

DNLM-IFFT: An Implementation of the Deceived Non Local Means Filter Using Integral Images and the Fast Fourier Transform for a Reduced Computational Cost

Saúl Calderón Ramírez and Manuel Zumbado Corrales^(✉)

Computing School, Costa Rican Institute of Technology,
PAttern Recognition and MACHine Learning Group (PARMA-Group),
159-7050 Cartago, Costa Rica
sacalderon@itcr.ac.cr, manu3193@gmail.com

Abstract. In this paper we propose an efficient implementation of the Deceived Non Local Means filter, using Integral Images and the Fast Fourier Transform, named DNLM-IFFT. The deceived non local means filter is part of the Deceived Weighted Averaging Filter Framework (DeWAFF), which defines an approach for image abstraction with a combination of unsharp masking for contrast and edges enhancement and weighted averaging filtering for noise reduction. The proposed optimization approach achieved a speedup factor up to 10.

1 Introduction

Image processing algorithms are commonly used nowadays in different fields, from medicine to sports (Aggarwal and Ryoo [1]; Ekin et al. [6]; Li et al. [8]). For example, in Sáenz et al. [12], the authors present applications of behavior analysis of cells in cancerous tissue, where an automatic cell segmentation and tracking system was proposed to assist researchers to achieve a faster and more accurate understanding of the behavior of a group of cells. The algorithms implemented in such systems for image enhancement, object segmentation and tracking, usually involve high computational cost, frequently making usage of these systems impractical. Hence, it becomes an important issue to optimize the execution time of cell tracking solutions in order to provide researchers in the field an efficient method, as researchers often need to process thousands of images from cell activity. The first stage in the proposed cell tracking pipeline in Sáenz et al. [12] corresponds to the preprocessing of the input image, designed to enhance edges and contrast and also to remove noise, for improving the cell segmentation and tracking stages performance.

This work presents a computational optimization of the Deceived Non Local Means filter (DNLM), a filter proposed in the Deceived Weighted Averaging Filter Framework (DeWAFF) (Calderón et al. [3]), for image enhancement and noise removal. The proposed optimization named as DNLM-IFFT, uses integral

images and the Fast Fourier Transform (FFT), as proposed for the original Non Local Means (NLM) filter in Wang et al. [16]. We present an evaluation of the proposed DNLM-IFFT approach for a speedup assessment. As demonstrated in Calderón et al. [3], the DNLM presented the best results in comparison to the deceived bilateral and the deceived scaled bilateral filters (part of the DeWAFF), as it is based in the NLM noise removal approach (Buades et al. [2]). However, the computational complexity of the DNLM brute force implementation often makes its usage impractical, as researchers frequently need to analyze thousands of images. For this reason, in this paper we focus on the optimization of the DNLM algorithm.

In Sect. 2, we describe important aspects of the DNLM filter. In Sect. 3, we present the proposed computational optimization of the DNLM. Section 5 presents the experiments and test results, comparing our strategy to the previous brute force implementation in order to demonstrate the viability of the proposed modification. Finally in Sect. 6 we present the conclusions and future work related.

2 Background

As presented in Calderón et al. [3], the DeWAFF consists in an image abstraction framework. Image abstraction is defined in Calderón et al. [3] as the enhancement of perceptually important characteristics of the image as contrast and edges, and the simplification of less relevant details for the application domain and noise. As demonstrated in Calderón and Siles [4]; Calderón et al. [3]; Sáenz et al. [12], image abstraction based preprocessing techniques are able to improve the performance of object segmentation, tracking and classification tasks.

The DeWAFF proposes the combination of an Adaptive Unsharp Masking (AUSM) based approach, as the one proposed in Ramponi and Polesel [11], with a weighted averaging filter, like the bilateral filter or the NLM filter. Such combination aims to achieve both contrast and edge enhancement, as also noise suppression. The combination proposed uses the AUSM image output F_{AUSM} as input of the weighted averaging filter algorithm, but uses the original input image U for weighting, *deceiving* the weighted averaging filter. Such combination diminishes the ringing effect, often present in USM filtered images. The following equation presents the general DeWAFF grasp for an input image U of $a \times b = c$ pixels, with $U^a = U$ and $U^b = F_{\text{AUSM}}$ for ringing suppression purposes:

$$U'(p) = \left(\sum_{i \in \Omega_p} w_k(U^a, p, i) \right)^{-1} \left(\sum_{i \in \Omega_p} w_k(U^a, p, i) U^b(i) \right). \quad (1)$$

For notation simplicity, an image pixel is represented as $i = (x_i, y_i)$, and Ω_p stands for the $n \times n$ window centered in pixel p .

The weighting functions studied in Calderón et al. [3] were the Bilateral Filter (BF), the Scaled Bilateral Filter (SBF) and NLM filter:

$$w_1 = w_{\text{BF}}, w_2 = w_{\text{SBF}}, w_3 = w_{\text{NLM}}. \quad (2)$$

The NLM kernel when used in Eq. 1 provides the best results under heavy and moderated noise conditions, given the similarity approach of weighting the neighborhood Φ_i of pixel i with $m \times m$ pixels, instead of using the pixels intensity similarity, as in the bilateral filter and the SBF approaches. The following equation presents the original non local means weighting grasp:

$$w_{\text{NLM}}(p, i) = \exp\left(-\frac{D(i, j)}{h^2}\right), \quad (3)$$

with the Euclidean distance between the neighborhoods given by:

$$D(i, j) = \|\Phi_i - \Phi_j\|^2, \quad (4)$$

where h is the kernel parameter which defines neighborhood weighting discrimination.

In the worst case scenario of using a window of $a \times b = c$ and a neighborhood of $a \times b = c$, pixels, the computational complexity of the NLM is $O(c^3)$, where c corresponds to the total number of pixels in the input image U .

Let us further examine the Euclidean distance between neighborhoods:

$$D(i, j) = \sum_u (\Phi_i(u) - \Phi_j(u))^2, \quad (5)$$

where \sum_u stands for the summation of the pixels within the difference of the two neighborhoods Φ_i and Φ_j .

Developing the squared difference and rewriting the quadratic terms as $\Phi_i^2 = \sum_u \Phi_i(u)^2$, we get:

$$D(i, j) = \Phi_i^2 - 2 \sum_u \Phi_j(u) \Phi_i(u) + \Phi_j^2 = \Phi_i^2 - 2\Phi_j \cdot \Phi_i + \Phi_j^2, \quad (6)$$

where $\Phi_j \cdot \Phi_i$ refers to the matrix dot product. The following example illustrates the calculation of the Euclidean distance between neighborhoods with an input image $U \in \mathbb{R}^{5 \times 5}$ defined in Table 1.

Table 1. Image example U .

	1	2	3	4	5
1	5	12	1	3	2
2	5	2	3	1	4
3	3	1	2	3	1
4	4	3	2	1	3
5	1	5	6	5	3

For such illustration purposes, we take a window $\Omega_{(3,3)} \in \mathbb{R}^{3 \times 3}$ and perform the calculation of the Euclidean distance between the neighborhoods $\Phi_{(3,3)}$ and $\Phi_{(2,2)}$:

$$\|\Phi_{(3,3)} - \Phi_{(2,2)}\|^2 = \left\| \begin{pmatrix} 2 & 3 & 1 & 5 & 12 & 1 \\ 1 & 2 & 3 & -5 & 2 & 3 \\ 3 & 2 & 1 & 3 & 1 & 2 \end{pmatrix} \right\|^2 = 10.3923^2 = 108. \quad (7)$$

Given the development of the quadratic difference between neighborhoods in the right part of Eq. 6, we can perform such calculation based in the element wise squared matrix of $U \in \mathbb{R}^{5 \times 5}$. This allows calculating the Euclidean distance done previously equivalent to perform:

$$\|\Phi_{(3,3)} - \Phi_{(2,2)}\|^2 = \Phi_{(3,3)}^2 - 2\Phi_{(2,2)} \cdot \Phi_{(3,3)} + \Phi_{(2,2)}^2 \quad (8)$$

$$= \sum \begin{pmatrix} 4 & 9 & 1 \\ 1 & 4 & 9 \\ 9 & 4 & 1 \end{pmatrix} - 2 \begin{pmatrix} 2 & 3 & 1 & 5 & 12 & 1 \\ 1 & 2 & 3 & -5 & 2 & 3 \\ 3 & 2 & 1 & 3 & 1 & 2 \end{pmatrix} + \sum \begin{pmatrix} 25 & 144 & 1 \\ 25 & 4 & 9 \\ 9 & 1 & 4 \end{pmatrix} \quad (9)$$

$$= 42 + -2 \cdot 78 + 222 = 42 - 156 + 222 = 108, \quad (10)$$

consistent with the result obtained in Eq. 7.

3 Proposed Method: DNLM-IFFT

As seen in previous section, the computational cost of the brute force implementation of the DNLM is very high. To address such problem, there are different approaches which have been designed for the original NLM algorithm, and could be ported to the DNLM in order to lower computational complexity, as seen in Karnati et al. [7]; Liu et al. [9]; Wang et al. [16]. However, approximations or modifications in the computation of the NLM kernel must allow the decoupling of the weighted image U^a and the filtered image U^b , for a DNLM implementation, as stated in Eq. 1. Also, we prefer an optimization of the NLM over approximations as seen in Dauwe et al. [5]; Vignesh et al. [14]; Xue et al. [17].

In Wang et al. [16], an efficient computation of the weighting function $w_{\text{NLM}}(p, i)$ is proposed using integral images and the FFT. The approach achieves same numeric results compared to the brute force implementation. The optimization proposed in Wang et al. [16], modifies the computation of the weighting kernel, by performing the calculation of the Euclidean distance defined in Eq. 6 as follows:

- Use of integral images to calculate the terms Φ_i^2 y Φ_j^2 .
- Calculate the FFT to obtain the term $-2\Phi_j \cdot \Phi_i$ which corresponds to a sample of the auto-correlation with the signal Φ_j .

3.1 Integral Images

The integral image of I as defined in Viola and Jones [15], with a pair wise pixel notation $i = (x, y)$, is given by:

$$I_{\Sigma}(x, y) = \sum_{u \leq x, v \leq y} I(u, v), \quad (11)$$

Table 2. Integral image for U^2 .

	1	2	3	4	5
1	25(A)	169	170	179(B)	183
2	50	198	208	218	238
3	59	208	222	241	262
4	75(C)	233	251	271(D)	301
5	76	259	313	358	397

which is the summation of the pixels to the left and above of the parameter pixel $i = (x, y)$. Following the example exposed previously, the integral image of the pixel wise squared matrix of U is calculated as shown in Table 2. For a more efficient calculation of an integral image, it is worth to note that:

$$I_{\Sigma}(x, y) = I(x, y) - I_{\Sigma}(x - 1, y - 1) + I_{\Sigma}(x, y - 1) + I_{\Sigma}(x - 1, y), \quad (12)$$

which for instance, for the previous example means that:

$$I_{\Sigma}(3, 3) = 4 - 198 + 208 + 208 = 222. \quad (13)$$

To use the integral image to calculate the summation over a window limited by pixels $A = (x_0, y_0)$, $B = (x_1, y_0)$, $C = (x_0, y_1)$ and $D = (x_1, y_1)$, as illustrated in Table 2, we compute the following in general:

$$\sum_{x_0 < x \leq x_1, y_0 < y \leq y_1} I(x, y) = I(D) + I(A) - I(B) - I(C), \quad (14)$$

and for the example developed in this paper, for a window of dimensions 3×3 around the pixel $(3, 3)$ the corner pixels would be given as, $A = (1, 1)$, $B = (4, 1)$, $C = (1, 4)$ y $D = (4, 4)$, as specified in Table 2. Thus, the calculation of the summation of the pixels over such window is calculated as follows:

$$\sum_{x_0 < x \leq x_1, y_0 < y \leq y_1} I(3, 3) = 271 + 25 - 179 - 75 = 42, \quad (15)$$

which is consistent with the first term in the Eq. 10. Using the integral image to calculate the summation over a window of an input matrix with c pixels presents a computational cost of $O(c)$, and must be made only once for the NLM filtering.

4 Correlation and the Fast Fourier Transform

As stated previously, the term in the quadratic equation of the Euclidean distance between the neighborhoods of pixels i and j :

$$S(i, j) = N_i^2 - 2 \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} N_j(a, b) N_i(a, b) + N_j^2 = N_i^2 - 2N_j \cdot N_i + N_j^2, \quad (16)$$

corresponding to the dot product of matrices $-2N_j \cdot N_i$, can be extracted by using the correlation of the neighborhood from pixel i , N_i . We can compute the dot product term using the FFT with a logarithmic computational complexity, lowering the overall DNLM computational cost.

5 Experiments and Results

We tested the computational execution time of the algorithm using images with resolutions of $480p$, $720p$ and $1080p$ executed 10 times each, for a typical window size of 21×21 and a neighborhood size of 7×7 , as recommended in Baudes et al. [2]. The results are shown in Table 4. The second experiment consists in evaluating the speedup by using different window and neighborhood sizes in an image of 1200×900 pixels. Its results are shown in Table 3.

Table 3. Time in seconds achieved with different image resolutions for the DNLM with a brute force implementation and the DNLM-IFFT.

Window size	Neighborhood size	DNLM	DNLM-IFFT	Speedup
21	7	566.21	51.90	10.91
21	15	652.97	72.60	8.99
21	21	707.81	184.44	3.84
43	7	2211.40	107.57	20.55
43	15	2644.30	388.02	6.81
43	21	2881.34	555.40	5.19
55	7	3548.80	328.17	10.81
55	15	4271.78	561.05	7.61
55	21	4504.16	298.80	15.07

All tests were executed in a desktop computer with an AMD Phenom FX-6300 processor at 4.6 GHz. Average values for these executions are displayed in Table 4.

Table 4. Time in seconds achieved with different image resolutions for the DNLM with a brute force implementation and the DNLM-IFFT.

Image size	DNLM	DNLM-IFFT	Speedup
480	156.3	16.3	9.6
720	480.3	47.9	10.0
1080	1093.6	110.3	9.9

6 Conclusions and Future Work

The proposed DNLM-IFFT approach achieves an effective speed-up up to an average factor of 10, as seen in Table 4. The results of the experiment according to Table 3, suggests that the DNLM-IFFT becomes more attractive with smaller neighborhood sizes, given a smaller speedup with bigger neighborhoods. For pre-processing low resolution video footages, execution time is reasonable, however, biomedical images and video analysis often require the analysis of thousands of images. As already mentioned, microbiology researchers from the University of Costa Rica need to analyze 170 000 images from glioblastoma tissue. This means that the preprocessing stage with the proposed DNLM-IFFT could take months, which is not appropriate.

As future work, we expect to explore alternative parallelization technologies, such as MPI and OpenMP for a cluster based on the Intel KNL architecture. In Cuomo [13], a GPU based parallelization approach for the Non Local Means filter was proposed, which suggests that a GPU based implementation of the DNLM might achieve greater performance gain when compared with a CPU based parallelization approach. Also, we can consider mixing the DNLM-IFFT proposed approach with other approximation grasps of the NLM, for instance using look-up tables, as seen in Mahmoudi and Sapiro [10].

The MATLAB code implemented for this paper can be downloaded from <https://github.com/manu3193/DNLM-IFFTT>.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. *J. ACM Comput. Surv. CSUR* **43**, 16 (2011)
2. Buades, A., Coll, B., Morel, J.M.: Neighborhood filters and PDEs. *Numer. Math.* **105**(1), 1–34 (2006)
3. Calderón, S., Sáenz, A., Mora, R., Siles, F., Orozco, I., Buemi, M.: Dewaff: a novel image abstraction approach to improve the performance of a cell tracking system. In: 2015 4th International Work Conference on Bioinspired Intelligence (IWOB), pp. 81–88. IEEE (2015)
4. Calderón, S., Siles, F.: Deceived bilateral filter for improving the classification of football players from TV broadcast. In: IEEE 3rd International Conference and Workshop on Bioinspired Intelligence (2014)
5. Dauwe, A., Goossens, B., Luong, H.Q., Philips, W.: A fast non-local image denoising algorithm. In: *Electronic Imaging 2008*, p. 681210. International Society for Optics and Photonics (2008)
6. Ekin, A., Tekalp, A.M., Mehrotra, R.: Automatic soccer video analysis and summarization. *IEEE Trans. Image Process.* **12**(7), 796–807 (2003)
7. Karnati, V., Uliyar, M., Dey, S.: Fast non-local algorithm for image denoising. In: 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 3873–3876. IEEE (2009)
8. Li, K., Miller, E.D., Chen, M., Kanade, T., Weiss, L.E., Campbell, P.G.: Computer vision tracking of stemness. In: 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2008, pp. 847–850. IEEE (2008)

9. Liu, Y.L., Wang, J., Chen, X., Guo, Y.W., Peng, Q.S.: A robust and fast non-local means algorithm for image denoising. *J. Comput. Sci. Technol.* **23**(2), 270–279 (2008)
10. Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Sig. Process. Lett.* **12**(12), 839–842 (2005)
11. Ramponi, G., Polesel, A.: Rational unsharp masking technique. *J. Electron. Imaging* **7**(2), 333–338 (1998)
12. Sáenz, A., Calderón, S., Castro, J., Mora, R., Siles, F.: Deceived bilateral filter for improving the automatic cell segmentation and tracking in the NF-kB pathway without nuclear staining. In: Braidot, A., Hadad, A. (eds.) VI Latin American Congress on Biomedical Engineering CLAIB 2014. IFMBE, vol. 49, pp. 345–348. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-319-13117-7_89
13. Cuomo, S., De Michele, P., Piccialli, F.: 3D data denoising via nonlocal means filter by using parallel GPU strategies. *Comput. Math. Methods Med.* **2014**, Article no. 523862 (2014)
14. Vignesh, R., Oh, B.T., Kuo, C.C.J.: Fast non-local means (NLM) computation with probabilistic early termination. *IEEE Sig. Process. Lett.* **17**(3), 277–280 (2010)
15. Viola, P., Jones, M.: Robust real-time object detection. *Int. J. Comput. Vis.* **4**(34–47) (2001)
16. Wang, J., Guo, Y., Ying, Y., Liu, Y., Peng, Q.: Fast non-local algorithm for image denoising. In: 2006 IEEE International Conference on Image Processing, pp. 1429–1432. IEEE (2006)
17. Xue, B., Huang, Y., Yang, J., Shi, L., Zhan, Y., Cao, X.: Fast nonlocal remote sensing image denoising using cosine integral images. *IEEE Geosci. Remote Sens. Lett.* **10**(6), 1309–1313 (2013)