

Non-Local Euclidean Medians

Kunal N. Chaudhury* Amit Singer**

Abstract—In this letter, we note that the denoising performance of Non-Local Means (NLM) can be improved at large noise levels by replacing the mean by the Euclidean median. We call this new denoising algorithm the Non-Local Euclidean Medians (NLEM). At the heart of NLEM is the observation that the median is more robust to outliers than the mean. In particular, we provide a simple geometric insight that explains why NLEM performs better than NLM in the vicinity of edges, particularly at large noise levels. NLEM can be efficiently implemented using iteratively reweighted least squares, and its computational complexity is comparable to that of NLM. We provide some preliminary results to study the proposed algorithm and to compare it with NLM.

Index Terms—Image denoising, non-local means, Euclidean median, iteratively reweighted least squares (IRLS), Weiszfeld algorithm.

I. INTRODUCTION

Non-Local Means (NLM) is a data-driven diffusion mechanism that was introduced by Buades et al. in [1]. It has proved to be a simple yet powerful method for image denoising. In this method, a given pixel is denoised using a weighted average of other pixels in the (noisy) image. In particular, given a noisy image $u = (u_i)$, the denoised image $\hat{u} = (\hat{u}_i)$ at pixel i is computed using the formula

$$\hat{u}_i = \frac{\sum_j w_{ij} u_j}{\sum_j w_{ij}}, \quad (1)$$

where w_{ij} is some weight (affinity) assigned to pixels i and j . The sum in (1) is ideally performed over the whole image. In practice, however, one restricts j to a geometric neighborhood of i , e.g., to a sufficiently large window of size $S \times S$ [1].

The central idea in [1] was to set the weights using image patches centered around each pixel, namely as

$$w_{ij} = \exp\left(-\frac{1}{h^2} \|\mathbf{P}_i - \mathbf{P}_j\|^2\right), \quad (2)$$

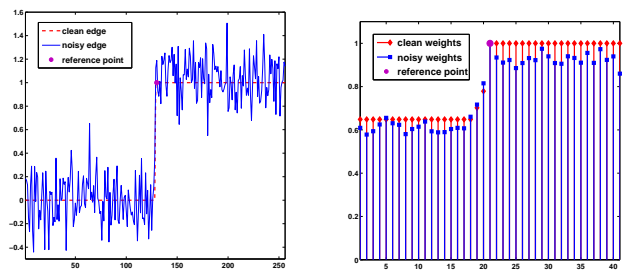
where \mathbf{P}_i and \mathbf{P}_j are the image patches of size $k \times k$ centered at pixels i and j . Here, $\|\mathbf{P}\|$ is the Euclidean norm of patch \mathbf{P} as a point in \mathbf{R}^{k^2} , and h is a smoothing parameter. Thus, pixels with similar neighborhoods are given larger weights compared to pixels whose neighborhoods look different. The algorithm makes explicit use of the fact that repetitive patterns appear in most natural images. It is remarkable that the simple formula

in (1) often provides state-of-the-art results in image denoising [2]. One outstanding feature of NLM is that, in comparison to other denoising techniques such as Gaussian smoothing, anisotropic diffusion, total variation denoising, and wavelet regularization, the so-called method noise (difference of the denoised and the noisy image) in NLM appears more like white noise [1], [2]. We refer the reader to [2] for a detailed review of the algorithm.

The rest of the letter is organized as follows. In Section II, we explain how the denoising performance of NLM can be improved in the vicinity of edges using the Euclidean median. Based on this observation, we propose a new denoising algorithm in Section III and discuss its implementation. This is followed by some preliminary denoising results in Section IV, where we compare our new algorithm with NLM. We conclude with some remarks in Section V.

II. BETTER ROBUSTNESS USING EUCLIDEAN MEDIAN

The denoising performance of NLM depends entirely on the reliability of the weights w_{ij} . The weights are, however, computed from the noisy image and not the clean image. Noise affects the distribution of weights, particularly when the noise is large. By noise, we will always mean zero-centered white Gaussian noise with variance σ^2 in the rest of the discussion.



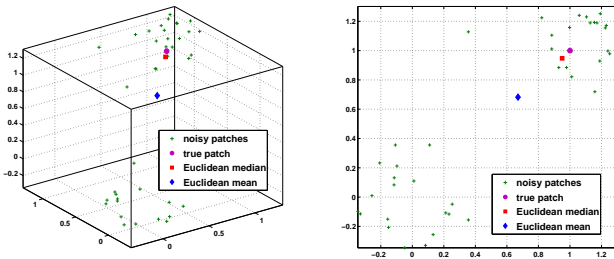
(a) Clean and noisy edge. (b) Weights.
Fig. 1. Ideal edge in one dimension.

To understand the effect of noise and to motivate the main idea of the paper, we perform a simple experiment. We consider the particular case where the pixel of interest is close to an (ideal) edge. For this, we take a clean edge profile of unit height and add noise ($\sigma = 0.2$) to it. This is shown in Fig. 1. We now select a point of interest a few pixels to the right of the edge (marked with a purple circle). The goal is to estimate its true value from its neighboring points using NLM. To do so, we take 3-sample patches around each point ($k = 3$), and a search window of $S = 41$. The patches are shown as points in 3-dimensions in Fig 2a. The clean patch for the point of interest is at $(1, 1, 1)$.

*Program in Applied and Computational Mathematics (PACM), Princeton University, Princeton, NJ 08544, USA (kchaudhu@math.princeton.edu). K. Chaudhury was partially supported by the Swiss National Science Foundation under grant PBELP2-135867.

**PACM and Department of Mathematics, Princeton University, Princeton, NJ 08544, USA (amits@math.princeton.edu). A. Singer was partially supported by Award Number DMS-0914892 from the NSF, by Award Number FA9550-09-1-0551 from AFOSR, by Award Number R01GM090200 from the National Institute of General Medical Sciences, and by the Alfred P. Sloan Foundation.

We use (2) to compute the weights, where we set $h = 10\sigma$. The weights corresponding to the point of interest are shown in Fig. 1b. Using the noisy weights, we obtain an estimate of around 0.65. This estimate has a geometric interpretation. It is the center coordinate of the Euclidean mean $\sum_j w_j \mathbf{P}_j / \sum_j w_j$, where w_j are the weights in Fig. 1b, and \mathbf{P}_j are the patches in Fig. 2a. The Euclidean mean is marked with a blue diamond in Fig. 2. Note that the patches drawn from the search window are clustered around the centers $A = (0, 0, 0)$ and $B = (1, 1, 1)$. For the point of interest, the points around A are the outliers, while the ones around B are the inliers. The noise reduces the relative gap in the corresponding weights, causing the mean to drift away from the inliers toward the outliers.



(a) 3d patch space close to the edge. (b) 2d projection (first 2 coordinates).

Fig. 2. Outlier model of the patch space for the point of interest in Fig. 1a. Due to its robustness to outliers, the Euclidean median behaves as a better estimator of the clean patch than the Euclidean mean (see text for explanation).

Note that the Euclidean mean is the minimizer of $\sum_j w_j \|\mathbf{P} - \mathbf{P}_j\|^2$ over all patches \mathbf{P} . Our main observation is that, if we instead compute the minimizer of $\sum_j w_j \|\mathbf{P} - \mathbf{P}_j\|$ over all \mathbf{P} , and take the center coordinate of \mathbf{P} , then we get a much better estimate. Indeed, the denoised value turns out to be around 0.92 in this case. The above minimizer is called the Euclidean median (or the geometric median) in the literature [3]. We will often simply call it the median. We repeated the above experiment using several noise realizations, and consistently got better results using the median. Averaged over 10 trials, the denoised value using the mean and median were found to be 0.62 and 0.93, respectively. Indeed, in Fig. 2, note how close the median is to the true patch compared to the mean. This does not come as a surprise since it is well-known that the median is more robust to outliers than the mean. This fact has a rather simple explanation in one dimension. In higher dimensions, note that the former is the minimizer of the weighted ℓ^2 norm of the distances $\|\mathbf{P} - \mathbf{P}_j\|$, while the latter is the minimizer of the weighted ℓ^1 norm of these distances. It is this use of the ℓ^1 norm over the ℓ^2 norm that makes the Euclidean median more robust to outliers [3].

III. NON-LOCAL EUCLIDEAN MEDIANS

Following the above observation, we propose Algorithm 1 below which we call the Non-Local Euclidean Medians (NLEM). We use $S(i)$ to denote the search window of size $S \times S$ centered at pixel i .

Algorithm 1 Non-Local Euclidean Medians

Input: Noisy image $u = (u_i)$, and parameters h, S, k .

Return: Denoised image $\hat{u} = (\hat{u}_i)$.

(1) Extract patch \mathbf{P}_i of size $k \times k$ at every pixel i .

(2) For every pixel i , do

(a) Set $w_{ij} = \exp(-\|\mathbf{P}_i - \mathbf{P}_j\|^2/h^2)$ for every $j \in S(i)$.

(b) Find patch \mathbf{P} that minimizes $\sum_{j \in S(i)} w_{ij} \|\mathbf{P} - \mathbf{P}_j\|$.

(c) Assign \hat{u}_i the value of the center pixel in \mathbf{P} .

The difference with NLM is in step 2b which involves the computation of the Euclidean median. That is, given points $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbf{R}^d$ and weights w_1, \dots, w_n , we are required to find $\mathbf{x} \in \mathbf{R}^d$ that minimizes the convex cost $\sum_{j=1}^n w_j \|\mathbf{x} - \mathbf{x}_j\|$. There exists an extensive literature on the computation of the Euclidean median; cf. [4], [5] and references therein. The simplest algorithm in this area is the so-called Weiszfeld algorithm [4], [5]. This is, in fact, based on the method of iteratively reweighted least squares (IRLS), which has received renewed interest in the compressed sensing community in the context of ℓ^1 minimization [6], [7]. Starting from an estimate $\mathbf{x}^{(k)}$, the idea is to set the next iterate as

$$\mathbf{x}^{(k+1)} = \arg \min_{\mathbf{x} \in \mathbf{R}^d} \sum_{j=1}^n w_j \frac{\|\mathbf{x} - \mathbf{x}_j\|^2}{\|\mathbf{x}^{(k)} - \mathbf{x}_j\|}.$$

This is a least-squares problem, and the minimizer is given by

$$\mathbf{x}^{(k+1)} = \frac{\sum_j \mu_j^{(k)} \mathbf{x}_j}{\sum_j \mu_j^{(k)}}, \quad (3)$$

where $\mu_j^{(k)} = w_j / \|\mathbf{x}^{(k)} - \mathbf{x}_j\|$. Starting with an initial guess, one keeps repeating this process until convergence. In practice, one needs to address the situation when $\mathbf{x}^{(k)}$ gets close to some \mathbf{x}_j , which causes $\mu_j^{(k)}$ to blow up. In the Weiszfeld algorithm, one keeps track of the proximity of $\mathbf{x}^{(k)}$ to all the \mathbf{x}_j , and $\mathbf{x}^{(k+1)}$ is set to be \mathbf{x}_i if $\|\mathbf{x}^{(k)} - \mathbf{x}_i\| < \varepsilon$ for some i . It has been proved by many authors that the iterates converge globally, and even linearly (exponentially fast) under certain conditions, e.g., see discussion in [5].

Following the recent ideas in [6], [7], we have also tried regularizing (3) by adding a small bias, namely by setting $\mu_j^{(k)} = w_j / (\|\mathbf{x}^{(k)} - \mathbf{x}_j\|^2 + \varepsilon_k^2)^{1/2}$. The bias ε_k is updated at each iterate, e.g., one starts with $\varepsilon_0 = 1$ and gradually shrinks ε_k to zero as the iteration progresses. The convergence properties of a particular flavor of this algorithm are discussed in [7]. We have tried both the Weiszfeld algorithm and the one in [6]. Experiments show us that faster convergence is obtained using the latter, typically within 3 to 4 steps. The overall computational complexity of NLEM is $O(k^2 S^2 N_{\text{iter}})$ per pixel, where N_{iter} is the average number of IRLS iterations. The complexity of the standard NLM is $O(k^2 S^2)$ per pixel. For example, for a 256×256 image, the typical run time of a C implementation of NLM ($S = 21, k = 7$) was about 10 seconds, and that for NLEM was about 32 seconds (on an Intel quad-core 2.83 GHz machine).

TABLE I
COMPARISON OF NLM AND NLEM IN TERMS OF PSNR AND SSIM, AT NOISE LEVELS $\sigma = 10, 20, \dots, 100$ (AVERAGED OVER 10 NOISE REALIZATIONS). SAME PARAMETERS USED: $S = 21, k = 7$, AND $h = 10\sigma$.

Image	Method	PSNR (dB)									
		<i>Checker</i>	NLM	40.04	35.16	31.74	27.84	25.21	23.13	21.39	19.96
	NLEM	39.73	34.66	31.66	29.37	26.71	24.76	23.22	21.82	20.50	19.45
<i>Circles</i>	NLM	37.31	34.67	31.79	28.82	26.46	24.69	23.10	21.58	20.23	19.03
	NLEM	34.27	34.08	32.33	30.05	27.92	26.24	24.87	23.57	22.36	21.16
<i>House</i>	NLM	34.22	29.78	26.88	25.21	24.07	23.37	22.78	22.38	22.06	21.81
	NLEM	33.96	30.10	27.15	25.39	24.30	23.51	22.96	22.54	22.17	21.95
<i>Barbara</i>	NLM	32.37	27.39	24.93	23.52	22.64	22.04	21.62	21.29	21.07	20.88
	NLEM	32.11	27.75	25.26	23.84	22.90	22.29	21.83	21.48	21.20	21.01
<i>Lena</i>	NLM	33.24	29.31	27.40	26.16	25.24	24.54	24.04	23.66	23.34	23.06
	NLEM	33.15	29.45	27.61	26.40	25.53	24.84	24.31	23.90	23.53	23.24
		SSIM (%)									
<i>Checker</i>	NLM	99.41	98.66	97.59	95.51	92.32	88.35	83.37	78.05	72.31	66.47
	NLEM	99.36	98.51	97.60	96.30	94.03	91.18	87.81	84.04	79.59	74.51
<i>Circles</i>	NLM	96.31	94.02	91.43	88.66	85.79	82.49	78.28	73.28	68.13	63.54
	NLEM	95.28	93.60	91.20	88.74	86.30	83.75	80.90	77.85	74.21	70.21
<i>House</i>	NLM	86.90	81.80	76.93	72.97	69.57	66.86	64.38	62.24	60.33	58.55
	NLEM	86.86	81.25	77.61	73.85	70.51	67.77	65.21	62.96	60.97	59.03
<i>Barbara</i>	NLM	89.95	79.48	71.32	65.20	60.79	57.42	54.64	52.58	50.78	49.27
	NLEM	90.25	80.38	72.49	66.48	62.04	58.57	55.64	53.45	51.50	49.86
<i>Lena</i>	NLM	86.95	80.41	76.45	73.42	70.80	68.45	66.46	64.66	62.94	61.32
	NLEM	87.06	80.57	76.77	73.97	71.50	69.21	67.20	65.32	63.52	61.81

IV. EXPERIMENTS

We now present the result of some preliminary denoising experiments. For this, we have used the synthetic images shown in Fig. 3 and some natural images. For NLEM, we computed the Euclidean median using the simple IRLS scheme described in [6]. For all experiments, we have set $S = 21, k = 7$, and $h = 10\sigma$ for both algorithms. These are the standard settings originally proposed in [1].

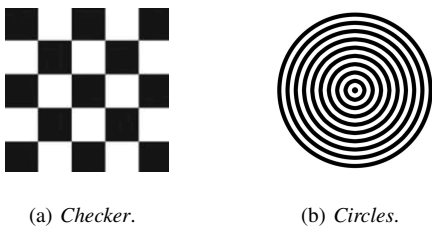
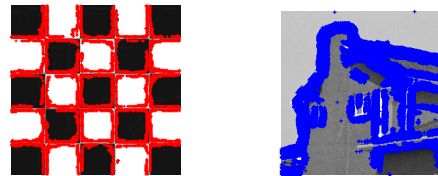


Fig. 3. Synthetic grayscale test images, original size 256×256 . Black is at intensity 0, and white is at intensity 255.

First, we considered the *Checker* image. We added noise with $\sigma = 100$ resulting in a Peak-Signal-to-Noise Ratio (PSNR) of 8.18dB. The PSNR improved to 17.94dB after applying NLM, and with NLEM this further went up to 19.45dB (averaged over 10 noise realizations). This 1.5dB improvement is perfectly explained by the arguments provided in Section II. Indeed, in Fig.4a, we have marked those pixels where the estimate from NLEM is significantly closer to the clean image than that obtained using NLM. More precisely, denote the original image by f_i , the noisy image by u_i , and the denoised images from NLM and NLEM by \hat{u}_i and \hat{u}'_i . Then the “+” in the figure denotes pixels i where $|\hat{u}'_i - f_i| < |\hat{u}_i - f_i| - 10$. Note that these points are concentrated around the edges where the median performs better than the mean.

So what happens when we change the noise level? We



(a) *Checker* ($\sigma=100$).

(b) *House* ($\sigma=60$).

Fig. 4. The “+” symbol marks pixels where the estimate returned by NLEM is significantly better than that returned by NLM.

went from $\sigma = 10$ to $\sigma = 100$ in steps of 10. The plots of the corresponding PSNRs are shown in Fig. 5a. At low noise levels ($\sigma < 30$), we see that NLM performs as good or even better than NLEM. This is because at low noise levels the true neighbors in patch space are well identified, at least in the smooth regions. The difference between them is then mainly due to noise, and since the noise is Gaussian, the least-squares approach in NLM gives statistically optimal results in the smooth regions. On the other hand, at low noise levels, the two clusters in Fig. 2 are well separated and hence the weights w_{ij} for NLM are good enough to push the mean towards the right cluster. The median and mean in this case are thus very close in the vicinity of edges. At higher noise levels, the situation completely reverses and NLEM performs consistently better than NLM. In Fig. 5a, we see that this phase transition occurs at around $\sigma = 30$. The improvement in PSNR is quite significant beyond this noise level, as large as 2.1dB. The exact PSNRs are given in Table I.

Next, we tried the above experiment on the *Circles* image. The results are given in Table I. The phase transition in this case occurs around $\sigma = 25$. As in the previous case, NLEM performs better beyond the phase transition, and the gain in PSNR over NLM is as large as 2.2dB. The method

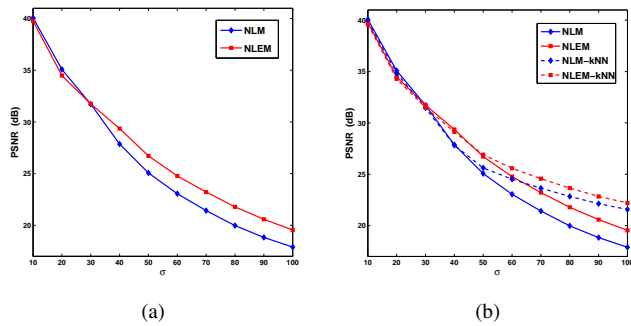


Fig. 5. Comparison of denoising performance for the *Checker* image at noise $\sigma = 80$. NLM-kNN and NLEM-kNN refer to the respective modifications of NLM and NLEM where we use only the largest 50% of the weights.

noise for NLM and NLEM obtained from a typical denoising experiment are shown in Fig. 6. Note that the method noise for NLEM appears more white (with less structural features) than that for NLM.

Finally, we considered some benchmark natural images, namely *House*, *Barbara*, and *Lena*. The PSNRs obtained from NLM and NLEM for these images at different noise levels are shown in Table I. The table also compares the Structural Similarity (SSIM) indices [12] obtained from the two methods. Note that a phase transition in SSIM is also observed for some of the images. In Fig. 4b, we show the pixels where NLEM does better (in the sense defined earlier) than NLM for the *House* image at $\sigma = 60$. We again see that these pixels are concentrated around the edges.

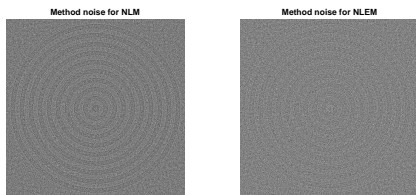


Fig. 6. Comparison of the method noise for the *Circles* image at $\sigma = 80$.

The improvements in PSNR and SSIM are quite dramatic for the synthetic images *Checker* and *Circles*. This is expected because they contain many edges and the edges have large transitions. The improvement in PSNR and SSIM are less dramatic for the natural images. But, considering that NLM already provides top quality denoising results, this small improvement is already significant. We have also noticed that the performance of NLEM (and that of NLM) can be further improved by considering only the top 50% of the S^2 -neighbors with largest weights in step 2b of Algorithm 1. This simple modification improves both NLM and NLEM (and also their running time), while still maintaining their order in terms of PSNR performance. A comparison of the four different methods is given in Fig. 5b. While we have considered a fixed setting for the parameters, our general observation based on extensive experiments is that NLEM consistently performs better than NLM beyond the phase transition, irrespective of the parameter settings. In future, we plan to investigate

ways of further improving NLEM, and study the effect of the parameters on its denoising performance.

V. DISCUSSION

The purpose of this note was to communicate the idea that one can improve (often substantially) the denoising results of NLM by replacing the ℓ^2 regression on patch space by the more robust ℓ^1 regression. This led us to propose the NLEM algorithm. The experimental results presented in this paper reveal two facts: (a) Phase transition phenomena – NLEM starts to perform better (in terms of PSNR) beyond a certain noise level, and (b) The bulk of the improvement comes from pixels close to sharp edges. The latter fact indicates that NLEM is better suited for denoising images that have many sharp edges. This suggests that we could get similar PSNR improvements if we simply used NLEM in the vicinity of edges and the less expensive NLM elsewhere. Unfortunately, it is difficult to get a reliable edge map from the noisy image. On the other hand, observation (b) suggests that, by comparing the denoising results of NLM and NLEM, one can devise a robust method of detecting edges at large noise levels. We note that Wang et al. have recently proposed a non-local median-based estimator in [9]. This can be considered as a special case of NLEM corresponding to $k = 1$, where single pixels (instead of patches) are used for the median computation. On the other hand, some authors have proposed median-based estimators for NLM where the noisy image is median filtered before computing the weights [8], [9]. In fact, most of the recent innovations in NLM were concerned with better ways of computing the weights; e.g., see [10], [11], and reference therein. It would be interesting to see if the idea of ℓ^1 regression could be used to complement these innovations.

REFERENCES

- [1] A. Buades, B. Coll, J. M. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling and Simulation*, vol. 4, pp. 490-530, 2005.
- [2] A. Buades, B. Coll, J. M. Morel, “Image Denoising Methods. A New Nonlocal Principle,” *SIAM Review*, vol. 52, pp. 113-147, 2010.
- [3] P. J. Huber, E. M. Ronchetti, *Robust Statistics*, Wiley Series in Probability and Statistics, Wiley, 2009.
- [4] E. Weiszfeld, “Sur le point par lequel le somme des distances de n points donnees est minimum,” *Tôhoku Math. J.*, vol. 43, pp. 355-386, 1937.
- [5] G. Xue, Y. Ye, “An efficient algorithm for minimizing a sum of Euclidean norms with applications,” *SIAM Journal on Optimization*, vol. 7, pp. 1017-1036, 1997.
- [6] R. Chartrand, Y. Wotao, “Iteratively reweighted algorithms for compressive sensing,” *IEEE ICASSP*, pp. 3869-3872, 2008.
- [7] I. Daubechies, R. Devore, M. Fornasier, C. S. Gunturk, “Iteratively reweighted least squares minimization for sparse recovery,” *Communications on Pure and Applied Mathematics*, vol. 63, pp. 1-38, 2009.
- [8] C. Chung, R. Fulton, D. D. Feng, S. Meikle, “Median non-local means filtering for low SNR image denoising: Application to PET with anatomical knowledge,” *IEEE Nuclear Science Symposium Conference Record*, pp. 3613-3618, 2010.
- [9] Y. Wang, A. Szlam, G. Lerman, “Robust locally linear analysis with applications to image denoising and blind inpainting,” submitted.
- [10] T. Tasdizen, “Principal neighborhood dictionaries for non-local means image denoising,” *IEEE Trans. Image Processing*, vol. 18, pp. 2649-2660, 2009.
- [11] D. Van De Ville, M. Kocher, “Nonlocal means with dimensionality reduction and SURE-based parameter selection,” *IEEE Trans. Image Processing*, vol. 20, pp. 2683-2690, 2011.
- [12] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.