

 Open access • Journal Article • DOI:10.1109/LSP.2005.859509

## Fast image and video denoising via nonlocal means of similar neighborhoods

— [Source link](#) 

Mona Mahmoudi, Guillermo Sapiro

**Institutions:** University of Minnesota

**Published on:** 21 Nov 2005 - IEEE Signal Processing Letters (IEEE SIGNAL PROCESSING LETTERS)

**Topics:** Non-local means, Video denoising, Grayscale, Pixel and Image processing

Related papers:

- [A Review of Image Denoising Algorithms, with a New One](#)
- [A non-local algorithm for image denoising](#)
- [Bilateral filtering for gray and color images](#)
- [Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering](#)
- [Scale-space and edge detection using anisotropic diffusion](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/fast-image-and-video-denoising-via-nonlocal-means-of-similar-k8l1lu0lyw>

**FAST IMAGE AND VIDEO DENOISING VIA NON-LOCAL MEANS  
OF SIMILAR NEIGHBORHOODS**

By

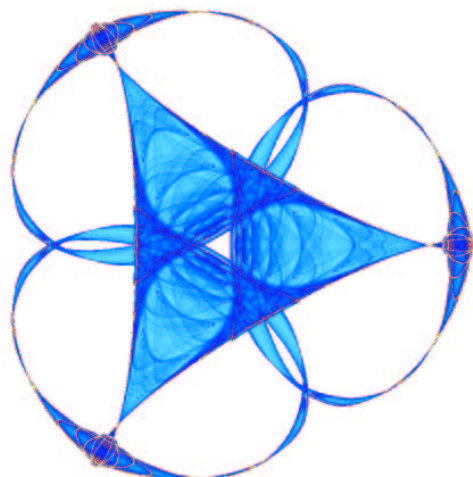
**Mona Mahmoudi**

and

**Guillermo Sapiro**

**IMA Preprint Series # 2052**

(June 2005)



**INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS**

UNIVERSITY OF MINNESOTA  
400 Lind Hall  
207 Church Street S.E.  
Minneapolis, Minnesota 55455-0436

Phone: 612/624-6066 Fax: 612/626-7370

URL: <http://www.ima.umn.edu>

# Fast Image and Video Denoising via Non-Local Means of Similar Neighborhoods

Mona Mahmoudi and Guillermo Sapiro

**Abstract**—In this note, improvements to the non-local means image denoising method introduced in [2], [3] are presented. The original non-local means method replaces a noisy pixel by the weighted average of pixels with related surrounding neighborhoods. While producing state-of-the-art denoising results, this method is computationally impractical. In order to accelerate the algorithm, we introduce filters that eliminate unrelated neighborhoods from the weighted average. These filters are based on local average gray values and gradients, pre-classifying neighborhoods and thereby reducing the original quadratic complexity to a linear one and reducing the influence of less-related areas in the denoising of a given pixel. We present the underlying framework and experimental results for gray level and color images as well as for video.

*Index Terms* - Image and video denoising, non-local neighborhood filters, contexts, computational complexity.

## I. INTRODUCTION

Denoising is still one of the most fundamental, widely studied, and largely unsolved problems in image processing. The purpose of denoising (or restoration) is to estimate the original image (or a “better” representative of it) from noisy data. Many methods for image denoising have been suggested, and an outstanding review of them can be found in [2]. This paper also proposes a very elegant non-local image denoising method shown to produce state-of-the-art results. In this method, the restored gray value of each pixel is obtained by the weighted average of the gray values of all pixels in the image. Each weight is proportional to the similarity between the local neighborhood of the pixel being processed and the neighborhood corresponding to the other image pixels (the optimality of this approach under reasonable criteria is shown in [2] as well). The basic idea is that images contain repeated structures, and averaging them will reduce the (random) noise. This new concept for image denoising is popular in other image processing areas, such as texture synthesis, where a new pixel is synthesized as the weighted average of known image pixels with similar neighborhoods [4], [9], [10]. The authors of [6] proposed a method closely related to the one in [2], where the denoised pixel is obtained sampling from similar contexts that are learned from the image. The paper includes fundamental theoretical results showing the optimality of the proposed technique (which by the way, is of course related to original ideas by Shanon on producing new values by sampling from their contexts).

The work here presented is partially supported by the Office of Naval Research, the National Science Foundation, and the National Geospatial-Intelligence Agency.

Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, {mahmo022,guille}@umn.edu

Although the quality of the results in [2] is state-of-the-art, this method is quite slow to be practically realizable. The high computational complexity is due to the cost of weights calculation for all pixels in the image during the process of denoising. For every pixel being processed, the whole image is searched and differences between corresponding neighborhoods are computed (see below). The complexity is then quadratic in the number of image pixels. This has been addressed in a follow-up paper by the authors, [3], limiting the weight computation to a sub-image surrounding the pixel being processed (as commonly done for example for motion estimation in video compression). This sub-image will still have to be quite large for high resolution images and to make sure enough similar neighborhoods are included in the computation. In this note we address the computational complexity of the algorithm proposed in [2], [3] in a different fashion, which we believe is more in harmony with the concepts introduced in these papers. We significantly improve the computational complexity at no quality cost (and the quality can even be improved, see experimental section for details).

The basic idea here proposed is to pre-classify the image blocks according to fundamental characteristics such as their average gray values and gradient orientation. This is performed in a first path, and while denoising in the second path, only blocks with similar characteristics are used to compute the weights. Accessing these blocks can be efficiently implemented with simple look-up tables. The basic idea is then to combine ideas from [2], namely weighted average based on neighborhoods similarity, with concepts which are classical in information theory and were introduced in image denoising in [6], namely contexts. As in [6], and in contrast with [2], the algorithm running time is linear in the number of image pixels. In contrast with [6], the “contexts” are not learned (which is only asymptotically optimal), but pre-determined, mainly based on prior information about what is important to determine block similarities. And in contrast with the speed-up method proposed in [3], the blocks/neighborhoods subset selection is based also on block similarity and not on spatial proximity, much in the spirit of the algorithm itself.

The remainder of this note is as follows. In Section II, a brief description of the non-local means method of [2] is presented. In Section III, we introduce our new approach in detail. Finally, Section IV presents examples and concluding remarks are provided in Section V.

## II. THE NON-LOCAL MEANS IMAGE DENOISING METHOD

In this section, a brief overview of the non-local means method introduced in [2] is presented. Let  $v(i)$  and  $u(i)$  be

the observed noisy and original images respectively, where  $i$  is the pixel index. The restored values can be derived as the weighted average of all gray values in the image (indexed in the set  $I$ ):

$$NL(v)(i) = \sum_{j \in I} w(i, j)v(j), \quad (1)$$

where  $NL(v)(i)$  is the restored value at pixel  $i$ . The weights express the amount of similarity between the neighborhoods of each pair of pixels involved in the computation ( $i$  and  $j$ ),

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}}, \quad (2)$$

where  $Z(i)$  is a normalizing factor,  $Z(i) = \sum_j w(i, j)$ , and  $h$  is the decay parameter of the weights. In the above equation,  $v(N_i)$  is the vector of neighborhood pixel values,  $v(N_i) := (v(j))$ ,  $j \in N_i$ , where  $N_i$  defines the neighborhood of pixel  $i$ , normally a square-block of pre-defined size around  $i$ . The vector norm used in Equation (1) is simply the Euclidean difference, weighted by a Gaussian of zero mean and variance  $a$  [2]. For an image with  $M$  pixels,  $M$  weights have to be computed for each pixel. Computation of the  $M^2$  overall weights makes the algorithm inefficient and impractical. Reducing the total number of computed weights by neglecting in advance neighborhoods with expected small weights is important in order to improve the computational complexity of this non-local means algorithm. This reduction may also improve the overall denoising quality by removing the influence of pixels belonging to unrelated neighborhoods. We show how to approach this next.

### III. NEIGHBORHOODS CLASSIFICATION

In this section, two types of filters are suggested to pre-classify the image blocks and thereby reduce the number of weight computations in the non-local means denoising algorithm. One of the filters is based on average neighborhood gray values and the second one is based on gradient (directions). Finding easily computed measures for neighborhood similarity is fundamental to make the non-local means algorithm practical.

First, the average gray values in the neighborhood of each pixel is introduced as one measure of similarity between pixels. Intuitively, considering zero-mean additive noise, similar neighborhoods should have similar average gray values. In our proposed algorithm, for each pixel  $i$  a maximum of  $2n + 1$  weights are calculated, for the  $2n + 1$  pixels  $j$  with closest neighborhood average gray value to that of  $i$ . Depending on the selected value of  $n$ , the average of the obtained  $2n + 1$  neighborhoods might be too far from the average for the neighborhood of the pixel being processed. Therefore, in addition to using a fixed pre-defined number of blocks, we consider the ratio of average gray values in the neighborhoods of pixels  $i$  and  $j$  when computing  $w(i, j)$ . The weight  $w(i, j)$  will have a non-zero value (that is, the corresponding neighborhood will be considered) if  $\eta_1 < \frac{\bar{v}(i)}{\bar{v}(j)} < \eta_2$ , where  $\bar{v}(i)$  and  $\bar{v}(j)$  are the average gray values in the neighborhoods of pixels  $i$  and  $j$ , and  $\eta_1 < 1$  and  $\eta_2 > 1$  are two constants close to one.<sup>1</sup>

<sup>1</sup>Considering this ratio criteria alone might lead to too many blocks if large similar areas exist in the image, thereby hurting the computational speed.

Following a first path where these average gray values have been computed for all the needed image blocks, these closest blocks can be easily accessed with a  $O(1)$  complexity look-up table addressed by the (quantized) average gray value of the neighborhood for the current pixel being processed.

Another method to approximate the similarity between two neighborhoods is their average gradient. If  $\nabla v(i) = (v_x(i), v_y(i))$  stands for the image gradient, the average gradient in the neighborhood of pixel  $i$  is defined as

$$\overline{\nabla v}(i) = (\overline{v_x}(i), \overline{v_y}(i)), \quad (3)$$

where  $\overline{v_x}(i)$  and  $\overline{v_y}(i)$  are the average horizontal and vertical derivatives in the neighborhood of pixel  $i$  (derivatives computed with standard numerics). In contrast with works such as [5], [8], where the magnitude of the gradient is considered, we here use the gradient orientation. Note for example that a noise-free image block will have very different average gradient magnitude with the same block but with (zero-mean) additive noise, while the gradient direction is expected to be similar. The average gradient orientations difference at pixels  $i$  and  $j$  is given by

$$\theta(i, j) = \angle(\overline{\nabla v}(i), \overline{\nabla v}(j)), \quad (4)$$

and can also be used as a measure to filter-out unrelated neighborhoods (once again, blocks can be easily pre-classified and accessed with look-up tables).

To find a threshold, above which  $\theta$  is considered as outlier (meaning not the same type of neighborhood block), we use robust statistics following [1], [7]

$$\sigma_\theta = 1.4826 \text{ median}_{I \times I} [|\theta(i, j)| - \text{median}_{I \times I} (|\theta(i, j)|)]. \quad (5)$$

We have observed that the gradient orientation is not a reliable measure for neighborhood similarity when the overall gradient magnitude of the block is small. Therefore, once the blocks have similar averages, the weight  $w(i, j)$  is computed (non-zero) if the gradients in pixel  $i$  or  $j$  are small or  $\theta(i, j) < \sigma_\theta$ . A similar formula as in Equation (5) is used to compute the threshold under which the gradient is considered small.

To recapp, all the above mentioned filters are brought together in the following equation:

$$w(i, j) = \begin{cases} \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|_{2,a}^2}{h^2}}, & [(\|\nabla v(i)\| < \sigma_\nabla) \\ & \text{or } (\|\nabla v(j)\| < \sigma_\nabla) \\ & \text{or } (|\theta(i, j)| < \sigma_\theta)] \\ & \text{and } (\eta_1 < \frac{\bar{v}(i)}{\bar{v}(j)} < \eta_2) \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

where  $\sigma_\nabla$  is the threshold under which the gradient magnitude is considered small. As mentioned before, using look-up tables, the conditions in the above formula reduce the complexity from quadratic to linear.

### IV. EXPERIMENTAL RESULTS

We tested our algorithm both on gray scale and color images. We used a  $11 \times 11$  neighborhood window for average gradient computation and a  $7 \times 7$  window for average gray

value computations and similarity tests. The parameters  $n$  and  $h$  were chosen following experimentation, while we selected  $\eta_1 = 0.9$  and  $\eta_2 = 1.1$ .

In Figure 1 we compare the original non-local means algorithm with our proposed modifications. For this image, which was originally noisy,  $h = 10$  and  $n = 100$ . The proposed algorithm is 10.53 times faster than the original method in [2]. It can also be observed that some details of the building are better preserved in our algorithm as a result of removing unrelated pixels from the weighted average. In general, the proposed modification works better in parts with more details, e.g., parts of the building in Figure 1, while flat regions like the sky are better denoised by including more blocks as in [2]. The main reason is the large number of similar pixels in flat parts compared to detailed parts. Therefore, there is a trade off in the selection of the number of blocks with similar average to be considered: A large number of blocks reduces the speed of the denoising process, though it results in better denoising mainly in flat parts of the image.<sup>2</sup> In Figure 2 we modified the algorithm parameters following this observation, in order to further smooth the image, while the obtained speed is 7.15 times less than before. In the example in Figure 3, the selected parameters are  $n = 50$  and  $h = 7$ , while our method is 24 times faster than the original method.



Fig. 1. Top to bottom and left to right: Original noisy image, denoised image by our algorithm, denoised image by the original method in [2].

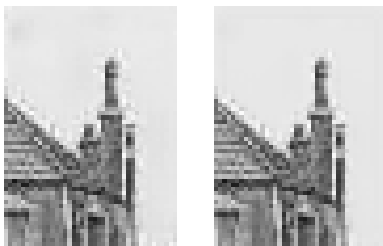


Fig. 2. The left image is denoised with  $n = 50$  while the right image is denoised with  $n = 500$ , a 5 times larger  $\sigma_{\nabla}$ ,  $\eta_1 = .5$ , and  $\eta_2 = 1.5$ .

Examples for color images are presented in Figure 4. In computing the weights, the  $L_2$  norm of pixels' difference

<sup>2</sup>The thresholds can depend on the block characteristics themselves, e.g., allowing for more blocks when the average gradient magnitude of the block is small.

vector (RGB) is used instead of the difference of gray values in the gray scale images. In Equation (6),  $\sigma_{\nabla}$  is a  $3 \times 1$  vector,  $\sigma_{\theta}$  is defined using the average of 3 orientations, and instead of  $v$  in  $\bar{v}$  the average of the 3 RGB values is used.

Finally, in Figure 5 four frames of a noisy image sequence and their denoised version are presented. For video, the computational improvement introduced in this paper becomes even more critical. As clearly detailed in [3], there is no need for optical flow computation, and the only modification is that the neighborhood around the pixel being denoised is compared with neighborhoods also in adjacent frames. As detailed in [10], where this was used for texture synthesis, searching for a flat 2D neighborhood in the 3D data (space plus time) is more appropriate than searching for a 3D neighborhood.

## V. CONCLUSIONS

In this note, improvements to the original non-local means image denoising method introduced in [2] were proposed. In order to significantly accelerate the algorithm, we introduced filters to eliminate unrelated neighborhoods from the weighted average used to denoise each image pixel. These filters are based on average gray values as well as gradients, pre-classifying neighborhoods and thereby reducing the quadratic complexity to a linear one and diminishing the influence of less-related areas in the denoising of a given pixel.

The work here presented can be considered as a combination of techniques from [2] with those in [6]. Part of our ongoing efforts include the investigation of image characteristics that provide good context classifications for image denoising. Results in this direction will be reported elsewhere.

## REFERENCES

- [1] M. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Processing*, pp.421-432, March 1998.
- [2] A. Buades, B. Coll, and J. M. Morel, "On image denoising methods," *CMLA Preprint* (<http://www.cmla.ens-cachan.fr/Cmla/>), to appear in *SIAM Multiscale Modeling and Simulation*, 2004.
- [3] A. Buades, B. Coll, and J. M. Morel, "Denoising image sequences does not require motion estimation," *CMLA Preprint* **2005:18**, (<http://www.cmla.ens-cachan.fr/Cmla/>), 2005.
- [4] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," *IEEE International Conference on Computer Vision, Corfu, Greece*, pp. 1033-1038, Sept. 1999.
- [5] S. Kindermann, S. Osher, and P. W. Jones, "Deblurring and denoising of images by nonlocal functionals," *UCLA-CAM Report* **04-75** (<http://www.math.ucla.edu/applied/cam/index.html>), December 2004
- [6] E. Ordentlich, G. Seroussi, S. Verdú, M. J. Weinberger, and T. Weissman, "A discrete universal denoiser and its application to binary images," *Proc. of IEEE International Conference on Image Processing (ICIP03)*, Barcelona, Spain, September 2003.
- [7] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*, New York: Wiley, 1987.
- [8] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D* **60**, pp. 259-268, 1992.
- [9] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [10] L. Yatziv, G. Sapiro, and M. Levoy, "Light field completion," *Proc. IEEE-International Conference on Image Processing, Singapore*, 2004.



Fig. 3. Left to right: Original image, denoised image by our algorithm, denoised image by the method in [2].



Fig. 4. Left to right: Original, noisy, and denoised image by our method.

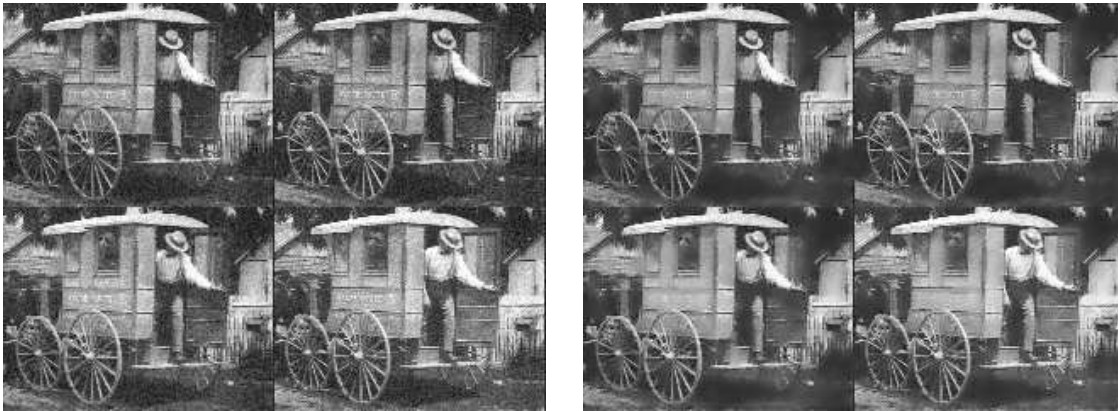


Fig. 5. Four noisy frames (left) of a video sequence denoised by our algorithm (right).