_p	r_s	$\tilde{\tau}$	PSNR	t	PSNR	t	PSNR	t	PSNR	t	PSNR	t	PSNR	t
5	1	10	4	38.59	1.23	38.63	0.72	37.90	5.00	37.96	2.23	37.30	1.17	37.34	0.39
10	1	10	6.6	34.98	1.25	34.98	0.77	34.30	5.20	34.44	2.68	33.52	1.22	33.68	0.50
15	1	10	10	32.82	1.25	32.83	0.83	32.07	5.30	32.24	3.09	31.21	1.25	31.42	0.60
20	2	10	10	32.48	2.36	32.56	1.58	31.55	9.83	31.75	5.83	30.32	2.34	30.51	1.08
25	2	10	10	31.33	2.36	31.40	1.46	30.46	9.86	30.71	5.56	29.15	2.37	29.40	1.05
30	2	10	13	30.28	2.37	30.48	1.57	29.54	9.85	29.85	6.16	28.16	2.38	28.47	1.19
35	3	17	8	29.75	10.79	30.23	5.20	28.89	45.87	29.46	19.36	27.22	10.83	27.73	3.30
40	3	17	8	28.80	10.78	29.44	4.98	28.09	45.83	28.78	18.82	26.30	10.85	26.99	3.26

a suitable subset of candidates to compute the noiseless estimate of a pixels [8],[1],[13], as previous proposals may instead discard highly correlated candidates and therefore potentially weaken the denoising process.

It is worth pointing out that more effective and complex schemes have been proposed in literature to bound dissimilarity functions derived from the L_2-norm [10]. Yet, these approaches are conceived to deliver notable computational benefits with patch sizes as large as required in typical template matching applications, whilst they can hardly provide similar advantages when applied to much smaller patches (e.g. 7×7) as those usually deployed for the purpose of image denoising in algorithms like NLM. In particular, we verified experimentally that a more advanced incremental scheme such as IDA [12]) does not provide additional advantages with respect to the simple bounding function defined in (7). Moreover, many of such methods constrain the patch size, e.g. to be even, while this is not the case of original NLM formulation.

3 Experimental Results

In this Section we compare the proposed approach, which will be referred to in these experiments as *bNLM* (*bounded*-NLM), to the NLM algorithm, so as

Table 2. Comparing bNLM to [8],[1],[13] according to the results reported in [1] (Table 3) and in [13] (Table 4). Results related to method [1] in Table 3 concern increasing block overlaps (a,b,c), as described in [1].

	PSNR	S.U.
NLM (2)	30.31	//
NLM (4)	30.70	//
bNLM	30.72	2.17
[8]	29.80	2.38
[1],a	30.26	1.34
[1],b	30.08	2.14
[1],c	29.83	2.9

Table 3. *Barbara*, $\sigma = 20$, $r_p = 4$, $r_s = 10$

	PSNR	S.U.
NLM (2)	27.83	//
NLM (4)	30.47	//
bNLM	30.63	2.91
[8]	27.78	2.13
[1]	27.51	2.68
[13]	27.60	4.35

Table 4. *Peppers*, $\sigma = 20$, $r_p = 3$, $r_s = 11$

to assess quantitatively its effectiveness in both speeding up and improving the original technique. We have chosen to use the NLM version based on the weighting function defined in (4) for two reasons. First, the chosen one is the most recent NLM formulation proposed by the authors. Second, the publicly available authors' code and suggested parameter values concern this more recent version only: using their code and parameter settings allow the comparison to be carried out with the best possible fairness. Indeed, to actually implement and evaluate our own method we have modified the authors' code only in those parts strictly related to our proposal and then run the program with exactly the same parameters as suggested by them[3], which concern specifically the patch size r_p, the search area size r_s and parameter h.

As for the parameter introduced by our method, i.e. the maximum dissimilarity τ, the experiments have been conducted using the normalized value $\tilde{\tau}$:

$$\tau = \tilde{\tau}^2 \cdot (2r_p + 1)^2 \tag{9}$$

Thus, $\tilde{\tau}$ compares directly to the average pixelwise difference and ranges within $[0, 255]$, so it is an easier parameter to interpret and set with respect to τ.

Table 1 reports the results yielded by NLM and bNLM on standard benchmark datasets, where each image is corrupted by additive Gaussian noise with standard deviation ranging from $\sigma = 5$ up to $\sigma = 40$. As for the choice of parameter $\tilde{\tau}$ for bNLM, we have set a fixed value for each noise level, chosen so to privilege denoising accuracy. Yet, depending on the application settings, other choices of this parameter may be preferred so to favor computational savings rather than denoising accuracy: this can be easily achieved by decreasing the value of τ so to increase the number of candidates being discarded by (5). The Table reports, for each image, noise level and evaluated method, the PSNR (Peak Signal-to-Noise Ratio) and the measured execution time (in seconds). To measure execution times, the same platform has been used to run both the NLM and the bNLM code, i.e. an Intel Core i7 with 64 GB RAM. As vouched by the Table, bNLM consistently outperform NLM in terms of both computational

[3] www.ipol.im/pub/art/2011/bcm_nlm

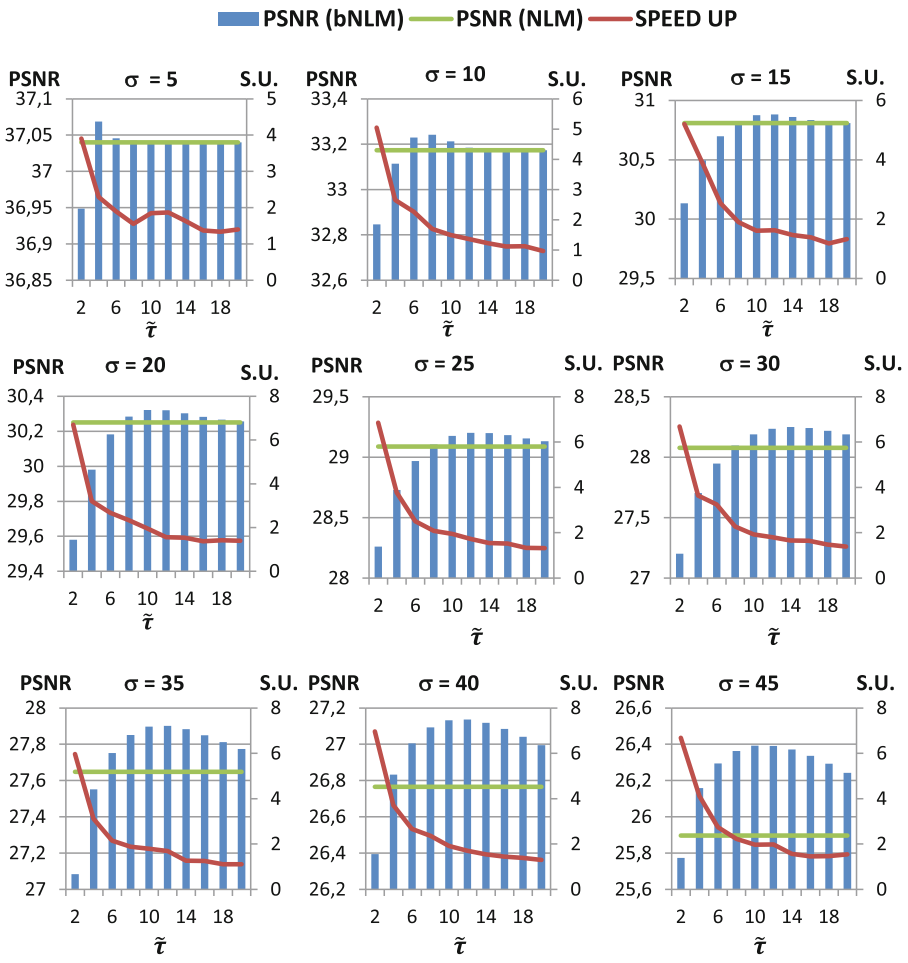

Fig. 1. Quantitative comparison between bNLM and NLM on the *Barbara* test image. Each chart is related to a different noise level and reports as a function of $\tilde{\tau}$ both the PSNR yielded by bNLM and NLM (range reported on the left vertical axis) as well as the speed-up provided by bNLM over NLM (range on the right vertical axis).

savings as well as denoising accuracy, yielding higher PSNRs and higher efficiency with all test images, the improvement in terms of PSNR ranging up to 0.91.

The experimental results provided in this Section indicate also that, when using similar patch and search area sizes, the proposed method compares favorably with respect to previous pre-selection algorithms aimed at speeding up NLM, such as [8],[1],[13]. Indeed, the results reported in Table 1 in [1] show

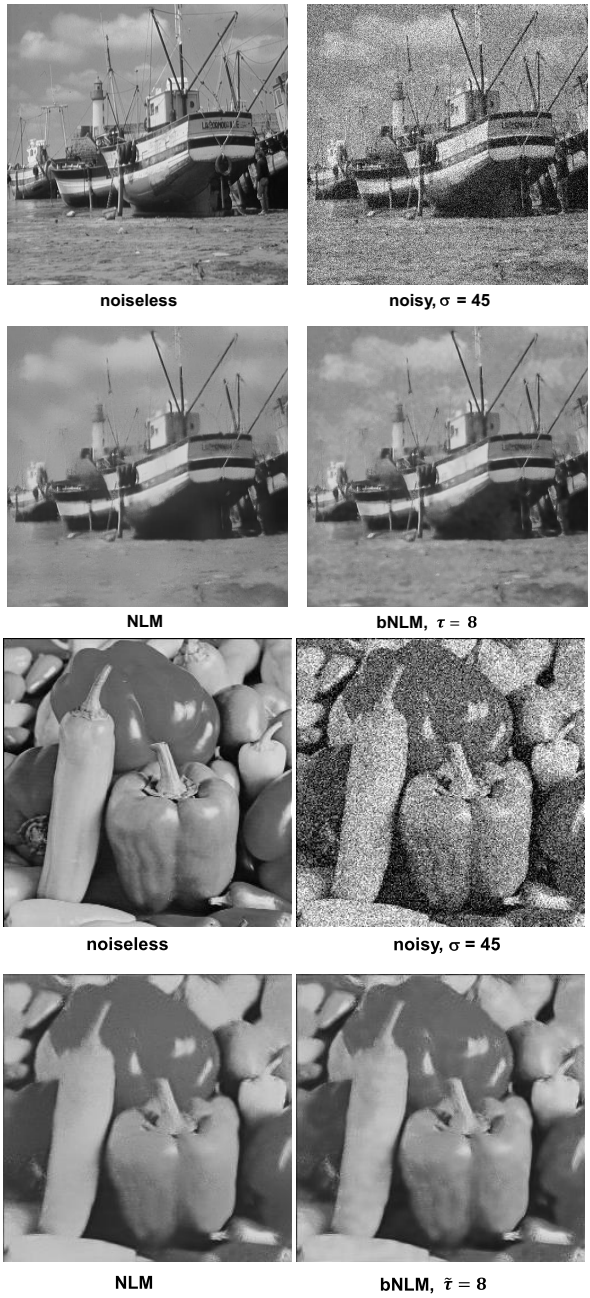

Fig. 2. Comparison between NLM and bNLM on *Boat* (top) and *Peppers* (bottom). Both figures show, respectively, the noiseless image (top left), the noisy image with $\sigma = 45$ (top right), the noiseless estimation by NLM (bottom left) and by bNLM with the suggested $\tilde{\tau}$ value as in Table 1 (reported in the bottom right part of each Figure)

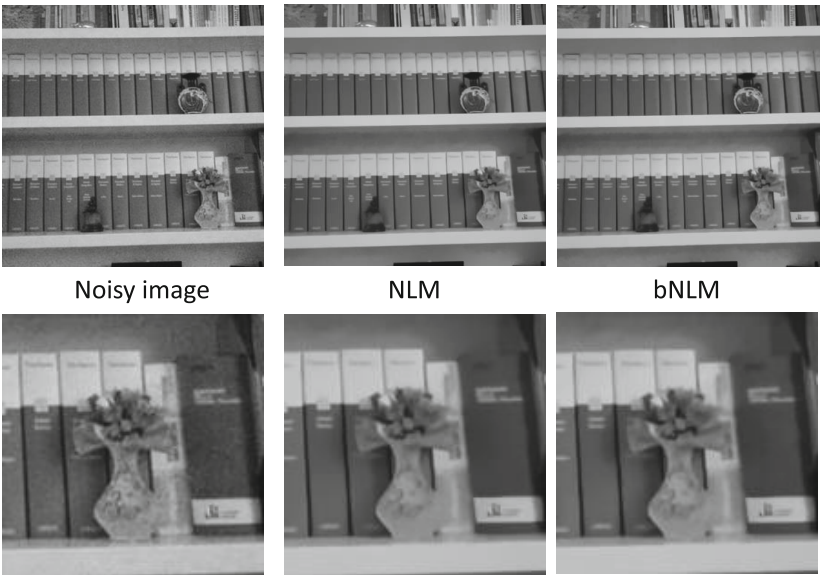

Fig. 3. Comparison on real noise. Upper row: qualitative comparison between bNLM and NLM on an image affected by real noise. Lower row: comparison on a zoomed detail of the original image (bottom right corner).

that on *Barbara* neither [1] nor [8] can improve the PSNR with respect to NLM. Similarly, Tables I and II in [13], show that with *Peppers* and *Lena* and a search windows of size $23x23$, NLM yields always a higher PSNR with respect to [8], [1] and [13]. A more detailed comparison between bNLM and [8],[1],[13] is reported in Tables 3 and 4. To obtain this data, bNLM parameters have been set exactly as in the considered experiments in [1] and [13]. Also, the Tables report the PSNR yielded by both NLM algorithms, i.e. as based on either formula (2) or formula (4). The two Tables highlight that bNLM provides always the highest PSNR with respect to previous fast NLM-based methods while turning out either second-best (Table b) or third-best (Table a) in terms of speed-up.

To provide more insights on the behavior of our method, in Fig. 1 we consider the *Barbara* image corrupted by Gaussian noise, with $\sigma = 5$ up to $\sigma = 45$, and compare bNLM to NLM in terms of both PSNR (left vertical axis) as well as speed-up (right vertical axis) while varying $\tilde{\tau}$ values. As it can be observed from the Figure, with small τ values bNLM can report remarkable speed-ups with respect to NLM without introducing a significant deterioration of the PSNR. By increasing $\tilde{\tau}$, bNLM starts discarding a smaller number of candidates, this resulting in lower speed-ups but also higher PSNRs. It is worth highlighting how, with proper choices for $\tilde{\tau}$, in each of the experiments depicted in Fig. 1 bNLM can yield higher PSNRs than NLM, in particular at the higher noise levels, this validating the idea that discarding dissimilar patches is beneficial to

improve not only efficiency but denoising accuracy as well. In the three most challenging denoising experiments considered in Fig. 1, bNLM can be tuned to deliver both a substantial speed-up (i.e. larger than 4) and a higher PSNR with respect to NLM. When $\tilde{\tau} \rightarrow \infty$, bNLM behaves exactly as NLM (no candidate is discarded), so that, by moving rightward along the horizontal axis of the charts, speed-ups get close to 1 and the PSNRs yielded by the two algorithms becomes more and more similar.

In addition to quantitative comparisons, we propose qualitative experiments aimed at assessing the perceived denoising accuracy. Accordingly, Fig. 2 compares NLM and bNLM on the *Boat* and *Peppers* test images corrupted by Gaussian noise ($\sigma = 45$). The Figure shows how the proposed algorithm can restore a higher amount of details in the estimated noiseless image, while NLM tends to introduce more over-smoothing. This is particularly evident in the ground surface and in the recovered cloud patterns in the sky of the *Boat* image, as well as on the surface of the foreground vegetables depicted in *Peppers*.

Finally, Fig. 3, addresses the case of images corrupted by real noise. In the figure, the upper row shows the original test image acquired by the camera of a Nexus 5 smartphone under indoor lighting conditions, together with the output yielded by NLM and bNLM, the parameters of the two algorithms set as in Table 1, $\sigma = 10$. The bottom row shows also a zoomed detail of the original image (taken from the bottom right corner) and the corresponding output by the two compared methods. As it can be observed, bNLM can effectively smooth out noise while preserving edges, the perceived quality being substantially equivalent for the two considered methods. bNLM, though, turns out the most efficient algorithm in these settings, running in 0.91 seconds, i.e. remarkably faster than NLM, which requires 2.11 seconds.

4 Concluding Remarks

A candidate selection scheme for the NLM image denoising algorithm has been proposed. By deploying a lower bound of the dissimilarity function employed to compute NLM weights, the proposed approach can safely discard dissimilar patches, so as to peculiarly provide both higher efficiency as well as improved denoising accuracy with respect to NLM. Experimental results show that our proposal is more beneficial as the noise level corrupting the image gets higher. Possible extensions to this approach would include exploiting more effective lower bounding function in spite of (8), so to increase the computational savings associated with the proposed algorithm, at the same time rejecting weakly correlated patches with respect to that of the current position.

References

1. Brox, T., Kleinschmid, O., Cremers, D.: Efficient nonlocal means for denoising of textural patterns. IEEE Trans. Image Processing **17**(7), 1083–1092 (2008)

2. Buades, A., Coll, B., Morel, J.: A review of image denoising methods, with a new one. SIAM Multiscale Modeling and Simulation **4**(2), 490–530 (2005)
3. Crow, F.: Summed-area tables for texture mapping. Computer Graphics **18**(3), 207–212 (1984)
4. Danielyan, A., Katkovnik, V., Egiazarian, K.: Bm3d frames and variational image deblurring. IEEE Trans. Image Processing **21**(4), 1715–1728 (2012)
5. Karnati, V., Uliyar, M., Dey, S.: Fast non-local algorithm for image denoising. In: Proc. Int. Conf. on Image Processing (ICIP) (2009)
6. Lewis, J.: Fast template matching. Vision Interface, pp. 120–123 (1995)
7. Liu, Y., Wang, J., Chen, X., Guo, Y., Peng, Q.: A robust and fast non-local means algorithm for image denoising. J. Computer Science and Technology **23**(2), 270–279 (2008)
8. Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. IEEE Signal Processing Letters **12**(12), 839–842 (2005)
9. McDonnell, M.: Box-filtering techniques. Computer Graphics and Image Processing **17**(1), 65–70 (1981)
10. Ouyang, W., Tombari, F., Mattoccia, S., Di Stefano, L., Cham, W.K.: Performance evaluation of full search equivalent pattern matching algorithms. Trans. Pattern Analysis and Machine Intelligence (PAMI) **34**(1), 127–143 (2012)
11. Tasdizen, T.: Principal neighborhood dictionaries for nonlocal means image denoising. IEEE Trans. Image Processing **18**(12), 2649–2660 (2009)
12. Tombari, F., Mattoccia, S., Di Stefano, L.: Full search-equivalent pattern matching with incremental dissimilarity approximations. Trans. Pattern Analysis and Machine Intelligence (PAMI) **31**(1), 129–141 (2009)
13. Vignesh, R., Oh, B., Kuo, C.: Fast non-local means (nlm) computation with probabilistic early termination. IEEE Signal Processing Letters **17**(3), 277–280 (2010)