Fast nonlocal filtering applied to electron cryomicroscopy

Conference Paper *in* Proceedings / IEEE International Symposium on Biomedical Imaging: from nano to macro. IEEE International Symposium on Biomedical Imaging · May 2008

DOI: 10.1109/ISBI.2008.4541250 · Source: DBLP

5 authors, including:

Alexandre Cunha
California Institute of Technology
43 PUBLICATIONS **531** CITATIONS

SEE PROFILE

Stanley Osher
University of California, Los Angeles
609 PUBLICATIONS **81,921** CITATIONS

SEE PROFILE

FAST NONLOCAL FILTERING APPLIED TO ELECTRON CRYOMICROSCOPY

Jérôme Darbon[1], *Alexandre Cunha*[2], *Tony F. Chan*[1], *Stanley Osher*[1], *Grant J. Jensen*[3]

[1]Department of Mathematics, University of California Los Angeles
[2]Center for Advanced Computing Research, California Institute of Technology
[3]Division of Biology, California Institute of Technology
{jerome,chan,sjo}@math.ucla.edu, {cunha,jensen}@caltech.edu

ABSTRACT

We present an efficient algorithm for nonlocal image filtering with applications in electron cryomicroscopy. Our denoising algorithm is a rewriting of the recently proposed nonlocal mean filter. It builds on the separable property of neighborhood filtering to offer a fast parallel and vectorized implementation in contemporary shared memory computer architectures while reducing the theoretical computational complexity of the original filter. In practice, our approach is much faster than a serial, non–vectorized implementation and it scales linearly with image size. We demonstrate its efficiency in data sets from *Caulobacter crescentus* tomograms and a cryoimage containing viruses and provide visual evidences attesting the remarkable quality of the nonlocal means scheme in the context of cryoimaging. With such development we provide biologists with an attractive filtering tool to facilitate their scientific discoveries.

Index Terms— Nonlocal mean filtering, image denoising, electron cryomicroscopy, image vectorization, SIMD, parallel image processing.

1. INTRODUCTION

Electron cryomicroscopy is a remarkable technology enabling new discoveries at subcellular scale. Studies at such scale aim to understand the structure and function of the macromolecular machinery responsible for regulating cell mechanisms with the ultimate goal of transforming living cells to attain desired configurations and perform specific tasks, such as, cure diseases and transform plants in fuel.

Researchers rely on pictures of samples maintained at cryo (freezing) temperatures to investigate a particular cell type in its near native state. A picture of a cell is formed after beam of electrons is projected through the sample and captured by a charged-coupled device, CCD, camera. Due to a low dosage of electrons, necessary to avoid damaging the thin biological sample, a poorly resolved, noisy, low contrast image is formed (see Fig. 1). In the case of cryotomography, a series of unfiltered 2D projections are combined to build a 3D reconstruction of the cell. In this process, noise is transferred to the reconstructed object. All this makes filtering a *de facto* component of the cryoimage processing pipeline.

A number of factors contribute to make filtering of cryoimages (an image generated using the electron cryomicroscopy technique) a challenging problem. First and foremost, the amount of noise is significantly high (see Fig. 1), far beyond what is commonly found in regular photography and other microscopy technologies. A signal to noise ratio of 1 or lower is not uncommon [1]. Second, large, high resolution images are the rule and the trend is to continue increasing the CCD resolution to obtain images with even higher levels of detail (current cameras can generate images having up to 4096^2 pixels). Third, structures of interest in the images can be either a single filament or a bundle of filaments [2], which are a few points wide that when mixed with noise are challenging to visualize even for a trained eye. Fourth, validating the results might be difficult since the shapes of the pictured cell and of its macromolecular structures are most likely unknown (hence the reason to photograph the cell in the first place) and therefore a counterpart to help make comparisons is unknown. The experienced biologist plays a major role in the validation.

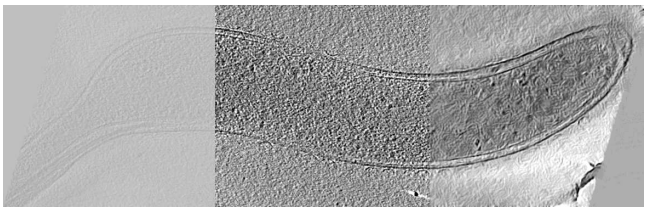


Fig. 1. This composition shows a section of a typical low contrast tomogram after reconstruction (left portion), the level of noise more apparent after histogram normalization (middle), and the image after applying our filter.

The solution to this challenge is a *robust* and *efficient* filter capable of *preserving fiber like structures*. The *nonlocal means* filter, NL-means for short, recently proposed by Buades, Coll, and Morel [3], and experimented and extended by others, e.g. [4, 5], has the right features that interest us: it is simple, *straightforward* to implement, and the results shown in the literature are outstanding. The filter is primarily designed to reduce noise without destroying textures and fine structures, a superior feat when compared to other equally extraordinary filters which, contrary to NL–means, interpret textures as *oscillating patterns* and remove them altogether to form a separation of the image into a cartoon part and an oscillating (noise + texture) part. We have recently discovered the work of Dabov *et.al.* [6] which has strong similarities to NL–means.

To the best of our knowledge, practical attempts of coding NL–means have led to slow implementations which somehow shadowed the full potential of the method. The straightforwardness of the proposed algorithm in [3] does not directly translate into an efficient implementation. We depart from the sliding window scheme typically adopted by neighborhood filtering implementations and

Research of J. Darbon and S. Osher was supported by ONR grant N000140710810. Research of A. Cunha and G.J. Jensen was supported in part by a grant gift to Caltech from the Gordon and Betty Moore Foundation. Research of T.F. Chan was supported by ONR grant N00014-06-0345 and NSF grant DMS-0610079.

resort on more sophisticated ideas leading to a faster implementation. We claim that *attaining a desirable level of applicability*, as advocated by [3], is due to our algorithm and implementation bringing the NL–means ideas to a full realization.

We emphasize here efficiency as it permits practitioners to execute trial and error runs at an affordable pace enabling a prompt progress of their ultimate investigations. Considering the uncertainties and difficulties in picturing frozen cells as described earlier practitioners would greatly benefit from a fast implementation. This would allow and encourage them to quickly test different denoising scenarios and make choices satisfying their own quality criteria – visual inspection continues to be the most reliable metric to assess denoising results.

For lack of space, we will not compare NL–means with other well established filters (e.g. anisotropic diffusion, bilateral, median, Wiener, total variation based filters) and refer the reader to the literature for pragmatical comparisons (see *e.g.* [3, 5]). These popular filters comprise the majority of attempts in denoising cryoimages (see *e.g.* [1, 7, 8]). We believe results can be further improved in terms of robustness and efficiency and we shall present here a few supporting examples.

Outline. In the next section we present the nonlocal mean approach and our fast algorithm. Results are presented in section 3 and we offer some conclusions thereafter.

2. THE NONLOCAL ALGORITHM

In this section we briefly review the nonlocal approach of Buades *et. al.* [3] before presenting our fast algorithm.

2.1. The Nonlocal Means Approach

The NL-means scheme thrives when an image contains many repetitive structured patterns. It uses redundant information to reduce noise by performing a weighted average of pixel values. Formally, we assume images are defined over a discrete regular grid Ω of dimension d and cardinality $|\Omega|$. Let us denote by v the original noisy image. The value of the restored image u at a site $s \in \Omega$ is defined as the convex combination

$$u(s) = \frac{1}{Z(s)} \sum_{t \in \mathcal{N}(s)} w(s,t) v(t) \ , \tag{1}$$

where $w(\cdot,\cdot)$ are non-negative weights, $Z(s)$ is a normalization constant such that for any site s we have $Z(s) = \sum_{t \in \mathcal{N}(s)} w(s,t)$, and $\mathcal{N}(s)$ corresponds to a set of neighboring sites of s. Following Buades *et. al.* $\mathcal{N}(\cdot)$ will be referred to as the searching window. The weight $w(s,t)$ measures the similarity between two square patches centered, respectively, at sites s and t, and it is defined as follows:

$$w(s,t) = g_h \left(\sum_{\delta \in \Delta} G_\sigma(\delta) \left(v(s+\delta) - v(t+\delta) \right)^2 \right) \ , \tag{2}$$

where G_σ is a Gaussian kernel of variance σ^2, $g_h : \mathbb{R}^+ \to \mathbb{R}^+$ is a continuous non-increasing function with $g_h(0) = 1$ and $\lim_{x \to +\infty} g_h(x) = 0$, and Δ represents the discrete patch region containing the neighboring sites δ. The parameter h is used to control the amount of filtering. Typical examples of the g_h function are $g_h(x) = \frac{1}{1+(x^2/h^2)}$ and $g_h(x) = e^{-x^2/h^2}$, the latter used in [3]. We adopt the former which leads to a more efficient implementation. In summary, NL-means restores an image by performing a weighted average of pixel values taking into account spatial and intensity similarities between pixels. Similarity is computed between equally sized patches as they capture the local structures (geometry and texture) around the sites in consideration. Note that pixels outside $\mathcal{N}(s)$ do not contribute to the value of $u(s)$. This property allows us to separate the image into independent disjoint pieces and process them in parallel, as it is done in domain decomposition schemes.

Following [3], we assume searching windows $\mathcal{N}$ and patches Δ have uniform cardinalities of, respectively, $(2K+1)^d$ and $(2P+1)^d$ with $\mathcal{N} = [\![-K, K]\!]^d$ and $\Delta = [\![-P, P]\!]^d$. Using equations (1) and (2) one realizes the nonlocal mean algorithm proposed in [3] which has $O(|\Omega| K^d P^d)$ for time complexity. Note that this complexity is exponential with respect to space dimension d but it is polynomial with respect to the number of pixels $|\Omega|$. In practice, where dimensions are fixed and low, the algorithm remains polynomial.

To make averaging more robust, one would like to set the searching window $\mathcal{N}(\cdot)$ as large as possible and in the limit extend it to the entire image. However, this would lead to excessively long computation times and thus we constrain searching only in local neighborhoods, as suggested in [3]. We refer the reader to [9] for a variational formulation of neighborhood filtering in the lines of nonlocal mean.

2.2. A Fast Algorithm

Our fundamental contribution consists in a method for computing very efficiently the weights $w(s,t)$ given by Eq.(2). The weight computation is by far the most time consuming part when generating the restored image u. For the sake of clarity we present our algorithm for 1-dimensional images; extending it to higher dimensions is straightforward. Under this 1D assumption we have $\Omega = [\![0, n-1]\!]$, an image with n pixels.

Given a translation vector d_x, we introduce a new image S_{d_x} as

$$S_{d_x}(p) = \sum_{k=0}^{p} \left(v(k) - v(k+d_x) \right)^2 , \ \ p \in \Omega. \tag{3}$$

S_{d_x} corresponds to the discrete integration of the squared difference of the image v and its translation by d_x. Note that Eq. (3) may require access to pixels outside the image domain. To avoid memory corruption in our implementation we extend the image boundaries either by symmetry or in a periodic fashion.

Our strategy to compute the weight for two pixels s and t follows. Recall that in 1D we have patches of the form $\Delta = [\![-P, P]\!]$. Contrary to [3] we replace the Gaussian kernel by a constant without noticeable differences. Thus Eq. (2) rewrites as follows: $w(s,t) = g_h(\sum_{\delta_x \in \Delta} (v(s+\delta_x) - v(t+\delta_x))^2)$. Now let $d_x = (t-s)$ and define $\hat{p} = s + \delta_x$. With this reparametrization we write $w(s,t) = g_h(\sum_{\hat{p}=s-P}^{s+P} (v(\hat{p}) - v(\hat{p}+d_x))^2)$. If we split the sum and use the identity in Eq. (3) we obtain

$$w(s,t) = g_h \left(S_{d_x}(s+P) - S_{d_x}(s-P) \right) \tag{4}$$

which is in fact independent of t provided the quantity S_{d_x} is known. This is the key expression that allow us to compute the weight for a pair of pixels in constant time. Generalizing to higher dimensions corresponds to have integrations along the image orthogonal axes in Eq. (3). It is not difficult to see that this approach yields a weight computation formula requiring $O(2^d)$ operations for $d-$dimensional images. This quantity is *independent* of the size of the patches, contrary to the formulation in [3] which requires $O(P^d)$ operations per patch.

In summary, the algorithm works as follows: first, all values S_{d_x} are computed using Eq. (3); then weights are computed using Eq.(2) and Eq.(4); and finally the filtering is performed using Eq (1). This procedure is repeated for all possible translations given by the dimensions of the searching window $\mathcal{N}$. Pseudo-code of our algorithm is shown in Algorithm 1.

Algorithm 1 – Fast Nonlocal Mean – 1D

Input: v, K, P, h
Output: u
Temporary variables: images S_{d_x}, Z, M
Initialize u, M and Z to 0.
for all $d_x \in [\![-K, K]\!]$ **do**
  Compute S_{d_x} using Eq. (3)
  for all $s \in [\![0, n-1]\!]$ **do**
    compute weights w using Eq.(2) and Eq.(4)
    $u(s) \leftarrow u(s) + w \cdot u(s + d_x)$
    $M(s) = \max(M(s), w)$
    $Z(s) \leftarrow Z(s) + w$
  end for
end for
for all $s \in [\![0, n-1]\!]$ **do**
  $u(s) \leftarrow u(s) + M(s) \cdot v(s)$
  $u(s) \leftarrow \frac{u(s)}{Z(s)+M(s)}$
end for
return u

Time and memory complexity in the Random Access Model. Compared to the original NL-means approach our algorithm requires three additional images to store partial results but it has a much better time complexity of $O(|\Omega|K^d 2^d)$ when compared to $O(|\Omega|K^d P^d)$ of Buades *et. al.* – an improvement of order $O(P^d)$. It is also faster, when $d > 1$, than a sliding window-based approach that one could obtain if adapting [10]. This would result in a $O(P^{d-1})$ time complexity per pixel (K is nonexistent in median filtering). The method of [4] also improved the time complexity of the original NL-means [3]. However, it makes use of hashing techniques that only behave in constant time when considering an *amortized* time complexity.

Vectorization and parallelization. The nonlocal mean formulation permits decomposing an image into disjoint parts allowing for parallel denoising of the distinct image parts. This fact has already been explored in [5] to improve computation times. However, a much higher speed up is achieved when combining image decomposition with our strategy enabling vectorization of operations using contemporary SIMD (Single Instruction Multiple Data) instructions set. This is equivalent to parallelism at the data level. To use SIMD instructions our implementation must comply with the strict requirement of accessing aligned data in memory, something not necessary if implementing nonlocal mean with sliding windows, as we suspect it is done in [3, 4]. Our implementation employs the latest SIMD instructions set and benefits from the adoption of the prefix sum construct [11] to efficiently build S_{d_x} in a cache aware manner. It is important to mention that although usage of SIMD instructions and maximization of cache hits do not change the time complexity of an algorithm in the RAM model they drastically reduce the computational load of its implementation. Complexity analysis in the cache oblivious model would be more faithful to our design and implementation.

3. RESULTS

We implement our algorithm in C/C++ and demonstrate it in cryotomograms of *Caulobacter crescentus* bacteria and for a cryoimage containing viruses. Figures and tables together with their captions are self explanatory. All results are based on programs written by our own. We report timing and scalability numbers in tables to show the applicability of our work in large images and multicore computing. All results were obtained for $P = 3$, $K = 7$, and $h = 1$. Parallel runs were performed on a 32 GB shared memory, 2.8GHz, Dual-Core AMD Opteron Processor 8220 based server having 8 dual core chips, each core with 1 MB of cache. Times were averaged after 40 runs and do not include I/O. Resulting images were post-processed to improve visualization via histogram manipulation.

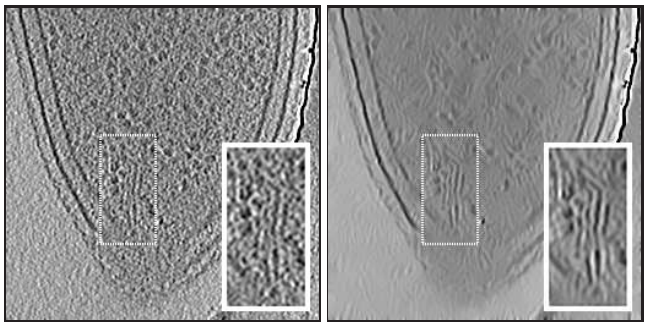
Regarding the performance of our formulation in 3D, we speculate and compare our results with those in [5]. Observe that an image with 4096^2 pixels requires the same amount of data processing as in a 3D image with 256^3 voxels. We would then expect equivalent filtering times for these. Given the performance numbers presented in our timing tables and those in [5] we estimate that our formulation in 3D would be approximately 20 times faster than the parallel implementation of [5]. We believe our estimation is conservative after considering the differences in image size, hardware, and searching window and patch sizes reported in [5].

Timings for filtering images of different sizes					
image size	256^2	512^2	1024^2	2048^2	4096^2
time (s)	0.07	0.35	1.54	6.05	26.32
scalability	–	5.0	4.4	3.9	4.4

Table 1. This table shows filtering times in seconds for images of different sizes using 2 cores of a dual core AMD Athlon X2 64. Image size increases linearly by a factor of 4. Similarly, processing time increases linearly by an average factor of 4.4 (average of values in the `scalability` row). An entry in the `scalability` row is the ratio of two consecutive values in the `time` row, and it shows how time scales as we increase image size. Note that small images residing in memory are filtered in real time.

Scalability results for a 8192 x 8192 cryoimage				
version	**cores**	**time(s)**	**speedup**	**efficiency(%)**
serial	1	7,071.42	1	100
parallel	1	313.12	23	2,300
	2	163.59	43	2,150
	4	102.21	69	1,725
	8	66.83	106	1,325
	16	64.15	110	688

Table 2. Scalability results after filtering a large image with 8192^2 pixels. In the `serial` row we report the time spent by the serial, non–vectorized version of NL–means (this corresponds to a straightforward, slow sliding window implementation). Note the significant speedup (ratio between serial and parallel times) obtained with the parallel versions. As expected, efficiency (percentage of speedup to number of cores) decreases as we employ more cores. We suspect that at 16 cores other variables other than memory size dominate performance preventing further reduction in processing time.



noisy bacterial filaments denoised filaments

Fig. 2. It is now known that many bacterial proteins do form filaments *in vivo* which play critical roles in cell shape and division. Electron cryomicroscopy is helping elucidate the molecular mechanisms related to these filaments by providing images where they can be seen in near–native state acting within their cellular context [2]. Denoising of filaments helps clarify their conformation. Above is a slice from a *Caulobacter crescentus* tomogram which has been NL-means denoised to significantly enhance visualization of filaments almost unseen in the noisy image.

4. CONCLUSION

We presented a fast algorithm to compute nonlocal filtering and demonstrated results after denoising cryoimages. Our algorithm is an efficient version of the nonlocal mean filter capable of producing equally good results. Our vectorized, data parallel implementation for shared memory architectures based on the latest SIMD instructions set is fast and scales well with image size and number of cores. We are currently extending our implementation to 3D and we are looking into making it available to the public at large afterward.

Acknowledgments. A. Cunha thanks Lu Gan, Songye Chen, Zhiheng Yu, and Prabha Dias for providing images and for fruitful discussions about electron cryomicroscopy.

5. REFERENCES

[1] Reiner Hergel and Achilleas S. Frangakis, "Denoising of electron tomograms," in *Electron Tomography*, pp. 331–352. Springer, 2nd edition, 2006.

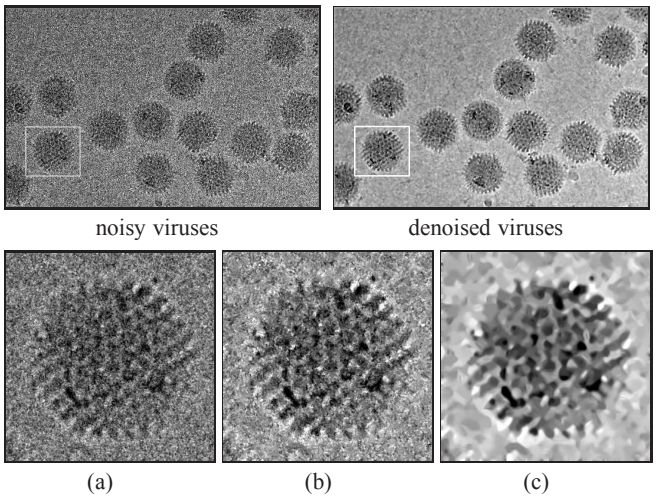
[2] A. Briegel, D.P. Dias, Z. Li, R.B. Jensen, A.S. Frangakis, and G.J. Jensen, "Multiple large filament bundles observed in Caulobacter crescentus by electron cryotomography," *Molecular Microbiology*, vol. 62, no. 1, pp. 5–14, 2006.

[3] A. Buades, B. Coll, and J.M. Morel, "A review of denoising algorithm, 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 encoding via nonlocal means of similar neighborhoods," *IEEE Signal Processing Letters*, vol. 12, pp. 839–842, 2005.

[5] P. Coupé, P. Yger, and C. Barillot, "Fast non local means denoising for 3D MRI images," in *MICCAI (2)*, Rasmus Larsen, Mads Nielsen, and Jon Sporring, Eds. 2006, vol. 4191 of *Lecture Notes in Computer Science*, pp. 33–40, Springer.

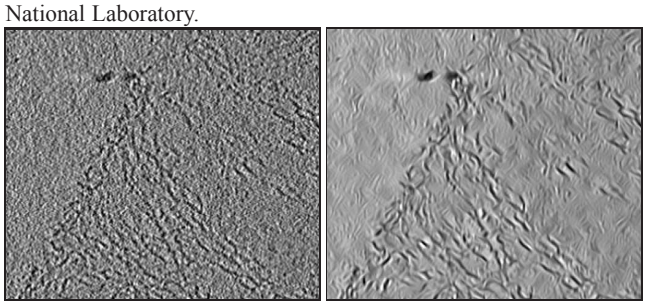
[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, , no. 8, pp. 2080–2095, August 2007.



noisy viruses denoised viruses

(a) (b) (c)

Fig. 3. Our filter has excelled in denoising this 2048^2 cryoimage containing viruses (only a portion of the entire image is shown on the top row). The NL–means thrives denoising the repetitive spikes present on the virus surface. This is most noticeable in (c), which was obtained after 4 passes of filtering (a). Image (b) is the result after one single pass. Image courtesy of J. He at Brookhaven National Laboratory.



noisy *succuli* denoised *succuli*

Fig. 4. Bacterial cell integrity is maintained by the *sacculus*, which is the stress-bearing layer of the cell wall synthesized from peptidoglycan. Tomograms of purified, hydrated *sacculi* from the gram-negative bacterium *Caulobacter crescentus* were denoised to clearly show how the fiber-like densities, which are interpreted to be glycan strands, are parallel to the cell surface, supporting a organizational model which was proposed more than 30 years ago.

[7] Wen Jiang, Matthew L. Baker, Qiu Wu, Chandrajit Bajaj, and Wah Chiu, "Applications of a bilateral denoising filter in biological electron microscopy," *Journal of Structural Biology*, vol. 144, no. 1-2, pp. 114–122, October 2003.

[8] Radosav S. Pantelic, Geoffery Ericksson, Nicholas Hamilton, and Ben Hankamer, "Bilateral edge filter: photometrically weighted, discontinuity based edge detection," *Journal of Structural Biology*, vol. 160, no. 1, pp. 93–102, October 2007.

[9] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," Tech. Rep. CAM-23, UCLA, 2007.

[10] T. Huang, G. Yang, and G. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Transactions on Accoustics, Speech and Signal Processing*, vol. 27, no. 1, pp. 13–18, 1979.

[11] G. Blelloch, *Prefix Sums and Their Applications*, chapter Synthesis of Parallel Algorithms, Morgan Kauffman, 1990.