A Simple Trick to Speed Up the Non-Local Means

Laurent Condat

A Simple Trick to Speed Up the Non-Local Means

Laurent Condat, *Member, IEEE*

***Abstract*—We show that the popular Non-Local Means method for image denoising can be implemented exactly, easily and with lower computation time using convolutions.**

***Index Terms*—Non-Local Means, Denoising, Patches**

I. Introduction

Images are often corrupted by noise during acquisition [1], so that effective noise suppression methods are required. The *Non-local means* (NL-means) is a popular technique developed by Buades *et al.* [2], [3]. Its efficiency and conceptual simplicity have made it very popular, but its main drawback is its high computational complexity. For an image of size $N \times M$, patches of size 7×7 and a search window of size 21×21, which is a typical use setting [2], the complexity is $7^2.21^2.N.M$. This makes accelerations necessary to maintain low computation times. Numerous methods were proposed to accelerate the NL-means, such as preselection of the contributing neighborhoods based on average value and gradient [4], average and variance [5], higher-order statistical moments [6], cluster tree arrangement [7], mean values at different resolutions [8], or probabilistic early termination [9]. The use of a data-driven lower dimensional subspace of the space of image neighborhoods has been investigated as well [10]–[12]. Also, the computation of the distance between different neighborhoods can be optimized using the fast Fourier transform [13], [14], a moving average filter [15] or integral images of certain error terms [8], [16]. Finally, Adams *et al.* proposed clever data structures which are used to perform multi-dimensional smoothing, including NL-means filtering [17], [18].

In this work, we show that there exists an easy and exact implementation of the NL-means using convolution routines, which accelerates the method significantly in comparison with the classical naive implementation. Despite its simplicity, this observation has not been made so far in the literature, to the best of the author's knowledge. In Section II, we briefly describe the NL-means method and we present our new implementation based on convolutions in Section III.

II. The Non-Local Means Method

We consider the following observation model. We have at our disposal the image $y = (y[\mathbf{k}])_{\mathbf{k}\in\Omega}$, where $\Omega \subset \mathbb{Z}^2$ is the image domain, of size $N \times M$; y is a noisy version of the unknown image x corrupted by additive white Gaussian noise (AWGN):

$$y[\mathbf{k}] = x[\mathbf{k}] + e[\mathbf{k}], \quad \forall \mathbf{k} \in \Omega, \tag{1}$$

L. Condat is with the Image Team of the GREYC laboratory, a joint CNRS-UCBN-ENSICAEN research unit in Caen, France. Contact: see http://www.greyc.ensicaen.fr/˜lcondat/.

where $e[\mathbf{k}] \sim \mathcal{N}(0,\sigma^2)$ for every $\mathbf{k} \in \Omega$ and σ^2 is the noise variance. The denoised image z, which is an estimate of x, is formed as follows:

$$z[\mathbf{k}] = \frac{\sum_{\mathbf{n}\in\mathbb{Z}^2} w[\mathbf{k},\mathbf{n}]y[\mathbf{k}+\mathbf{n}]}{\sum_{\mathbf{n}\in\mathbb{Z}^2} w[\mathbf{k},\mathbf{n}]} \tag{2}$$

with the weights

$$w[\mathbf{k},\mathbf{n}] = f(\mathbf{n})\,\exp\!\left(-\frac{\sum_{\mathbf{m}\in\mathbb{Z}^2} h(\mathbf{m})\big(y[\mathbf{k}+\mathbf{n}+\mathbf{m}] - y[\mathbf{k}+\mathbf{m}]\big)^2}{\lambda \sum_{\mathbf{m}\in\mathbb{Z}^2} h(\mathbf{m})}\right) \tag{3}$$

for every $\mathbf{n} \neq \mathbf{0}$, where λ controls the amount of smoothing. The weight attached to the current pixel is computed differently [3]:

$$w[\mathbf{k},\mathbf{0}] = \max_{\mathbf{n}\neq\mathbf{0}} w[\mathbf{k},\mathbf{n}] \tag{4}$$

Let us call the set of pixels $\{y[\mathbf{m}+\mathbf{k}]\ ;\ \mathbf{m} \in [-P,\dots,P]^2\}$ the *patch* of the image y centered at the pixel $\mathbf{k}$. The NL-means consists in blending every pixel of y with other pixels, based on the similarity of their surrounding patches. Typically, the search for the similar patches is limited to the neighborhood of the local pixel so that f is the indicator function of the search region: $f = \mathbf{1}_{[-S,\dots,S]^2}$. f can be a more complex symmetric function decaying away from zero, like a Gaussian function in [2]. h is an indicator function, which characterizes the size of the patches: $h = \mathbf{1}_{[-P,\dots,P]^2}$.

III. The NL-Means Expressed Using Convolutions

The proposed acceleration consists in letting convolutions appear in the computation process. For this, let us define the images $u_\mathbf{n}$ and $v_\mathbf{n}$ by

$$u_\mathbf{n}[\mathbf{k}] = \big(y[\mathbf{k}+\mathbf{n}] - y[\mathbf{k}]\big)^2 \tag{5}$$

and

$$v_\mathbf{n} = u_\mathbf{n} * h, \tag{6}$$

where $*$ indicates the convolution, for every $\mathbf{n} \in [-S,\dots,S]^2$ and $\mathbf{k} \in \Omega$, using symmetric boundary conditions for pixel values outside Ω. Next, we assume that f and h are arbitrary but symmetric functions and that f has a compact support included in $[-S,\dots,S]^2$. Then, we have

$$w[\mathbf{k},\mathbf{n}] = f(\mathbf{n})\exp\big(-v_\mathbf{n}[\mathbf{k}]/C\big), \tag{7}$$

where $C = \lambda \sum_{\mathbf{m}\in\mathbb{Z}^2} h(\mathbf{m})$ is a constant. Moreover,

$$w[\mathbf{k},-\mathbf{n}] = w[\mathbf{k}-\mathbf{n},\mathbf{n}]. \tag{8}$$

Hence, the proposed implementation simply consists in swapping the loops with respect to $\mathbf{k}$ and $\mathbf{n}$, which yields the following algorithm:

Proposed Algorithm

1) for every $\mathbf{n} \in [-S, \dots, S]^2$ such that $\mathbf{n} < \mathbf{0}$ (in lexicographic order),
2) compute the image of squared pixel values $u_\mathbf{n}$
3) perform the convolution $v_\mathbf{n} := u_\mathbf{n} * h$
4) for every $\mathbf{k} \in \Omega$, update the pixel values $z[\mathbf{k}] := z[\mathbf{k}] + f(\mathbf{n}) \exp(-v_\mathbf{n}[\mathbf{k}]/C) y[\mathbf{k}+\mathbf{n}]$ and $z[\mathbf{k}+\mathbf{n}] := z[\mathbf{k}+\mathbf{n}] + f(\mathbf{n}) \exp(-v_\mathbf{n}[\mathbf{k}]/C) y[\mathbf{k}]$
5) add the contribution of the noisy pixel values to their denoised versions: $\forall \mathbf{k} \in \Omega$, $z[\mathbf{k}] := z[\mathbf{k}] + \max_{\mathbf{n} \neq \mathbf{0}} y[\mathbf{k}]$.
6) normalize the pixel values: $\forall \mathbf{k} \in \Omega$, $z[\mathbf{k}] := z[\mathbf{k}] / \sum_{\mathbf{n} \in \mathbf{Z}^2} w[\mathbf{k}, \mathbf{n}]$.

For the classical NL-means with $h = \mathbf{1}_{[-P,\dots,P]^2}$, the convolution with h is separable, so that the complexity of the proposed algorithm is $O((2P+1)(2S+1)^2 N.M)$ instead of $O((2P+1)^2(2S+1)^2 N.M)$. A non-optimized C implementation of our algorithm running on a 2.4 GHz Macbook Pro yields a computation time of $15.4s$ versus $50.2s$ for the naive implementation, when denoising a 512×512 grayscale image, with $P = 3$ and $S = 10$.

Further on, the moving average convolution with the 1-D filter

$$H(z) = z^{-P} + \cdots + z^{P} \tag{9}$$

can be simplified, using the equality

$$H(z) = \frac{-z^{-P-1} + z^{P}}{1 - z^{-1}}, \tag{10}$$

which yields the moving average implementations given in [15] and [16]. Using this recursive implementation of moving averages, the complexity is reduced to $O((2S+1)^2 N.M)$.

Furthermore, the proposed algorithm opens the door to the use of non-binary filters h, yielding “fuzzy” patches. For instance, a Gaussian filter h can be employed, using the fast recursive implementations available for Gaussian filtering [19], [20]. Preliminary results indicate that better denoising results are obtained using the tensor-product of the simple two-poles recursive filter

$$H(z) = \frac{4}{(2 - z^{-1})(2 - z)}. \tag{11}$$

With the latter, the complexity is reduced to $O((2S+1)^2 N.M)$ as well.

References

[1] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data,” *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[2] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Proc. of IEEE CVPR*, vol. 2, 2005, pp. 60–65.

[3] ——, “A review of image denoising algorithms, with a new one,” *SIAM Journal on Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[4] M. Mahmoudi and G. Sapiro, “Fast image and video denoising via nonlocal means of similar neighborhoods,” *IEEE Signal Process. Lett.*, pp. 839–842, 2005.

[5] P. Coupé, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, “An optimized blockwise nonlocal means denoising lter for 3-D magnetic resonance images,” *IEEE Trans. Med. Imag.*, vol. 27, no. 4, pp. 425–441, Apr. 2008.

[6] A. Dauwe, B. Goossens, H. Luong, and W. Philips, “A fast non-local image denoising algorithm,” in *Proc. of SPIE Electronic Imaging*, vol. 6812, San Jose, CA, Jan. 2008, pp. 1331–1334.

[7] T. Brox, O. Kleinschmidt, and D. Cremers, “Efcient nonlocal means for denoising of textural patterns,” *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1083–1092, Jul. 2008.

[8] V. Karnati, M. Uliyar, and S. Dey, “Fast non-local algorithm for image denoising,” in *Proc. of IEEE ICIP*, Cairo, Egypt, 2009.

[9] R. Vignesh, B. T. Oh, and C.-C. J. Kuo, “Fast non-local means (NLM) computation with probabilistic early termination,” *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 277–280, 2010.

[10] N. Azzabou, N. Paragios, and F. Guichard, “Image denoising based on adapted dictionary computation,” in *Proc. of IEEE ICIP*, vol. III, 2007, pp. 109–112.

[11] J. Orchard, M. Ebrahimi, and A. Wong, “Efficient nonlocal-means denoising using the SVD,” in *Proc. of IEEE ICIP*, San Diego, CA, 2008.

[12] T. Tasdizen, “Principal neighborhood dictionaries for nonlocal means image denoising,” *IEEE Trans. Image Process.*, vol. 18, no. 12, pp. 2649–2660, Dec. 2009.

[13] J. Wang, Y. Guo, Y. Ying, Y. Liu, and Q. Peng, “Fast non-local algorithm for image denoising,” in *Proc. of IEEE ICIP*, Atlanta, GE, 2006, pp. 1429–1432.

[14] Y.-L. Liu, J. Wang, X. Chen, Y.-W. Guo, and Q.-S. Peng, “A robust and fast non-local means algorithm for image denoising,” *J. Computer Science and Technology*, vol. 23, no. 2, pp. 270–279, Mar. 2008.

[15] B. Goossens, H. Luong, A. Pižurica, and W. Philips, “An improved non-local denoising algorithm,” in *Proc. of LNLA*, Lausanne, Switzerland, 2008.

[16] J. Darbon, A. Cunha, T. F. Chan, and S. Osher, “Fast nonlocal ltering applied to electron cryomicroscopy,” in *Proc. of IEEE ISBI*, 2008, pp. 1331–1334.

[17] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, “Gaussian KD-trees for fast high-dimensional filtering,” in *Proc. of ACM SIGGRAPH*, New Orleans, Louisiana, USA, 2009.

[18] A. Adams, J. Baek, and A. Davis, “Fast high-dimensional filtering using the permutohedral lattice,” in *Proc. of EuroGraphics*, 2010.

[19] R. Deriche, “Recursively implementing the gaussian and its derivatives,” in *Proc. of IEEE ICIP*, 1992, pp. 263–267.

[20] L. Van Vliet, I. Young, and P. W. Verbeek, “Recursive gaussian derivative filters,” in *Proc. of ICPR*, Brisbane, Australia, 1994, pp. 509–514.