

## A FIXED-POINT IMAGE DENOISING ALGORITHM WITH AUTOMATIC WINDOW SELECTION

*Jussi Määttä<sup>1\*</sup>, Samuli Siltanen<sup>2</sup> and Teemu Roos<sup>1</sup>*

<sup>1</sup>Helsinki Institute for Information Technology HIIT,  
Department of Computer Science,  
University of Helsinki, Finland

<sup>2</sup>Department of Mathematics and Statistics,  
University of Helsinki, Finland

### ABSTRACT

A novel image denoising approach based on iterated median filtering is proposed. It is well suited for removing white noise and produces visually pleasing smooth surfaces while preserving edges and without producing artifacts. The denoised image is the fixed point of a nonlinear operator and can be obtained as the limit of a convergent sequence. We show that the sequence converges at an exponential rate. An algorithm implementing the proposed method is described; in addition to the fixed-point iteration, it automatically selects a suitable window and optimizes a scalar parameter based on an assumed noise level. The results of preliminary simulation experiments on well-known test images are presented. While the proposed method does not outperform earlier methods, it offers a theoretically well-understood foundation for future development; promising further research directions are discussed.

*Index Terms*— Image denoising, nonlinear filters, iterative algorithms, model selection, statistical analysis.

### 1. INTRODUCTION

In this paper, we define the task of image denoising as follows. We are given a discrete grayscale image that has been contaminated with additive i.i.d. Gaussian noise. The variance of the noise is assumed to be known. The goal is to recover the original image.

To solve this problem, one must define a model for images. While there have been attempts to explicitly define generic image models (e.g. [1]), most image denoising methods seem to implicitly incorporate a model within the flow of a complicated algorithm. As an example of the latter, we mention the highly successful BM3D algorithm [2] for which it seems difficult to describe the class of images that are modeled.

Many well-known denoising methods start from relatively simple premises and build complexity upon them to improve the results. For instance, wavelet thresholding is a lucid and

powerful method which can be improved by, for instance, adding adaptivity [3] or employing the minimum description length principle [4]. One method that stands out as being straightforward to describe is the total variation approach, introduced by Rudin, Osher and Fatemi [5].

In this paper, we propose a new denoising algorithm that is easy to understand and implement and performs relatively well, although not as well as its more complicated competitors. Our method is based on iterated median filtering and its output qualitatively resembles that of total variation denoising: there are no artifacts typical to wavelet-based methods and smooth surfaces are favored over textures.

### 2. METHODS

#### 2.1. Model

Our setting is the Banach space  $\mathcal{X} = \mathbb{R}^{N \times M}$  with the norm  $\|x\| = \max_{i,j} |x(i,j)|$ . In all our examples, we have  $N = M = 512$  and the pixel values of noise-free images lie within the range  $[0, 255]$ .

We denote the median of a multiset of reals  $A = \{a_1, \dots, a_k\}$  by  $\text{med } A$  (if  $k$  is even, the median is the average of the two middle numbers). Given a finite window  $W \subset \mathbb{Z}^2$ , the median filter is defined as  $M_W: \mathcal{X} \rightarrow \mathcal{X}$ ,  $M_W(x)(i,j) = \text{med}\{x(i+s, j+t) : (s,t) \in W\}$ . When out-of-bound indices appear, we use the symmetric extension (that is, the image is padded with mirror reflections).

Let  $W$  be a window, assume  $(0,0) \in W$  and let  $\lambda > 0$ . Given a noisy image  $v \in \mathcal{X}$ , we seek a solution to the equation

$$u = \left(\frac{1}{\lambda+1}\right)v + \left(\frac{\lambda}{\lambda+1}\right)M_W(u) \quad (1)$$

as a denoised version of  $v$ . If we keep  $W$  fixed and denote  $\alpha = \lambda/(\lambda+1) \in (0,1)$ , equation (1) can be written simply as  $u = (1-\alpha)v + \alpha M(u)$ .

We offer the following interpretation for (1). Note first that the noisy image  $v$  is proportional to  $u - \alpha M(u)$ . If  $\alpha$  is close to zero, we have  $v \propto (u - \alpha M(u)) \approx (u - \alpha u) \propto u$  since the difference is dominated by the effect of  $\alpha$ ; here almost no denoising takes place. On the other hand, if  $\alpha$  is close to

Corresponding author: Jussi Määttä (jussi.maatta@helsinki.fi).

J.M. and T.R. are supported by the Academy of Finland under the Finnish Centre of Excellence in Computational Inference Research (COIN). S.S. is supported by the Academy of Finland under the Finnish Centre of Excellence in Inverse Problems Research (decision number 250215).

one, we have  $v \propto (u - \alpha M(u)) \approx (u - M(u))$  which has the interpretation that  $v$  is the result of amplifying the non-smooth component of  $u$ . The task of finding a good value for  $\alpha$  (equivalently, for  $\lambda$ ) is then a question of balancing between dampening noise and deviating from the input image.

The value of  $\lambda$  will be adaptively chosen so that the Frobenius norm  $\|u - v\|_F$  is approximately  $\sqrt{NM}\sigma^2$ , the expected energy of the noise. Intuitively, the greater the value of  $\lambda$ , the more noise equation (1) will be able to suppress. In fact, preliminary numerical results suggest that the optimal value of  $\lambda$  grows roughly linearly as a function of  $\sigma^2$ .

The model defined by (1) has a simple characterization. An image is unaltered,  $u = v$ , if and only if  $v = M_W(v)$ . That is, an image is considered noise-free precisely when it is a root image of the median filter  $M_W$ .

Notice also that given  $u$ , one can recover the noisy image  $v$ .

As shown by the following theorems, the equation (1) in fact has a unique solution.

**Theorem 1.** *Let  $W \subset \mathbb{Z}^2$  be finite. Then  $\|M_W(x) - M_W(y)\| \leq \|x - y\|$  for all  $x, y \in \mathcal{X}$*

For a proof of the above theorem, see the appendix.

**Theorem 2.** *The equation (1) has a unique solution  $u \in \mathcal{X}$  that is the limit of the sequence  $u_0 \in \mathcal{X}$ ,  $u_n = (1 - \alpha)v + \alpha M(u_{n-1})$ . Moreover, this sequence satisfies the inequalities*

$$\begin{aligned} \|u - u_n\| &\leq \frac{\alpha^n}{1 - \alpha} \|u_1 - u_0\|, \\ \|u - u_n\| &\leq \frac{\alpha}{1 - \alpha} \|u_n - u_{n-1}\|. \end{aligned}$$

*Proof.* Consider the operator  $T(x) = (1 - \alpha)v + \alpha M(x)$ . By Theorem 1, it is a contraction:  $\|T(x) - T(y)\| = \alpha \|M(x) - M(y)\| \leq \alpha \|x - y\|$ . The claim now directly follows from the Banach fixed-point theorem (see e.g. [6]).  $\square$

## 2.2. Window selection

The success of (1) must depend on the choice of the window  $W$ . We propose to use the family  $W_r = \{(s, t) \in \mathbb{Z}^2 : s^2 + t^2 \leq r^2\}$ . In practice, we only consider the values  $1 \leq r \leq R$  with  $R = 5$ ; larger windows are unlikely to provide significant improvements.

To determine a suitable radius for a given input image, we have devised four heuristics based on cross-validation and two model selection criteria: the Bayesian information criterion (BIC) [7] and the sequentially normalized least squares (SNLS) criterion [8].

The basic approach is always the same: For each radius  $r$ , define a model  $\mathcal{M}_r$  that attempts to predict the value of each pixel of the noisy image given the pixels in its neighborhood  $W'_r = W_r \setminus \{(0, 0)\}$ . Then use a suitable model selection criterion to pick the best model, say  $\mathcal{M}_{r^*}$ , and use the window  $W_{r^*}$  for the denoising procedure.

We mention that the problem of window size selection has been extensively studied for e.g. the local polynomial approximation framework (see [9] and the references therein). However, it is not clear whether the use of these intricate approaches can be justified for our setting where the final result emerges from the fixed point of a nonlinear operator.

Our heuristics are presented below. Their evaluation is postponed to Section 3.

**LinMod-BIC.** Let  $1 \leq r \leq R$  and enumerate the elements of  $W'_r$  as  $(s_1, t_1), (s_2, t_2), \dots, (s_{k_r}, t_{k_r})$ . The model  $\mathcal{M}_r$  predicts

$$\hat{v}_r(i, j) = \sum_{\ell=1}^{k_r} \beta_\ell v(i + s_\ell, j + t_\ell)$$

where the parameters  $\beta_\ell$  (independent of  $(i, j)$ ) minimize the mean squared error  $\hat{\sigma}_r^2 = (NM)^{-1} \sum_{i,j} (v(i, j) - \hat{v}_r(i, j))^2$ . The score is

$$\text{BIC}(\mathcal{M}_r) = \log(\hat{\sigma}_r^2) + \frac{k_r \log(NM)}{NM}.$$

The smaller the score, the better.

**LinMod-SNLS.** The model  $\mathcal{M}_r$  is the same as for LinMod-BIC. The formula for SNLS( $\mathcal{M}_r$ ) is too complex to describe here; see [8] for details. We note that the SNLS score requires an ordering for the pixels (although its effect vanishes asymptotically), so we use a random permutation.

**Mean-LOO.** For each  $1 \leq r \leq R$ , the model  $\mathcal{M}_r$  predicts

$$\hat{v}_r(i, j) = \frac{1}{|W'_r|} \sum_{(s,t) \in W'_r} v(i + s, j + t).$$

We choose the model that minimizes  $\hat{\sigma}_r^2$ . This corresponds to leave-one-out (LOO) cross-validation.

**Median-LOO.** Similar to Mean-LOO, except that the model  $\mathcal{M}_r$  predicts

$$\hat{v}_r(i, j) = \text{med}\{v(i + s, j + t) : (s, t) \in W'_r\}.$$

## 2.3. The full algorithm

We now assume that one of the four heuristics described above has been chosen.

The procedure *MedIter* performs denoising according to equation (1) and using the iterations implied by Theorem 2. The input variable  $u_{init}$  is an arbitrary initial guess for the output  $u$ .

*MedIter* is used as a subroutine of *AdaptMedIter*, which adapts the value of  $\lambda$  to match the variance of the noise. It finds an upper bound for  $\lambda$  with the hard limit  $\lambda \leq 128$ ; since smaller values of  $\lambda$  imply faster convergence in *MedIter*, we start from a small initial guess. After an upper bound has been found, *AdaptMedIter* performs binary search to find a good value for  $\lambda$ .

The full algorithm is the following. Given a noisy image  $v$  and the noise variance  $\sigma^2$ , select the best window  $W$  by computing the score for each radius. Then call *AdaptMedIter*( $v, \sigma^2, W$ ) to obtain the denoised image.

---

**Procedure MedIter**( $v, u_{init}, \lambda, W$ )

---

```
error  $\leftarrow +\infty$ 
 $\alpha \leftarrow \lambda / (\lambda + 1)$ 
 $u \leftarrow u_{init}$ 
while error > 0.1 do
  |  $u_{prev} \leftarrow u$ 
  |  $u \leftarrow (1 - \alpha)v + \alpha M_W(u)$ 
  | error  $\leftarrow \alpha(1 - \alpha)^{-1} \|u - u_{prev}\|$ 
end
return  $u$ 
```

---

### 3. RESULTS

#### 3.1. Large-scale experiment

We tested the performance of our algorithm on the van Hateren dataset [10] of natural images. We took a random subset of 400 images, cropped them to the middle  $512 \times 512$  pixels, and rescaled the pixel values so that the minimum and maximum values in each image are 0 and 255, respectively. To each image, we added i.i.d. Gaussian noise with standard deviations  $\sigma \in \{10, 30, 50\}$ . We then computed the four model selection scores for the noisy image and performed the denoising using each window. Each experiment was repeated ten times with different random noise realizations. All results stated in this subsection are from this setup.

Denoising performance was measured using both the peak signal-to-noise ratio,  $\text{PSNR} = 10 \log_{10}(255^2 / \text{MSE})$ , and the SSIM index [11].

As a baseline to which we compare our method, we chose to use total variation (TV) denoising [5], more specifically, the implementation by Peyre [12] that uses Chambolle’s algorithm [13]. This comparison is motivated by the observations that both methods (i) have relatively simple foundations, and (ii) tend to produce smooth surfaces separated by sharp edges.

Figure 1 shows the relationship between the radius of the window and the PSNR. For  $\sigma = 10$ , using the smallest window ( $r = 1$ ) gives the best results. In all other cases,  $r = 2$  tends to perform best. The corresponding results for SSIM are qualitatively similar. The figure suggests that considering larger radii would not bring significant benefits.

The effects of using the various window selection heuristics described in the previous section are shown in Figure 2. For comparison, we have included both the baseline method (total variation denoising) and the “oracle” heuristic that always chooses the window size that performs the best under the image quality measure being considered. In all cases, Mean-LOO is the best of the four heuristics tested. Still, the baseline consistently performs better than any of the heuristics. Notably, though, using the oracle for window selection results in performance comparable to the baseline especially with the SSIM measure. This suggests that coming up with better heuristics should be a major objective for future research.

---

**Procedure AdaptMedIter**( $v, \sigma^2, W$ )

---

```
// Initial upper bound for  $\lambda$ 
 $u \leftarrow v$ ;  $\lambda_{LB} \leftarrow 0$ ;  $\lambda_{UB} \leftarrow 1$ 
while true do
  |  $u \leftarrow \text{MedIter}(v, u, \lambda_{UB}, W)$ 
  | if  $\|u - v\|_F \geq \sqrt{NM\sigma^2}$  then
  | | break
  | else
  | |  $\lambda_{LB} \leftarrow \lambda_{UB}$ 
  | |  $\lambda_{UB} \leftarrow 2\lambda_{UB}$ 
  | | if  $\lambda_{UB} \geq 128$  then
  | | | break
  | | end
  | end
end

// Binary search for the best  $\lambda$ 
 $\lambda \leftarrow +\infty$ 
while true do
  |  $\lambda_{prev} \leftarrow \lambda$ 
  |  $\lambda \leftarrow (\lambda_{LB} + \lambda_{UB})/2$ 
  | if  $|\lambda - \lambda_{prev}| \leq 0.1$  then
  | | break
  | end
  |  $u \leftarrow \text{MedIter}(v, u, \lambda, W)$ 
  | if  $\|u - v\|_F \leq \sqrt{NM\sigma^2}$  then
  | |  $\lambda_{LB} \leftarrow \lambda$ 
  | | else
  | | |  $\lambda_{UB} \leftarrow \lambda$ 
  | | end
  | end
end
return  $u$ 
```

---

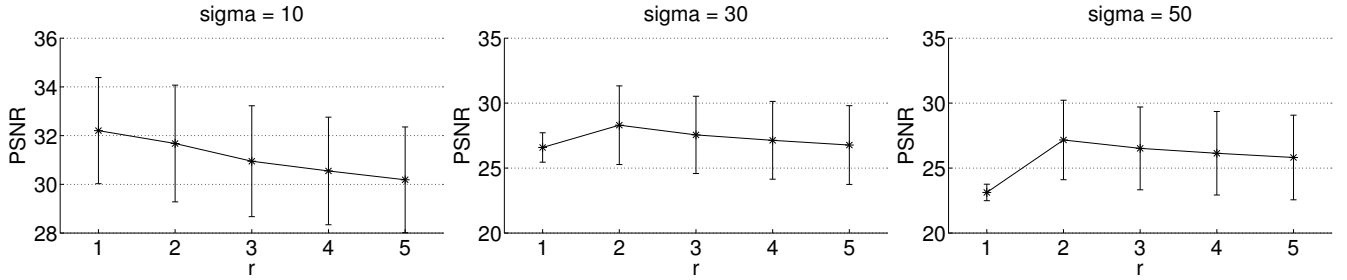
The second best heuristic is Median-LOO. Finally, the two heuristics based on a linear model appear inferior.

#### 3.2. Performance on common test images

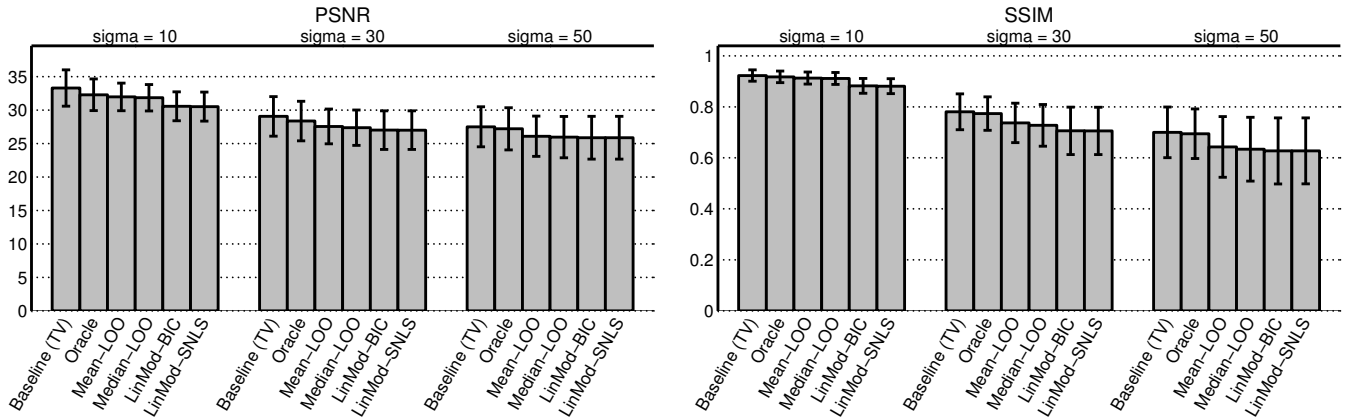
In this experiment, we measure denoising performance on the well-known test images *lena*, *barbara* and *goldhill*. We only use the Mean-LOO heuristic as it achieved the best performance in the previous experiment. We compare the PSNR values of our method, TV denoising, BayesShrink [3] and BM3D [2]. The results are presented in Table 1.

To illustrate the effect of window choice, we look more closely on the performance of our algorithm on *barbara* with  $\sigma = 30$ . The Mean-LOO heuristic chooses  $r = 4$ , resulting in  $\text{PSNR} = 23.6$ , but this choice is hardly optimal:  $r = 2$  would give  $\text{PSNR} = 24.0$ . Figure 3 shows the algorithm’s output for each  $1 \leq r \leq 5$  alongside with the respective PSNR’s and the baseline result.

Running the full algorithm on *barbara* with  $\sigma = 30$



**Fig. 1.** The effect of window radius on denoising performance on the van Hateren dataset, as measured by the peak signal-to-noise ratio (PSNR). The error bars show one standard deviation. Using the SSIM measure gives similar results (not displayed here due to limited space).



**Fig. 2.** Average denoising performance using the various window selection heuristics (left: PSNR, right: SSIM) with the van Hateren dataset, compared to the baseline (total variation denoising). The “oracle” heuristic always chooses the best window. The error bars show one standard deviation.

took approximately 52 seconds (averaged over 30 trials) on a MacBook Pro (2.4 GHz Intel Core i5) with MATLAB R2013a. The corresponding runtime for the TV algorithm was 16 seconds.

#### 4. DISCUSSION

Our proposed denoised method performs reasonably well when using the Mean-LOO heuristic for window selection. For a low noise level ( $\sigma = 10$ ), the algorithm performed almost as well as the baseline under the SSIM measure. The algorithm is extremely simple to implement<sup>1</sup>, has a clear interpretation and converges at a known rate. We anticipate that these properties may be useful in fields such as medical imaging, where every extraneous source of complexity must be carefully justified.

There is still room for improvement in the window selection mechanism, as discussed in Section 3.2. As can be seen from Figure 2, developing a heuristic that could match the oracle would result in an algorithm with performance comparable to total variation denoising in a variety of settings.

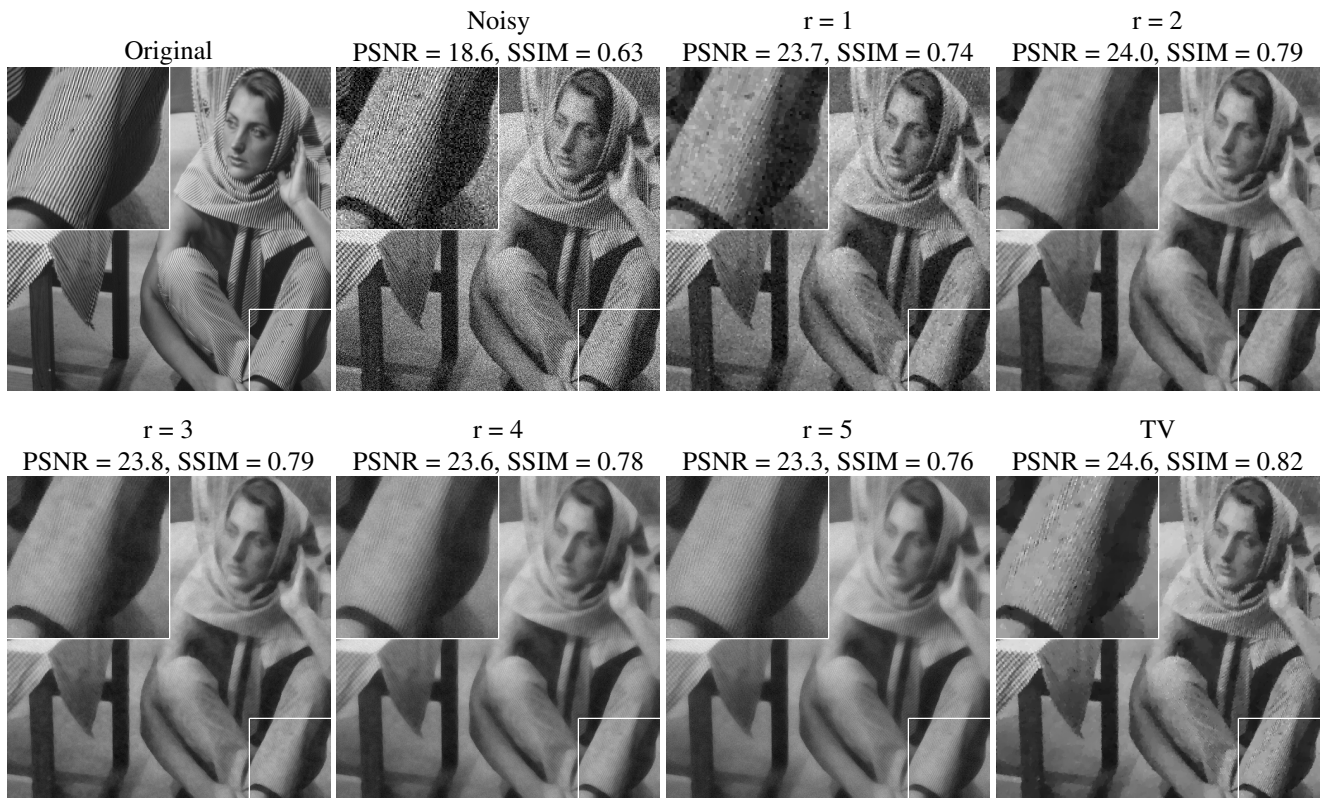
<sup>1</sup>Esp. in Matlab, where the built-in function `ordfilt2` comes in useful.

We have made some experiments with the weighted median filter [14], and it may potentially improve denoising performance in some cases. It is not yet clear to us how the window selection procedure should be adapted to this setting.

Another possible extension would be to use a different window for each pixel. It is easy to show that this would not affect convergence. Adapting the algorithm to local properties of the image could potentially offer drastic improvements.

#### 5. REFERENCES

- [1] D. Mumford and B. Gidas, “Stochastic models for generic images,” *Q. Appl. Math.*, vol. 59, no. 1, pp. 85–111, 2001.
- [2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE T. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [3] S. G. Chang, B. Yu, and M. Vetterli, “Adaptive wavelet thresholding for image denoising and compression,”



**Fig. 3.** Denoising results for the test image *lena* with  $\sigma = 30$ , using the radii  $1 \leq r \leq 5$ . The  $128 \times 128$  region at the bottom right is shown magnified in the upper left corner. The Mean-LOO heuristic chooses  $r = 4$ . The output of the baseline method (total variation denoising) is also shown.

- IEEE T. Image Process.*, vol. 9, no. 9, pp. 1532–1546, 2000.
- [4] T. Roos, P. Myllymäki, and J. Rissanen, “MDL denoising revisited,” *IEEE T. Signal Proces.*, vol. 57, no. 9, pp. 3347–3360, 2009.
- [5] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D*, vol. 60, no. 1–4, pp. 259–268, 1992.
- [6] E. Zeidler, *Nonlinear functional analysis and its applications. I*, Springer-Verlag, New York, 1986.
- [7] G. Schwarz, “Estimating the dimension of a model,” *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 3 1978.
- [8] J. Rissanen, T. Roos, and P. Myllymäki, “Model selection by sequentially normalized least squares,” *J. Multivariate Anal.*, vol. 101, no. 4, pp. 839–849, 2010.
- [9] V. Katkovnik, K. Egiazarian, and J. Astola, “Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule,” *J. Math. Imaging Vis.*, vol. 16, no. 3, pp. 223–235, 2002.
- [10] J. H. van Hateren and A. van der Schaaf, “Independent component filters of natural images compared with simple cells in primary visual cortex,” *P. Roy. Soc. B-Biol. Sci.*, vol. 265, no. 1394, pp. 359–366, 1998.
- [11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE T. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.
- [12] G. Peyre, “Toolbox image - a toolbox for general purpose image processing,” 2008, URL: <https://www.ceremade.dauphine.fr/~peyre/matlab/image/content.html>.
- [13] Antonin Chambolle, “An algorithm for total variation minimization and applications,” *J. Math. Imaging Vis.*, vol. 20, no. 1–2, pp. 89–97, 2004.
- [14] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, “Weighted median filters: a tutorial,” *IEEE T. Circuits-II*, vol. 43, no. 3, pp. 157–192, 1996.
- [15] E. Arias-Castro and D. L. Donoho, “Does median filtering truly preserve edges better than linear filtering?,” *Ann. Stat.*, vol. 37, no. 3, pp. 1172–1206, 2009.

$\sigma$	Proposed	TV	BayesShrink	BM3D
10	32.9	33.3	33.4	35.9
20	29.2	30.1	30.2	33.1
30	27.0	28.4	28.5	31.3

$\sigma$	Proposed	TV	BayesShrink	BM3D
10	28.6	30.0	31.0	35.0
20	25.1	26.2	27.3	31.8
30	23.6	24.6	25.3	29.8

$\sigma$	Proposed	TV	BayesShrink	BM3D
10	31.3	31.8	31.9	33.6
20	28.2	28.9	28.7	30.7
30	26.1	27.4	27.1	29.2

**Table 1.** Denoising performance for three test images and three noise levels, as measured by the PSNR. The results for the proposed method and TV denoising are averages over 30 trials. The values for BayesShrink and BM3D are from the corresponding papers [3, 2].

## A. PROOF OF THEOREM 1

**Lemma 1.** Let  $a, b, c, d \in \mathbb{R}$  with  $a \leq b$  and  $c \leq d$ . Then  $\max\{|a - c|, |b - d|\} \leq \max\{|a - d|, |b - c|\}$ .

*Proof.* Easy to see e.g. by separately considering the six possible orderings of the four numbers.  $\square$

**Lemma 2.** Let  $J$  be a finite index set of  $N$  elements. Consider the numbers  $x(j), y(j) \in \mathbb{R}$  indexed by  $j \in J$ . Let  $n, m: \{1, 2, \dots, N\} \rightarrow J$  be bijections that satisfy

$$x(n(1)) \leq x(n(2)) \leq \dots \leq x(n(N)) \quad \text{and} \\ y(m(1)) \leq y(m(2)) \leq \dots \leq y(m(N)).$$

(In other words, the mappings  $n$  and  $m$  sort the numbers  $x(\cdot)$  and  $y(\cdot)$ , respectively.) Then we have

$$\max_{1 \leq i \leq N} |x(n(i)) - y(m(i))| \leq \max_{j \in J} |x(j) - y(j)|,$$

that is, by sorting  $x(\cdot)$  and  $y(\cdot)$  we cannot increase the maximum pointwise distance.

*Proof.* We denote  $\tilde{x}(i) = x(n(i))$  and  $y_0(i) = y(m(i))$ ; this means that we sort the numbers  $x(\cdot)$  and arrange the numbers  $y(\cdot)$  so that the pairwise differences  $|x(\cdot) - y(\cdot)|$  coincide with the differences  $|\tilde{x}(\cdot) - y_0(\cdot)|$ . Then, for each  $k = 1, 2, \dots, N$ , we define recursively

$$p_k = \arg \min_{k \leq i \leq N} y_{k-1}(i), \quad y_k(i) = \begin{cases} y_{k-1}(p_k), & \text{if } i = k, \\ y_{k-1}(k), & \text{if } i = p_k, \\ y_{k-1}(i), & \text{otherwise.} \end{cases}$$

These definitions mean that we sort the values of  $y_0$  using a selection sort-type algorithm, denoting by  $y_k$  a permutation of  $y$  with the first  $k$  values sorted. Hence,  $y_N$  is fully sorted. To be precise, we define the arg min in  $p_k$  to pick the  $i$  corresponding to the mapping  $m$  in case of a tie.

We want to show that

$$\max_{1 \leq i \leq N} |\tilde{x}(i) - y_k(i)| \leq \max_{j \in J} |x(j) - y(j)| \quad (2)$$

for every  $k = 0, 1, 2, \dots, N$ . The case  $k = 0$  is trivial. Suppose that the inequality (2) holds for some  $k = 0, 1, 2, \dots, N - 1$ . By definition,  $k + 1 \leq p_{k+1}$ , so  $\tilde{x}(k + 1) \leq \tilde{x}(p_{k+1})$ . Moreover,

$$y_k(p_{k+1}) = y_k(\arg \min_{k+1 \leq i \leq N} y_k(i)) \leq y_k(k + 1).$$

Now Lemma 1 gives

$$\max\{|\tilde{x}(k + 1) - y_k(p_{k+1})|, |\tilde{x}(p_{k+1}) - y_k(k + 1)|\} \\ \leq \max\{|\tilde{x}(k + 1) - y_k(k + 1)|, |y_k(p_{k+1}) - \tilde{x}(p_{k+1})|\},$$

so we have

$$\max_{1 \leq i \leq N} |\tilde{x}(i) - y_{k+1}(i)| \\ = \max\left\{|\tilde{x}(k + 1) - y_k(p_{k+1})|, |\tilde{x}(p_{k+1}) - y_k(k + 1)|, \right. \\ \left. \max_{\substack{1 \leq i \leq N, \\ i \notin \{k+1, p_{k+1}\}}} |\tilde{x}(i) - y_k(i)|\right\} \\ \leq \max\left\{|\tilde{x}(k + 1) - y_k(k + 1)|, |y_k(p_{k+1}) - \tilde{x}(p_{k+1})|, \right. \\ \left. \max_{\substack{1 \leq i \leq N, \\ i \notin \{k+1, p_{k+1}\}}} |\tilde{x}(i) - y_k(i)|\right\} \\ = \max_{1 \leq i \leq N} |\tilde{x}(i) - y_k(i)| \leq \max_{j \in J} |x(j) - y(j)|.$$

Hence, (2) holds for all  $k = 0, 1, 2, \dots, N$ . The case  $k = N$  was our claim.  $\square$

The desired result is a straightforward corollary of the previous lemma.

*Proof of Theorem 1.* A special case of Lemma 2 is that

$$\left| \text{med}_{j \in J} x(j) - \text{med}_{j \in J} y(j) \right| \leq \max_{j \in J} |x(j) - y(j)|.$$

This result was presented without a proof by Arias-Castro and Donoho [15, eq. (6.2)]. Now, when  $x, y \in \mathcal{X}$ , we have

$$\|M_W(x) - M_W(y)\| \\ = \max_{i,j} |M_W(x)(i, j) - M_W(y)(i, j)| \\ \leq \max_{i,j} \max_{(s,t) \in W} |x(i + s, j + t) - y(i + s, j + t)| \\ = \|x - y\|. \quad \square$$