Non-Local Patch Regression: Robust Image Denoising in Patch Space

Kunal N. Chaudhury** Amit Singer*

***Abstract*—It was recently demonstrated in [13] that the denoising performance of Non-Local Means (NLM) can be improved at large noise levels by replacing the mean by the robust Euclidean median. Numerical experiments on synthetic and natural images showed that the latter consistently performed better than NLM beyond a certain noise level, and significantly so for images with sharp edges. The Euclidean mean and median can be put into a common regression (on the patch space) framework, in which the ℓ^2 norm of the residuals is considered in the former, while the ℓ^1 norm is considered in the latter. The natural question then is what happens if we consider ℓ^p $(0 < p < 1)$ regression? We investigate this possibility in this paper.**

***Index Terms*—Image denoising, non-local means, non-local Euclidean medians, edges, inlier-outlier model, robustness, sparsity, non-convex optimization, iteratively reweighted least squares.**

I. Introduction

In the last few years, some very effective frameworks for image restoration have been proposed that exploit non-locality (long-distance correlations) in images, and/or use patches instead of pixels to robustly compare photometric similarities. The archetype algorithm in this regard is the Non-Local Means (NLM) [1]. The success of NLM triggered a huge amount of research, leading to state-of-the-art algorithms that exploit non-locality and/or the patch model in specialized ways; e.g., see [3], [4], [9], [5], [6], [7], [8], to name a few. We refer the interested reader to [2], [7] for detailed reviews. Of these, the best performing method till date is perhaps the hybrid BM3D algorithm [9], which effectively combines the NLM framework with other classical algorithms.

To setup notations, we recall the working of NLM. Let $u = (u_i)$ be some linear indexing of the input noisy image. The standard setting is that u is the corrupted version of some clean image $f = (f_i)$,

$$u_i = f_i + \sigma z_i, \tag{1}$$

where (z_i) is iid $\mathcal{N}(0,1)$. The goal is to estimate (approximate) f from the noisy measurement u, possibly given a good estimate of the noise floor σ. In NLM, the restored image $\hat{u} = (\hat{u}_i)$ is computed using the simple formula

$$\hat{u}_i = \frac{\sum_{j \in S(i)} w_{ij} u_j}{\sum_{j \in S(i)} w_{ij}}, \tag{2}$$

where w_{ij} is some weight (affinity) assigned to pixels i and j. Here $S(i)$ is the neighborhood of pixel i over which the averaging is performed. To exploit non-local correlations, $S(i)$ is ideally set to the whole image domain. In practice, however, one restricts $S(i)$ to a geometric neighborhood, e.g., to a sufficiently large window of size $S \times S$ around i [1]. The other idea in NLM is to set the weights using image patches centered around each pixel. In particular, for a given pixel i, let $\mathbf{P}_i$ denote the restriction of u to a square window around i. Letting k be the length of this window, this associates every pixel i with a point $\mathbf{P}_i$ in $\mathbf{R}^{k^2}$ (the patch space). The weights in standard NLM are set to be

$$w_{ij} = \exp\Big(-\frac{1}{h^2}\|\mathbf{P}_i - \mathbf{P}_j\|^2\Big), \tag{3}$$

where $\|\mathbf{P}_i - \mathbf{P}_j\|$ is the Euclidean distance between $\mathbf{P}_i$ and $\mathbf{P}_j$ as points in $\mathbf{R}^{k^2}$, and h is a smoothing parameter. Along with non-locality, it is the use of patches that makes NLM more robust in comparison to pixel-based neighborhood filters [12], [11], [10].

Recently, it was demonstrated in [13] that the denoising performance of NLM can be improved (often substantially for images with sharp edges) by replacing the ℓ^2 regression in NLM with the more robust ℓ^1 regression. More precisely, given weights w_{ij}, note that (2) is equivalent to performing the following regression (on the patch space):

$$\hat{\mathbf{P}}_i = \arg\min_{\mathbf{P}} \sum_{j \in S(i)} w_{ij}\|\mathbf{P} - \mathbf{P}_j\|^2, \tag{4}$$

and then setting $\hat{u}_i$ to be the center pixel in $\hat{\mathbf{P}}_i$. Indeed, this reduces to (2) once we write the regression in terms of the center pixel $\hat{u}_i$. The idea in [13] was to use ℓ^1 regression instead, namely, to compute

$$\hat{\mathbf{P}}_i = \arg\min_{\mathbf{P}} \sum_{j \in S(i)} w_{ij}\|\mathbf{P} - \mathbf{P}_j\|, \tag{5}$$

and then set $\hat{u}_i$ to be the center pixel in $\hat{\mathbf{P}}_i$. Note that (5) is a convex optimization, and the minimizer (the Euclidean median) is unique when $k > 1$ [14]. The resulting estimator was called the Non-Local Euclidean Medians (NLEM). A numerical scheme was proposed in [13] for computing the Euclidean median using a sequence of weighted least-squares. It was demonstrated that NLEM performed consistently better than NLM on a large class of synthetic and natural images, as soon as the noise was above a certain threshold. More specifically, it was shown that the bulk of the improvement in NLEM came from pixels situated close to edges. An inlier-outlier model of the patch space around an edge was proposed,

**Program in Applied and Computational Mathematics (PACM), Princeton University, Princeton, NJ 08544, USA (kchaudhu@math.princeton.edu). K. Chaudhury was 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.

and the improvement was attributed to the robustness of (5) in the presence of outliers [17].

In this paper, we show how a simple extension of the above idea can dramatically improve the denoising performance of NLM, and even that of NLEM. This is the content of Section II. In particular, a general optimization and algorithmic framework is provided that includes NLM and NLEM as special cases. Some numerical results on synthetic and natural images are provided in Section III to justify our claims. Possible extensions of the present work are discussed in Section IV.

II. Non-Local Patch Regression

A. Robust patch regression

It is well-known that ℓ^1 minimization is more robust to outliers than ℓ^2 minimization. A simple argument is that the *unsquared* residuals $\|\mathbf{P}-\mathbf{P}_j\|$ in (5) are better guarded against the aberrant data points compared to the squared residuals $\|\mathbf{P}-\mathbf{P}_j\|^2$. The former tends to better suppress the large residuals that may result from outliers. This basic principle of robust statistics can be traced back to the works of von Neumann, Tukey [16], and Huber [17], and lies at the heart of several recent work on the design of robust estimators; e.g., see [15], and the references therein.

A natural question is what happens if we replace the ℓ^1 regression in (5) by $\ell_{(p<1)}$ regression? In general, one could consider the following class of problems:

$$\hat{\mathbf{P}}_i = \arg\min_{\mathbf{P}} \sum_{j\in S(i)} w_{ij}\|\mathbf{P}-\mathbf{P}_j\|^p. \tag{6}$$

The intuitive idea here is that, by taking smaller values of p, we can better suppress the residuals $\|\mathbf{P}-\mathbf{P}_j\|$ induced by the outliers. This should make the regression even more robust to outliers, compared to what we get with $p=1$. We note that a flip side of setting $p<1$ is that (6) will no longer be convex (this is essentially because $t\mapsto|t|^p$ is convex if and only if $p\geq 1$), and it is in general difficult to find the global minimizer of a non-convex functional. However, we do have a good chance of finding the global optimum if we can initialize the solver close to the global optimum. The purpose of this note is to numerically demonstrate that, for all sufficiently large σ, the $\hat{u}$ obtained by solving (6) (and letting $\hat{u}_i$ to be the center pixel in $\hat{\mathbf{P}}_i$) results in a more robust approximation of f as $p\to 0$, than what is obtained using NLM. Henceforth, we will refer to (6) as Non-Local Patch Regression (NLPR), where p is generally allowed to take values in the range $(0,2]$.

B. Iterative solver

The usefulness of the above idea actually stems from the fact that there exists a simple iterative solver for (6). In fact, the idea was influenced by the well-known connection between 'sparsity' and 'robustness', particularly the use of $l_{(p<1)}$ minimization for best-basis selection and exact sparse recovery [18], [19], [22]. We were particularly motivated by the iteratively reweighted least squares (IRLS) approach of Daubechies et al. [21], and a regularized version of IRLS developed by Chartrand for non-convex optimization [19], [20]. We will adapt the regularized IRLS algorithm in [19], [20] for solving (6). The exact working of this iterative solver is as follows. We use the NLM estimate to initialize the algorithm, that is, we set

$$\mathbf{P}^{(0)} = \frac{\sum_{j\in S(i)} w_{ij}\mathbf{P}_j}{\sum_{j\in S(i)} w_{ij}}. \tag{7}$$

Then, at every iteration $k\geq 1$, we write $\|\mathbf{P}-\mathbf{P}_j\|^p = \|\mathbf{P}-\mathbf{P}_j\|^2\cdot\|\mathbf{P}-\mathbf{P}_j\|^{p-2}$ in (6), and use the current estimate to approximate this by $\|\mathbf{P}-\mathbf{P}_j\|^2\cdot\|\mathbf{P}^{(t-1)}-\mathbf{P}_j\|^{p-2}$. This gives us the surrogate least-squares problem

$$\mathbf{P}^{(t)} = \arg\min_{\mathbf{P}} \sum_{j\in S(i)} w_{ij}\frac{\|\mathbf{P}-\mathbf{P}_j\|^2}{\left(\|\mathbf{P}^{(t-1)}-\mathbf{P}_j\|^2+\varepsilon^{(t)}\right)^{1-p/2}}. \tag{8}$$

Here $\varepsilon^{(t)}>0$ is used as a guard against division by zero, and is gradually shrunk to zero as the iteration progresses. We refer the reader to [19] for details. The solution of (8) is explicitly given by

$$\mathbf{P}^{(t)} = \frac{\sum_{j\in S(i)} w_{ij}\mu_j^{(t)}\mathbf{P}_j}{\sum_{j\in S(i)} w_{ij}\mu_j^{(t)}}, \tag{9}$$

where

$$\mu_j^{(t)} = (\|\mathbf{P}^{(t-1)}-\mathbf{P}_j\|^2+\varepsilon^{(t)})^{p/2-1}.$$

The minimizer of (6) is taken to be the limit of the iterates, assuming that it exists. We note that (9) can be expressed as a gradient descent step (with appropriate step size) of a smooth surrogate of (6). This interpretation leads to the well-known Weiszfeld algorithm (for the special case $p=1$), which is known to converge linearly [26], [27]. Following a slightly different interpretation, a preliminary analysis of the global and local convergence properties of (9) (particularly in the non-convex regime $0<p<1$) was done in [29].

The overall computational complexity of NLPR is $O(k^2S^2I)$ per pixel, where I is the average number of iterations. For NLM, the complexity is $O(k^2S^2)$ per pixel.

Algorithm 1 Non-Local Patch Regression (NLPR)

Input: Noisy image $u=(u_i)$, and parameters h,S,k,p.
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) Sort $w_{ij}, j\in S(i)$, in non-increasing order.
(c) Let $j_1,j_2,\ldots,j_{S^2}$ be the re-indexing of $j\in S(i)$ as per the above order.
(d) Find patch $\mathbf{P}$ that minimizes $\sum_{r=1}^{[S^2/2]} w_{ij_r}\|\mathbf{P}-\mathbf{P}_j\|^p$.
(e) Set $\hat{u}_i$ to be the center pixel in $\mathbf{P}$.

C. Robustness using k-nearest neighbors

We noticed in [13] that a simple heuristic often provides a remarkable improvement in the performance of NLM. In (2), one considers all patches $\mathbf{P}_j, j\in S(i)$, drawn from the geometric neighborhood of pixel i. However, notice that when a patch is close to an edge, then roughly half of its neighboring patches are on one side (the correct side) of the edge. Following this observation, we consider only the top 50% of the the

neighboring patches that have the largest weights. That is, the selected patches correspond to the $\lceil r/2 \rceil$-nearest neighbors of $\mathbf{P}_i$ in the patch space, where $r = |S(i)|$. While this tends to inhibit the diffusion at low noise levels (in smooth regions), it was demonstrated in [13] that it can significantly improve the robustness of NLM and NLEM at large σ. We will also use this heuristic in NLPR. The overall scheme is summarized in Algorithm 1. We use $S(i)$ to denote a window of size $S \times S$ centered at pixel i in the algorithm.

III. Numerical Experiments

To understand the denoising performance of NLPR, we provide some limited results on synthetic and natural images. The main theme of our investigation would be to understand how the performance of NLPR changes with the regression index p. For a quantitative comparison of the denoising results, we will use the standard peak-signal-to-noise ratio (PSNR). For an N-pixel image, with intensity scaled to $[0, 1]$, this is defined to be $-10 \log_{10}(\epsilon)$, where $\epsilon = (1/N) \sum_{i=1}^{N} (\hat{u}_i - f_i)^2$.

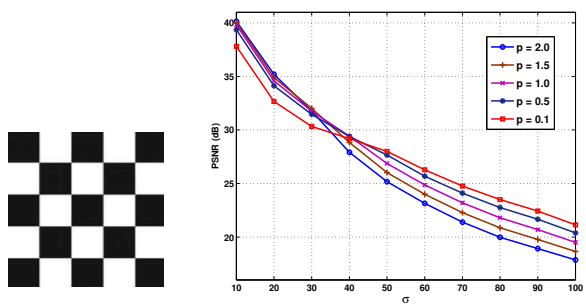
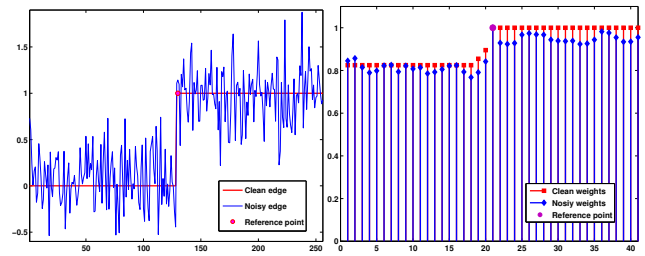


Fig. 1. Left: Test image *Checker*. Right: PSNRs obtained using NLPR for the test image *Checker* at different σ and p. To compare the different regressions, we skipped steps (b) and (c) in Algorithm 1, i.e., we consider all the neighboring patches, and not just the top 50%.

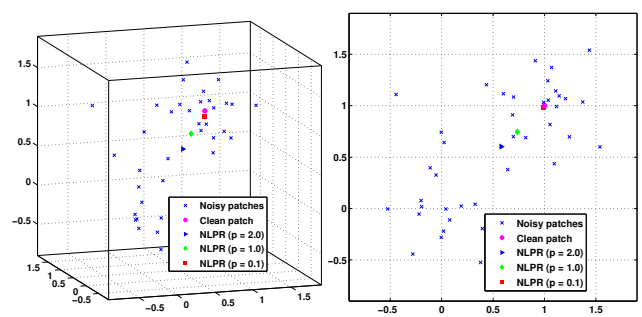
We first consider the test image of *Checker* used in [13]. This serves as a good model for simultaneously testing the denoising quality in smooth regions and in the vicinity of edges. We corrupt *Checker* as per the noise model in (1). We then compute the denoised image using Algorithm 1, with the exception that we skip steps (b) and (c), that is, we use the full neighborhood $S(i)$. We initialize the iterations of the IRLS solver using (7). For all the experiments in this paper, we fix the parameters to be $S = 21, k = 7$, and $h = 10\sigma$. These are the settings originally proposed in [1]. The results obtained using these settings are not necessarily optimal, and other settings could have been used as well. The point is to fix all the parameters in Algorithm 1, except p. This means that the same w_{ij} are used for different p. We now run the above denoising experiment for $\sigma = 10, 20, \ldots, 100$, and for $p = 0.1, 0.5, 1, 1.5, 2$.

The results are shown in Figure 1. We notice that, beyond a certain noise level, NLPR performs better when p is close to zero. In fact, the PSNR increases gradually from $p = 2$ to $p = 0.1$, for a fixed σ. At lower noise levels, the situation reverses completely, and NLPR tends to perform better around $p = 2$. A possible explanation is that the true neighbors in patch space are well identified at low noise levels, and since the noise is Gaussian, ℓ^2 regression gives statistically optimal results.



(a) Clean and noisy edge ($\sigma = 0.3$). (b) Weights.

Fig. 2. Ideal edge of length 256 used to evaluate the performance of NLPR. Each patch $\mathbf{P}_m$ is formed using 3 points around position m, i.e., the patch space is of dimension 3 (shown in Figure 3). The reference patch $\mathbf{P}_i$ corresponds to the position $i = 130$ (situated close to the edge). The weights in 2b are computed between the reference patch and the neighboring patches $\mathbf{P}_j, j \in [i - 20, i + 20]$.



(a) 3d patch space. (b) First 2 coordinates.

Fig. 3. Inlier-outlier model of the patch space for the reference point marked in Figure 2. Note that the estimate returned by NLPR gets better as p goes from 2 to 0.1. This is consistent with the results in Figure 1.

An analysis of the above results shows us that, as $p \to 0$, the bulk of the improvement comes from pixels situated in the vicinity of edges. A similar observation was also made in [13] for NLEM. To understand this better, we recall the ideal 0-1 edge model used in [13]. This is shown in Figure 2a. We add noise of strength $\sigma = 0.3$ to the edge, and denoise it using NLPR. We examine the regression at a reference point situated just right to the edge (cf. Figure 2b). The patch space at this point is specified using $k = 3$ and $S = 41$. The distribution of patches is shown in Figure 3. Note that the patches are clustered around the centers $A = (0, 0, 0)$ and $B = (1, 1, 1)$. For the reference point, the points around A are the outliers, while the ones around B are the inliers. We now perform ℓ^p regression on this distribution for $p = 0.1, 1$, and 2. The results obtained (Algorithm 1, steps (b) and (c) skipped) from a single noise realization are shown in Figure 3. The exact values of the estimate in this case are 0.61 ($p = 2$), 0.75 ($p = 1$), and 0.98 (0.1). The average estimate over 10 noise realizations are 0.58 ($p = 2$), 0.82 ($p = 1$), and 0.95 ($p = 0.1$).

We note that the working of the IRLS algorithm provides some insight into the robustness of ℓ^p regression. Note that

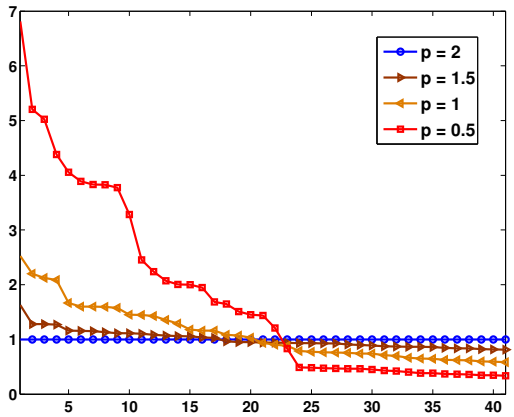


Fig. 4. The multipliers $\mu_j^{(\infty)}, j \in S(i)$, in (9) (sorted in non-increasing order) for the experiment with the ideal edge.

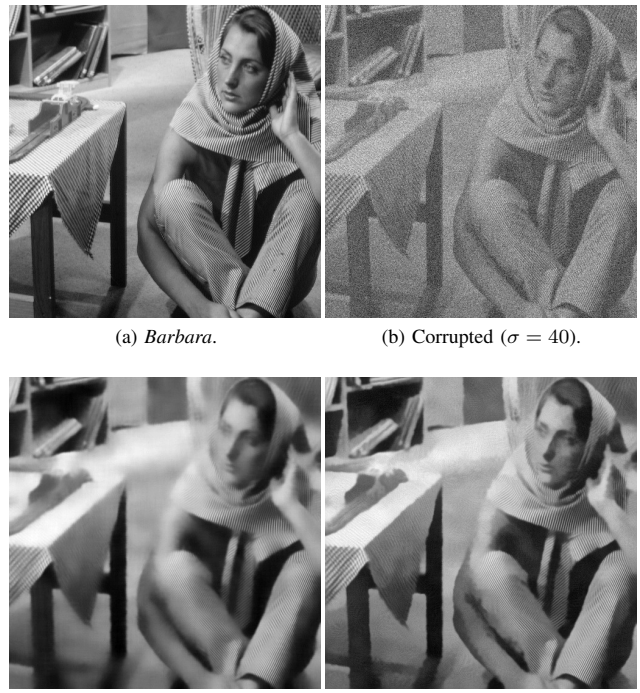
TABLE I
COMPARISON OF NLM AND NLPR ($p = 0.1$) AT NOISE LEVELS $\sigma = 10, 20, \ldots, 100$ (RESULTS AVERAGED OVER 10 NOISE REALIZATIONS)

Image	Method	PSNR (dB)									
House	NLM	**34.25**	29.76	26.88	25.21	24.08	23.34	22.81	22.42	22.05	21.80
	NLPR	33.23	**30.23**	**27.86**	**26.40**	**25.45**	**24.69**	**24.10**	**23.52**	**22.93**	**22.41**
Barbara	NLM	**32.38**	27.38	24.94	23.53	22.65	22.03	21.62	21.30	21.07	20.87
	NLPR	31.50	**28.42**	**26.51**	**25.39**	**24.57**	**23.84**	**23.21**	**22.60**	**22.06**	**21.56**
Boat	NLM	**30.78**	26.71	24.73	23.64	22.95	22.48	22.12	21.88	21.65	21.45
	NLPR	30.54	**27.23**	**25.50**	**24.50**	**23.87**	**23.40**	**22.95**	**22.54**	**22.11**	**21.68**
Cameraman	NLM	**31.39**	**27.90**	24.78	22.93	21.89	21.14	20.62	20.20	19.88	19.61
	NLPR	31.17	27.46	**25.15**	**25.15**	**22.68**	**22.12**	**21.67**	**21.36**	**20.97**	**20.63**
Peppers	NLM	**32.34**	27.66	24.95	23.13	21.89	21.01	20.43	19.98	19.63	19.40
	NLPR	31.20	**27.67**	**25.56**	**24.18**	**23.03**	**22.15**	**21.62**	**21.13**	**20.70**	**20.34**

when $p = 2$ (NLM), the reconstruction in (6) is linear; the contribution of each noisy patch $\mathbf{P}_j$ is controlled by the corresponding weight w_{ij}. On the other hand, the reconstruction is non-linear when $p < 2$. The contribution of each $\mathbf{P}_j$ is controlled not only by the respective weights, but also by the multipliers $\mu_j^{(t)}$. In particular, the limiting value of the multipliers dictate the contribution of each $\mathbf{P}_j$ in the final reconstruction. Figure (4) gives the distribution of the sorted multipliers (at convergence) for the experiment described above. In this case, the large multipliers correspond to the inliers, and the small multipliers correspond to the outliers. Notice that when $p = 0.5$, the tail part of the multipliers (outliers) has much smaller values (close to zero) compared to the leading part (inliers). In some sense, the iterative algorithm gradually 'learns' the outliers from the patch distribution as the iteration progresses, which are finally taken out of estimation.

IV. DISCUSSION

We compare the PSNRs obtained using NLPR ($p = 0.1$) with that of NLM for some standard natural images in Table I. We notice that, for each of the images, NLPR consistently outperforms NLM at large noise levels. The gain in PSNR is often as large as 2 dB. The results obtained for *Barbara* using NLM and NLPR are compared in Figure 5. Note that, as expected, robust regression provides a much better restoration of the sharp edges in the image than NLM. What is probably surprising is that the restoration is superior even in the textured regions. Note, however, that NLM tends to perform better in the smooth regions. For example, we observe some more noise grains in the smooth regions in Figure 5d compared that



(a) *Barbara*. (b) Corrupted ($\sigma = 40$).

(c) NLM output. (d) NLPR output ($p = 0.1$).

Fig. 5. Denoising results on the 256×256 *Barbara* image obtained using NLM and NLPR. The parameters used are: $S = 21, k = 7$, and $h = 10\sigma$. The PSNRs are respectively: (b) 16.11 dB, (c) 23.53 dB, and (d) **25.39** dB. Notice that the edges and the texture patterns (on the scarf, pants, and table cloth) are much better restored in NLPR.

in Figure 5c. This suggests that an 'adaptive' optimization framework, which combines ℓ^2 regression (in smooth regions) and $\ell_{(p\leq 1)}$ regression (in the vicinity of edges), might possibly perform better than a fixed ℓ^p regression. Some other possible extensions of the present work are as follows:

- Possibility of using more efficient numerical algorithms for solving (6).
- Finding better ways of estimating the denoised pixel $\hat{u}_i$ from the estimated patch $\hat{\mathbf{P}}_i$ (the projection method used here is probably the simplest).
- Note that we get an estimate a given pixel from all the neighboring (reconstructed) patches. Can the redundancy be exploited somehow to improve the estimation?
- Use of 'better' weights than the ones used in standard NLM [24], [25].
- Formulation of a 'joint' optimization framework for (6), where the optimization is performed with respect to w_{ij} and $\mathbf{P}$ [6].

To conclude, we note that it was recently observed in [28] that the weights in NLM and the associated estimate (2) can be derived from a two-step optimization of a non-convex cost. Based on this formulation, the authors in [28] argue that the w_{ij} in NLPR should be set using the p-th power of the residual, namely as $\|\mathbf{P}_i - \mathbf{P}_j\|^p$, as opposed to $\|\mathbf{P}_i - \mathbf{P}_j\|^2$ used in this paper (of course, this also requires us to use a different rule for setting h). This is an interesting observation that will be investigated in future work.

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] C. Kervrann, J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," IEEE Transactions on Image Processing, vol. 15(10), 2866-2878, 2006.

[4] G. Gilboa, S. Osher, "Nonlocal operators with applications to image processing," Multiscale Modeling and Simulation, vol. 7, no. 3, 1005-1028, 2008.

[5] M. Aharon, M. Elad, A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311-4322, 2006.

[6] G. Peyré, S. Bougleux, L. D. Cohen, "Non-local Regularization of Inverse Problems," Inverse Problems and Imaging, vol. 5(2), pp. 511-530. 2011.

[7] P. Chatterjee, P. Milanfar, "Patch-based near-optimal image denoising," IEEE Transactions on Image Processing, vol. 21, no. 4, pp. 1635-1649, 2012.

[8] C. Deledalle, V. Duval, J. Salmon, "Non-local methods with shape-adaptive patches (NLM-SAP)," Journal of Mathematical Imaging and Vision, vol. 43, no. 2, pp. 103-120, 2012.

[9] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," IEEE Transactions on Image Processing, vol. 16, pp. 2080-2095, 2007.

[10] C. Tomasi, R. Manduchi, "Bilateral filtering for gray and color images," IEEE International Conference on Computer Vision, pp. 839-846, 1998.

[11] S. M. Smith, J. M. Brady, "SUSAN - A new approach to low level image processing," International Journal of Computer Vision, vol. 23, no. 1, pp. 45-78, 1997.

[12] L. P. Yaroslavsky, Digital Picture Processing. Secaucus, NJ: Springer-Verlag, 1985.

[13] K. N. Chaudhury, A. Singer, "Non-local Euclidean medians," IEEE Signal Processing Letters, vol. 19, no. 11, pp. 745 - 748, 2012.

[14] P. Milasevic, G. R. Ducharme, "Uniqueness of the Spatial Median", Annals of Statistics, vol. 15, no. 3, 1332-1333, 1987.

[15] G. Lerman, M. McCoy, J. A. Tropp, T. Zhang, "Robust computation of linear models, or how to find a needle in a haystack," arXiv:1202.4044 [cs.IT].

[16] J. W. Tukey, Exploratory Data Analysis. Addison-Wesley, Reading, Mass., 1977.

[17] P. J. Huber, E. M. Ronchetti, Robust Statistics. Wiley Series in Probability and Statistics, Wiley, 2009.

[18] B. D. Rao, K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," IEEE Transactions on Signal Process., vol. 47, pp. 187-200, 1999.

[19] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," IEEE Signal Processing Letters, vol. 14, pp. 707-710, 2007.

[20] R. Chartrand, "Iteratively reweighted algorithms for compressive sensing," IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3869-3872, 2008.

[21] 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.

[22] R. Saaba, O. Yilmazb, "Sparse recovery by non-convex optimization – instance optimality," Applied and Computational Harmonic Analysis, vol. 29(1), pp. 30-48, 2010.

[23] J. Ji, "Robust inversion using biweight norm and its application to seismic inversion," Exploration Geophysics, vol. 43(2), 70-76, 2012.

[24] T. Tasdizen, "Principal neighborhood dictionaries for non-local means image denoising," IEEE Transaction on Image Processing, vol. 18, pp. 2649-2660, 2009.

[25] D. Van De Ville, M. Kocher, "Nonlocal means with dimensionality reduction and SURE-based parameter selection," IEEE Transactions on Image Processing, vol. 20, pp. 2683-2690, 2011.

[26] E. Weiszfeld, "Sur le point par lequel le somme des distances de n points donnes est minimum," Tohoku Mathematical Journal, vol. 43, pp. 355-386, 1937.

[27] R. Hartley, K. Aftab, J. Trumpf, "L1 rotation averaging using the Weiszfeld algorithm," IEEE Conference on Computer Vision and Pattern Recognition, pp. 3041-3048, 2011.

[28] Z. Sun, S. Chen, "Analysis of non-local Euclidean medians and its improvement", IEEE Signal Processing Letters, vol. 20, no. 4, pp. 303-306, 2013.

[29] K. N. Chaudhury, On the convergence of the IRLS algorithm in Non-Local Patch Regression, submitted.