Some nonlocal filters formulation using functional rearrangements *

Gonzalo Galiano and Julián Velasco

Dpt. of Mathematics, Universidad de Oviedo, Spain
galiano@uniovi.es, juliano@uniovi.es

Abstract. We present an exact reformulation of a broad class of nonlocal filters, among which the bilateral filters, in terms of two functional rearrangements: the decreasing and the relative rearrangements.
Independently of the *image* spatial dimension, these filters are expressed as integral operators defined in a one-dimensional space, corresponding to the level sets measures.
We provide some insight into the properties of this new formulation and show some numerical demonstrations to illustrate them.

Keywords: Neighborhood filter, bilateral filter, decreasing rearrangement, relative rearrangement, denoising, segmentation.

1 Introduction

Let $\Omega \subset \mathbb{R}^d$ be an open and bounded set, $u \in L^\infty(\Omega)$ be an intensity image, and consider the family of filters, for h and ρ positive constants,

$$\mathrm{F}\, u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_\Omega \mathcal{K}_h(u(\mathbf{x}) - u(\mathbf{y})) w_\rho(|\mathbf{x} - \mathbf{y}|) u(\mathbf{y}) d\mathbf{y},$$

where $C(\mathbf{x}) = \int_\Omega \mathcal{K}_h(u(\mathbf{x}) - u(\mathbf{y})) w_\rho(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}$ is a normalization factor.

Functions $\mathcal{K}_h(\xi) = \mathcal{K}(\xi/h)$ and w_ρ are the *range* kernel and the *spatial* kernel of the filter, respectively, making reference to their type of interaction with the image domain. A usual choice for $\mathcal{K}$ is the Gaussian $\mathcal{K}(\xi) = \exp(-\xi^2)$, while different choices of w_ρ give rise to several well known nonlocal filters, e.g.,

- The Neighborhood filter, see [7], for $w_\rho \equiv 1$.
- The Yaroslavsky filter [40], for $w_\rho(|\mathbf{x} - \mathbf{y}|) \equiv \chi_{B_\rho(\mathbf{x})}(\mathbf{y})$, the characteristic function of a ball centered at $\mathbf{x}$ of radios ρ.
- The SUSAN [34] or Bilateral filters [36], for $w_\rho(s) = \exp(-(s/\rho)^2)$.

Other related filters which can be easily included in our discussion are the joint or cross bilateral filters, see [27, 13].

Nonlocal filters have been introduced in the last decades as alternatives to local methods such as those expressed in terms of nonlinear diffusion partial

* The authors are partially supported by the Spanish DGI Project MTM2013-43671-P.

differential equations (PDE), among which the pioneering approaches of Perona and Malik [26], Álvarez, Lions and Morel [2] and Rudin, Osher and Fatemi [32] are fundamental. We refer the reader to [9] for a review of these methods.

Nonlocal filters have been mathematically analyzed from different points of view. For instance, Barash [4], Elad [14], Barash et al. [5], and Buades et al. [8] investigate the asymptotic relationship between the Yaroslavsky filter and the Perona-Malik equation. Gilboa et al. [20] study certain applications of nonlocal operators to image processing. In [28], Peyré establishes a relationship between nonlocal filtering schemes and thresholding in adapted orthogonal basis. In a more recent paper, Singer et al. [33] interpret the Neighborhood filter as a stochastic diffusion process, explaining in this way the attenuation of high frequencies in the processed images.

From the computational point of view, until the reformulation given by Porikli [29], their actual implementation was of limited use due to the high computational demand of the direct space-range discretization. Only window-sliding optimization, like that introduced by Weiss [37] to avoid redundant kernel calculations, or filter approximations, like the one introduced by Paris and Durand [25], were of computational use. In [25], the space and range domains are merged into a single domain where the bilateral filter may be expressed as a linear convolution, followed by two simple nonlinearities. This allowed the authors to derive simple down-sampling criteria which were the key for filtering acceleration.

However, in [29], the author introduced a new *exact* discrete formulation of the bilateral filter for spatial box kernel (Yarsolavsky filter) using the local histograms of the image, $h_{\mathbf{x}} = h|_{B_\rho(\mathbf{x})}$, where $B_\rho(\mathbf{x})$ is the box of radious ρ centered at pixel $\mathbf{x}$, arriving to the formula

$$\mathrm{F}\, u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \sum_{i=1}^{n} q_i h_{\mathbf{x}}(q_i) \mathcal{K}_h(u(\mathbf{x}) - q_i), \tag{1}$$

where the range of summation is over the quantized values of the image, $q_1, \ldots, q_n$, instead of over the pixel spatial range. In addition, a zig-zag pixel scanning technique was used so that the local histogram is actualized only in the borders of the spatial kernel box.

Formula (1) is an exact formulation of the box filter where all the terms but the local histogram may be computed separately in constant time, and it is therefore referred to as a *constant time* $O(1)$ *method*.

Unfortunately, the use of local histograms is only valid for constant-wise spatial kernels, and subsequent applications of the new formulation to general spatial kernels is, with the exception of polynomial and trigonometric polynomial kernels, only approximated. Thus, in [29] polynomial approximation was used to deal with the usual spatial Gaussian kernel. This idea was improved in [11] by using trigonometric expansions.

In [38], Yang et al. introduced a new $O(1)$ method capable of handling arbitrary spatial and range kernels, as an extension of the ideas of Durand et al [12]. They use the so-called Principle Bilateral Filtered Image Component J_k, given

by, for $u(\mathbf{x}) = q_k$, and for some neighborhood of $\mathbf{x}$, $N(\mathbf{x})$,

$$J_{q_k}(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in N(\mathbf{x})} \mathcal{K}(q_k - u(\mathbf{y}))w_\rho(|\mathbf{x}-\mathbf{y}|)u(\mathbf{y})}{\sum_{\mathbf{y}\in N(\mathbf{x})} \mathcal{K}(q_k - u(\mathbf{y}))w_\rho(|\mathbf{x}-\mathbf{y}|)}. \tag{2}$$

Then, the bilateral filter may be expressed as $\mathrm{F}\, u(\mathbf{x}) = J_{u(\mathbf{x})}(\mathbf{x})$. In practice, only a subset of the range values is considered, and the final filtered image is produced by linear interpolation. In this situation this filter is an approximation to the bilateral filter. The same authors have recently extended and optimized [39] the method of Paris et al. [25] by solving cost volume aggregation problems. Other approaches may be found in [1, 22, 31].

2 Nonlocal Filters in Terms of Functional Rearrangements

Apart from the pure mathematical interest, the reformulation of nonlocal filters in terms of functional rearrangements is useful for computational purposes, specially when the spatial kernel, w, is homogeneous, that is $w_\rho \equiv 1$. In this case, it may be proven [18] that the level sets of u are invariant through the filter, i.e. $u(\mathbf{x}) = u(\mathbf{y})$ implies $F(u)(\mathbf{x}) = F(u)(\mathbf{y})$, and thus, it is sufficient to compute the filter only for each (quantized) level set, instead of for each pixel, meaning a huge gain of computational effort.

For non-homogeneous kernels the advantages of the filter rearranged version are kernel-dependent, and in any case, the gain is never comparable to that of homogeneous kernels. The main reason is that the non-homogeneity of the spatial kernel breaks, in general, the invariance of level sets.

In the following lines, for a smoother introduction of the bilateral filter rearranged version, we provide a formal derivation deduced from the coarea formula. However, notice that the resulting formula is valid in a more general setting, see [19, Theorem 1]. In particular, the condition $\nabla u(\mathbf{y}) \neq 0$ used below is seen to be not necessary.

For this task, we recall the notion of *decreasing rearrangement*, $u_* : [0, |\Omega|] \to \mathbb{R}$, of a function $u : \Omega \subset \mathbb{R}^d \to \mathbb{R}$, which is defined [21, 23] as the (generalized) inverse of the *distribution function of* u, given by $m_u(q) = |\{\mathbf{x} \in \Omega : u(\mathbf{x}) > q\}|$, for $q \in \mathbb{R}$, where $|\cdot|$ denotes the Lebesgue measure.

Under suitable regularity assumptions, the coarea formula states

$$\int_\Omega g(\mathbf{y})|\nabla u(\mathbf{y})|d\mathbf{y} = \int_{-\infty}^{\infty}\int_{u=t} g(\mathbf{y})d\Gamma(\mathbf{y})dt,$$

where we used the notation $u = t$ to denote the set $\{\mathbf{x} \in \Omega : u(\mathbf{x}) = t\}$. Taking $g(\mathbf{y}) = \mathcal{K}_h(u(\mathbf{x}) - u(\mathbf{y}))w_\rho(|\mathbf{x}-\mathbf{y}|)u(\mathbf{y})/|\nabla u(\mathbf{y})|$, and using $u(\mathbf{x}) \in [0, Q]$ for all $\mathbf{x} \in \Omega$ we get, for the numerator of the filter F,

$$\begin{aligned} I(\mathbf{x}) &:= \int_\Omega \mathcal{K}_h(u(\mathbf{x}) - u(\mathbf{y}))w_\rho(|\mathbf{x}-\mathbf{y}|)u(\mathbf{y})d\mathbf{y} \\ &= \int_0^Q \mathcal{K}_h(u(\mathbf{x}) - t)t \int_{u=t} \frac{w_\rho(|\mathbf{x}-\mathbf{y}|)}{|\nabla u(\mathbf{y})|}d\Gamma(\mathbf{y})dt. \end{aligned}$$

Assuming that the decreasing rearrangement of u, u_*, is in fact strictly decreasing and introducing the change of variable $t = u_*(s)$ we find

$$\begin{aligned} I(\mathbf{x}) &= -\int_0^{|\Omega|} \mathcal{K}_h(u(\mathbf{x}) - u_*(s))u_*(s)\frac{du_*(s)}{ds}\int_{u=u_*(s)} \frac{w_\rho(|\mathbf{x}-\mathbf{y}|)}{|\nabla u(\mathbf{y})|}d\Gamma(\mathbf{y})ds \\ &= \int_0^{|\Omega|} \mathcal{K}_h(u(\mathbf{x}) - u_*(s))u_*(s)w_\rho(|\mathbf{x}-\cdot|)_{*u}(s)ds. \end{aligned}$$

Here, the notation v_{*u} stands for the *relative rearrangement of v with respect to u* [30] which, under regularity conditions, may be expressed as

$$v_{*u}(s) = \frac{\displaystyle\int_{u=u_*(s)} \frac{v(\mathbf{y})}{|\nabla u(\mathbf{y})|}d\Gamma(\mathbf{y})}{\displaystyle\int_{u=u_*(s)} \frac{1}{|\nabla u(\mathbf{y})|}d\Gamma(\mathbf{y})}. \tag{3}$$

Transforming the denominator of the filter, $C(\mathbf{x})$, in a similar way we get

$$\mathrm{F}\, u(\mathbf{x}) = \frac{\int_0^{|\Omega|} \mathcal{K}_h(u(\mathbf{x}) - u_*(s))w_\rho(|\mathbf{x}-\cdot|)_{*u}(s)u_*(s)ds}{\int_0^{|\Omega|} \mathcal{K}_h(u(\mathbf{x}) - u_*(s))w_\rho(|\mathbf{x}-\cdot|)_{*u}(s)ds}. \tag{4}$$

Remark 1. The relative rearrangement is defined in its full generality as the weak-$L^p(\Omega_*)$ directional derivative

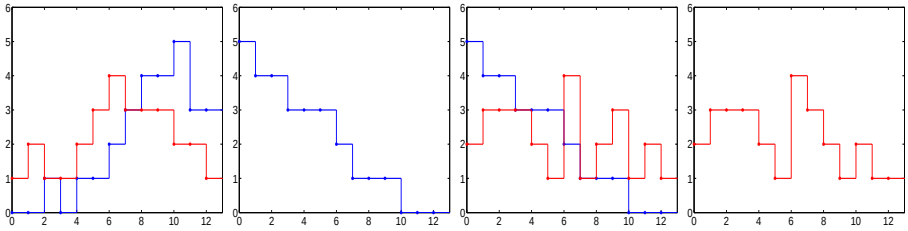
$$v_{*u} = \lim_{t\to 0} \frac{(u+tv)_* - u*}{t}. \tag{5}$$

Under the additional assumptions $u \in W^{1,1}(\Omega)$ (that is, $u \in L^1(\Omega)$ and $\nabla u \in L^1(\Omega)^d$), and $|\{\mathbf{y} \in \Omega : \nabla u(\mathbf{y}) = 0\}| = 0$, i.e. the non-existence of flat regions of u, the identity (3) is well defined and coincides with (5). In this case, the relative rearrangement represents an averaging procedure of the values of v on the level sets of u labeled by the superlevel sets measures, s. When formula (3) does not apply (flat regions of u) we may resort to (5) to interpret the relative rearrangement as the decreasing rearrangement of v restricted to flat regions.

Example: Rearrangement of constant-wise functions. Consider the constant-wise functions $u, v : [0, 13] \to \mathbb{R}$ given in Fig. 1 (a). Writing $\max(u) = 5 = q_1 > \ldots > q_6 = 0 = \min(u)$, we may express u as $u(x) = \sum_{i=1}^6 q_i \chi_{E_i}(x)$, where E_i are the level sets of u, $E_1 = (10, 11]$, $E_2 = (8, 10]$, etc. Then, the decreasing rearrangement of u is constant-wise too, and given by

$$u_*(s) = \sum_{i=1}^n q_i \chi_{I_i}(s),$$

with $I_i = [a_{i-1}, a_i)$ for $i = 1, \ldots, 6$, and with $a_0 = 0$, $a_1 = |E_1| = 1$, $a_2 = |E_1| + |E_2| = 3$,...,$a_6 = \sum_{i=1}^6 |E_i| = |\Omega| = 13$. The corresponding plot is shown in Fig. 1 (b).



(a) u (blue), v (red) (b) u_* (c) (d) v_{*u}

Fig. 1. Example of construction of the relative rearrangement. (c) shows u_* (blue) and v transported as by the displacement of u level sets (red). (d) shows the decreasing rearrangement of v restricted to the level sets of u, that is v_{*u}.

In Fig. 1 (c) we show the graphs $\{E_i, v(E_i)\}_{i=1}^6$ transported as it was done in the step before to construct u_*. For instance, the highest level set of u, $E_1 = (10, 11]$, was transported to $[0, 1]$; $E_2 = (8, 10]$ to $(1, 3]$, etc. Thus $\{E_1, v(E_1)\} = \{(10, 11], \{2\}\}$ is transported to $\{[0, 1], \{2\}\}$; $\{E_2, v(E_2)\} = \{(8, 10], \{3\}\}$ is transported to $\{[1, 3], \{3\}\}$, etc.

Finally, to obtain the decreasing rearrangement of v with respect to u, v_{*u}, we rearrange decreasingly the restriction of v to E_i, as shown in Fig. 1 (d).

2.1 Discrete Setting

To gain some insight into formula (4), let us consider a constant-wise interpolation of a given image, u, quantized in n levels labeled by q_i, with $\max(u) = q_1 > \ldots > q_n = 0$. That is $u(\mathbf{x}) = \sum_{i=1}^n q_i \chi_{E_i}(\mathbf{x})$, where E_i are the level sets of u,

$$E_i = \{\mathbf{x} \in \Omega : u(\mathbf{x}) = q_i\}, \quad i = 1, \ldots, n.$$

Similarly, let w_ρ be a constant-wise interpolation of the spatial kernel quantized in m levels, r_j, with $\max(w_\rho) = r_1 > \ldots > r_m = \min(w_\rho) \geq 0$. For each $\mathbf{x} \in \Omega$, consider the partition of E_i given by $F_j^i(\mathbf{x}) = \{\mathbf{y} \in E_i : w_\rho(|\mathbf{x}-\mathbf{y}|) = r_j\}$. Then, it may be shown [19] that for each $\mathbf{x} \in E_k$, $k = 1, \ldots, n$,

$$\mathrm{F}\, u(\mathbf{x}) \equiv \mathrm{F}_*\, u(\mathbf{x}) = \frac{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i) W_{im}(\mathbf{x}) q_i}{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i) W_{im}(\mathbf{x})}, \tag{6}$$

where $W_{im}(\mathbf{x}) = \sum_{j=1}^m r_j |F_j^i(\mathbf{x})|$. Here, $|F_j^i(\mathbf{x})|$ is the number of pixels in the q_i-level set of u that belongs to the r_j-level set of w_ρ. We refer to $F_* u(\mathbf{x})$ as to the *rearranged version* of $Fu(\mathbf{x})$.

2.2 Examples

The main difficulty for the computation of formula (6) is the determination of the measures of $F_j^i(\mathbf{x})$ which, in general, must be computed for each $\mathbf{x} \in \Omega$.

The Neighborhood filter. In this case, $w_\rho \equiv 1$, and therefore $m = 1$ and $F_1^i(\mathbf{x}) = E_i$ is independent of $\mathbf{x}$ for all $i = 1, \ldots, n$. Thus, formula (6) is computed only on the level sets of u, that is, for all $\mathbf{x} \in E_k$

$$F_* u(\mathbf{x}) = \frac{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i)|E_i|q_i}{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i)|E_i|}. \tag{7}$$

The Yaroslavsky or box filter. In this case, $w_\rho(|\mathbf{x}-\mathbf{y}|) = \chi_{B_\rho(\mathbf{x})}(\mathbf{y})$, and therefore there are only two levels $r_1 = 1,\ r_2 = 0$ of w_ρ corresponding to the sets $F_1^i(\mathbf{x}) = E_i \cap B_\rho(\mathbf{x})$ and $F_2^i(\mathbf{x}) = E_i \cap B_\rho(\mathbf{x})^c$. Thus, formula (6) reduces to: for each $\mathbf{x} \in E_k$, $k = 1, \ldots, n$

$$F_* u(\mathbf{x}) = \frac{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i)|F_1^i(\mathbf{x})|q_i}{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i)|F_1^i(\mathbf{x})|},$$

where $|F_1^i(\mathbf{x})|$ denotes the number of pixel of the i-level set of u which belong to the box $B_\rho(\mathbf{x})$.

The general bilateral filter. In this case, the full formula (6) must be used. The spatial kernel is a smooth varying function, e.g. a Gaussian, and therefore there is a continuous range of levels of w_ρ. For computational purposes the range of w_ρ is quantized to some finite number of levels, determined by the size of ρ and the machine ϵ.

The bilateral filter may be accelerated by manipulating the quantization levels of the image, and/or of the spatial range. As shown in [38] for the Yaroslavsky filter, the reduction of the image quantization levels leads to poor denoising results. However, we checked that a similar restriction applied to the spatial kernel reduces the execution time while conserving good denoising quality.

Finally, observe that in all these rearranged versions the range kernel $\mathcal{K}_h(u(\mathbf{x})-u(\mathbf{y}))$ is transformed into $\mathcal{K}_h(q_k - \mathbf{q})$ which, for coding, may be computed and stored outside the main loop running over all the pixels.

2.3 The Special Case of the Neighborhood Filter

The Neighborhood filter, that is the bilateral filter for a homogeneous spatial kernel $w_\rho \equiv 1$, is a special case from the rearranged version point of view. Formula (7) shows that in this case all the pixels of a given level set, E_i, are jointly filtered to the same value, producing a huge gain in execution time.

Moreover, the usual iterative scheme under which this filter is employed

$$u_{j+1}(\mathbf{x}) = \frac{1}{C_j(\mathbf{x})} \int_\Omega \mathcal{K}_h(u_j(\mathbf{x}) - u_j(\mathbf{y}))u_j(\mathbf{y})d\mathbf{y},$$

with $C_j(\mathbf{x}) = \int_\Omega \mathcal{K}_h(u_j(\mathbf{x})-u_j(\mathbf{y}))d\mathbf{y}$, also satisifes this level sets invariant structure property [17, 18], from where the following recurrent formula is deduced: For $u_0 = u$ (initial image) and for all $\mathbf{x} \in E_k^j \{\mathbf{x} \in \Omega : u_n(\mathbf{x}) = q_k\}, \quad j = 0, 1, \ldots,$

$$u_{j+1}(\mathbf{x}) = \frac{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i)|E_i^j|q_i}{\sum_{i=1}^n \mathcal{K}_h(q_k - q_i)|E_i^j|}.$$

As shown in [18], a notable property of this recurrent formula is the formation of large gradients around inflexion points of u_*, like in some type of shock filters [3]. Using the connection between the image histogram, $h_u(q)$ and the distribution function, $m_u(q)$, given by $m_u(q) = \int_q^{\max u} h(s)ds$, we see that critical points of the histogram coincides with inflexion points of the distribution function and, hence, of the decreasing rearrangement, u_*. Since histogram critical points detection is the base for some intensity based segmentation algorithms [10, 24], the iterated Neighborhood filter may be used as an *automatic* segmentation algorithm, in which the only tunning parameter is the window size, h. We provide some examples in the next section.

3 Experiments

3.1 Denoising

We conducted an experiment on standard natural images to check the performance of the discrete formula (6) in comparison to the *brute force* pixel based implementation of the bilateral filter, and to a well known state of the art denoising algorithm introduced by Yang et al. [38], see formula (2).

Both formula (6) and Yang's et al. algorithm are exact representations of the bilateral filter when the maximum number of spatial kernel quantization levels or the whole image range, respectively, are considered. Therefore, in this case, the only source of disagreement is caused by rounding error.

Thus, we experimented with smaller values of these parameters for the sake of execution time saving. We chose twenty values for the discretization of the bilateral spatial kernel (Gaussian) in the rearranged formula (6), and eight values for Yang's et al. algorithm, like in [38].

The first aim of our experiment was to investigate the quality of the algoruthms approximation, in terms of the peak signal to noise ratio (PSNR), to the ground truth, and to the exact pixel-based bilateral filter. The second, was the comparison of execution times as delivered straightly from the available codes. Notice that execution time depends on code optimization and therefore a rigorous study of this aspect requires some kind of code normalization which was out of the scope of our study.

We used three intensity images of different sizes corrupted with an additive Gaussian white noise of SNR = 10, according to the noise measure SNR = $\sigma(u)/\sigma(\nu)$, where σ is the empirical standard deviation, u is the original image, and ν is the noise. The images, available at the data base of the Signal and Image Processing Institute, University of Southern California, are *Clock* (256×256), *Boat* (512×512), and *Airport* (1024×1024).

We considered different spatial window sizes determined by ρ, with $\rho = 4\ , 8,\ 16,\ 32$. The range size of the filter was taken as $h = \rho$ which, according to [8], is the regime in which the corresponding iterative filter behaves asymptotically as a Perona-Malik type filter. The shape of the range filter is a Gaussian.

The discretization of the pixel-based and the rearranged version of both filters was implemented in non-optimized C++ codes by the authors, while for Yang's

algorithm available code[1] was employed. Time execution was measured by means of function `clock`.

In Table 1 we show the measures resulting from our experiments. We see that all the algorithms give similar results when compared to the ground truth. Thus, if this were the choice criterium, the faster, that is $Y8$, should be considered.

However, when compared to the exact bilateral filter (BPB), the PSNR's are quite different. The rearranged bilateral filter with twenty spatial kernel levels (BRR20) has always values of PSNR around 40dB, which makes it indistinguishable from the exact filtered image. The Yaroslavsky filter (YRR) lowers this figure to about 20dB. Yang's et al. with eight range values (Y8) gives always poorer results. In fact, the use of Yang's algorithm with the maximum number of levels (not shown in the table), although *should* give exact results, it does not, revealing other sources of error beyond rounding errors.

In Table 1 we also collect the execution times obtained in this experiment. Only for the smaller image sizes and $h-$values Y8 has a competitor in YRR. BRR20 gives execution times considerably higher than the other algorithms, for our non-optimized codes.

3.2 Segmentation with the Neighborhood Filter

To demonstrate the capability of the Neighborhood filter as a segmentation algorithm, we applied it to MRI brain segmentation. We used a phantom brain from the Simulated Brain Database [6] with a 9% of additive Riccian noise and compared the result to the grey-white matter segmentation performed with some standard packages: Freesurfer [15], FSL [16] and SPM8 [35]..

In Fig. 2, rows 1 to 3, we show an axial slice of the volume (initial image) and the corresponding segmentation in four, three and two regions reached by setting $h = 17,\ 20,\ 50$, respectively. The contour lines and the decreasing rearrangement of the original image, u, and the final filtered image are shown too. In rows 4 to 5, we show the masks of the segmented regions corresponding to $h = 17,\ 20$.

To check the Neighborhood filter segmentation performance, the Dice coincidence coefficient is computed for all the algorithms, see Table 2, showing a good performance of the Neighborhood filter in relation to the more sophisticated algorithms implemented in the mentioned packages. Notice that the Dice coefficient is one for a perfect match to the ground truth, and zero on the contrary.

Although we have shown the results for one slice, the Neighborhood filter is applied directly to the whole volume, meaning that the dimension reduction is from a three dimensional space (the space of voxels) to a one dimensional space (the space of level sets measures). Thus, the time execution of the NF is several orders of magnitude lower than the others (a standard volume takes few seconds in a standard laptop). However, this is no more than a toy example, from where general conclusions can not be inferred.

[1] C++ code in `http://www.cs.cityu.edu.hk/ qiyang/`

4 Summary

In this paper we used functional rearrangements to express bilateral type filters in terms of integral operators in the one-dimensional space $[0, |\Omega|]$.

In the case in which the spatial kernel, w_ρ, is homogeneous (e.g. the Neighborhood filter), the level set structure of the image is left invariant through the filtering process, allowing to compute the filter jointly for all the pixels in each level set, instead of pixel-wise. This can be done also for the associated iterative scheme, which is seen to be related to intensity-based histogram-related image segmentation.

If the spatial kernel is not homogeneous the invariance of the $u-$level sets through the filter is, in general, broken. Despite this fact, there still remains an important property of the rearranged version: the range kernel $\mathcal{K}_h(u(\mathbf{x}) - u(\mathbf{y}))$ is transformed into a pixel-independent kernel $\mathcal{K}(q_k - q_i)$, implying a large gain in computational effort, as already observed for particular cases in [29].

References

1. Adams A., Baek J, Davis M.A.: Fast high-dimensional filtering using the permutohedral lattice. Computer Graphics Forum,29(2), 753–762 (2010)
2. Álvarez L., Lions P.L., Morel J.M.: Image selective smoothing and edge detection by nonlinear diffusion. ii. Siam J Numer Anal 29(3), 845–866 (1992)
3. Álvarez L., Mazorra L.: Signal and image restoration using shock filters and anisotropic diffusion. Siam J Numer Anal 31(2), 590–605 (1994)
4. Barash D.: Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. IEEE T Pattern Anal 24(6), 844–847 (2002)
5. Barash D., Comaniciu D.: A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift. Image Vision Comput 22(1), 73–81 (2004)
6. Brainweb, `http://brainweb.bic.mni.mcgill.ca/brainweb`, Vol.`t1_icbm_normal_1mm_pn9_rf20_resampled_brain.nii.gz`
7. Buades A., Coll B., Morel J.M.: A review of image denoising algorithms, with a new one. Multiscale Model Sim 4(2), 490–530 (2005)
8. Buades A., Coll B., Morel J.M.: Neighborhood filters and pde's. Numer Math 105(1), 1–34 (2006)
9. Buades A., Coll B., Morel J.M.: Image denoising methods. a new nonlocal principle. Siam Rev 52(1), 113–147 (2010)
10. Chang C.I., Du Y., Wang J., Guo S.M., Thouin P.: Survey and comparative analysis of entropy and relative entropy thresholding techniques. In: Vision, Image and Signal Processing, IEE Proceedings-, IET, 153, 837–850 (2006)
11. Chaudhury K.N., Sage D., Unser M.: Fast O(1) bilateral filtering using trigonometric range kernels. TIP 2011 (2011)
12. Durand F., Dorsey J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM Siggraph 21, 257–266 (2002)
13. Eisemann E., Durand F. Flash photography enhancement via intrinsic relighting. Siggraph 23(3), 673–678 (2004)

14. Elad M.: On the origin of the bilateral filter and ways to improve it. IEEE T Image Process 11(10), 1141–1151 (2002)
15. Freesurfer `http://freesurfer.net`
16. FMRIB Software Library `http://fsl.fmrib.ox.ac.uk/fsl/fslwiki`
17. Galiano G., Velasco J.: On a non-local spectrogram for denoising one-dimensional signals. Appl Math Comput 244, 859-869 (2014)
18. Galiano G., Velasco J.: Neighborhood filters and the decreasing rearrangement. J Math Imaging Vision, DOI: 10.1007/s10851-014-0522-3 (2014)
19. Galiano G., Velasco J.: On a fast bilateral filtering formulation using functional rearrangements. Submitted. arXiv:1406.7128v1 (2015)
20. Gilboa G., Osher S.: Nonlocal operators with applications to image processing. Multiscale Model Sim 7(3), 1005–1028 (2008)
21. Hardy G.H., Littlewood J.E., Polya G.: Inequalities. Cambridge U.P. (1964)
22. Kass M., Solomon J.: Smoothed local histogram filters. ACM TOG 29(4), 100:1–100:10 (2010)
23. Lieb E.H., Loss M.: Analysis, vol 4. American Mathematical Soc. (2001)
24. Nath S., Agarwal S., Kazmi Q.A.: Image histogram segmentation by multi-level thresholding using hill climbing algorithm. Int J Comput Appl 35(1) (2011)
25. Paris S., Durand F.: A fast approximation of the bilateral filter using a signal processing approach. ECCV 2006 (2006)
26. Perona P., Malik J.: Scale-space and edge detection using anisotropic diffusion. IEEE T Pattern Anal 12(7), 629–639 (1990)
27. Petschnigg G., Agrawala M., Hoppe H., Szeliski R., Cohen M., Toyama K.: Digital photography with flash and no-flash image pairs. ACM Siggraph, 23(3), 664–672 (2004)
28. Peyré G.: Image processing with nonlocal spectral bases. Multiscale Model Sim 7(2), 703–730 (2008)
29. Porikli F.: Constant time O(1) bilateral filtering. CVPR 2008 (2008)
30. Rakotoson J.M.: Réarrangement Relatif: Un instrument d'estimations dans les problčmes aux limites, vol 64. Springer (2008)
31. Ram I., Elad M., Cohen I.: Image processing using smooth ordering of its patches. IEEE T Image Process 22(7), 2764–2774 (2013)
32. Rudin L.I., Osher S., Fatemi E.: Nonlinear total variation based noise removal algorithms. Physica D 60(1), 259–268 (1992)
33. Singer A., Shkolnisky Y., Nadler B.: Diffusion interpretation of nonlocal neighborhood filters for signal denoising. SIAM J Imaging Sci 2(1), 118–139 (2009)
34. Smith S.M., Brady J.M.: Susan. a new approach to low level image processing. Int J Comput Vision 23(1), 45–78 (1997)
35. Statistical Parametric Mapping (SPM) `http://www.fil.ion.ucl.ac.uk/spm`
36. Tomasi C., Manduchi R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision, 1998, IEEE, pp 839–846 (1998)
37. Weiss B.: Fast median and bilateral filtering. ACM Siggraph 25, 519–526 (2006)
38. Yang Q., Tan K.H., Ahuja N.: Real-time O(1) bilateral filtering. CVPR 2009 (2009)
39. Yang Q., Ahuja N., Tan K.H.: Constant time median and bilateral filtering. Int J Comput Vis DOI 10.1007/s11263-014-0764-y (2014)
40. Yaroslavsky L.P.: Digital picture processing. An introduction. Springer Verlag, Berlin (1985)

	GT				BPB			ET				ETBPB		
Clock (256×256)														
h	BPB	BRR	Y8	YRR	BRR	Y8	YRR	BPB	BRR	Y8	YRR	BRR	Y8	YRR
4	3.62	3.62	-1.24	3.64	42.6	1.17	29.7	0.56	0.17	0.02	0.02	3.29	28	28
8	4.08	4.08	3.89	4.03	43.1	21.3	27.2	2.06	0.25	0.02	0.02	8.24	103	103
16	4.4	4.4	4.1	4.31	41.5	20.3	21.1	7.98	0.62	0.02	0.03	12.9	399	266
32	4.31	4.31	4.04	3.55	38.4	16	12.4	31.41	1.8	0.02	0.06	17.4	1570	524
Boat (512×512)														
4	10.6	10.6	4.36	10.6	42.7	5.26	29.4	2.19	0.68	0.1	0.06	3.22	21.9	36.5
8	11	11	10.5	10.7	42.1	20.6	26	8.33	1.12	0.06	0.08	7.44	138.8	104
16	9.23	9.23	8.8	8.69	40.7	18.3	20.3	32.64	2.44	0.06	0.14	13.4	544	233
32	4.83	4.83	4.22	3.95	38.4	15.8	14.2	132.1	7.27	0.06	0.22	18.2	2201	600
Airport (1024×1024)														
4	3.83	3.83	3.15	3.83	42.6	11	29.4	8.76	2.78	0.24	0.23	3.15	36.5	38.1
8	3.89	3.89	3.85	3.87	42.2	21.7	26.7	33.37	4.29	0.22	0.3	7.78	151.7	111
16	3.6	3.59	3.59	3.59	41.1	17	21.7	132.8	9.36	0.22	0.58	14.2	603.8	229
32	2.45	2.44	2.35	2.38	38.7	11.5	15.5	529.5	28.1	0.2	0.92	18.8	2648	576

Table 1. Denoising experiment. From left to right. GT block: PSNR between the ground truth image and the pixel-based Bilateral filter (BPB), its rearranged version for 20 levels (BRR), Yang's algorithm with 8 interpolation elements (Y8), and the Yaroslavsky filter in its rearranged version (YRR). BPB block: PSNR between the BPB and the other algorithms. ET block: Execution times of BPB, BRR, Y8, and YRR. ETBPB block: Ratio between execution time of the algorithm and execution time of BPB.

	Dice coefficient			
	Freesurfer	FSL	SPM	NF
white	0.9490	0.9435	0.9468	0.9563
grey	0.8509	0.8599	0.8835	0.8797

Table 2. Segmentation experiment. Comparison among several algorithms.

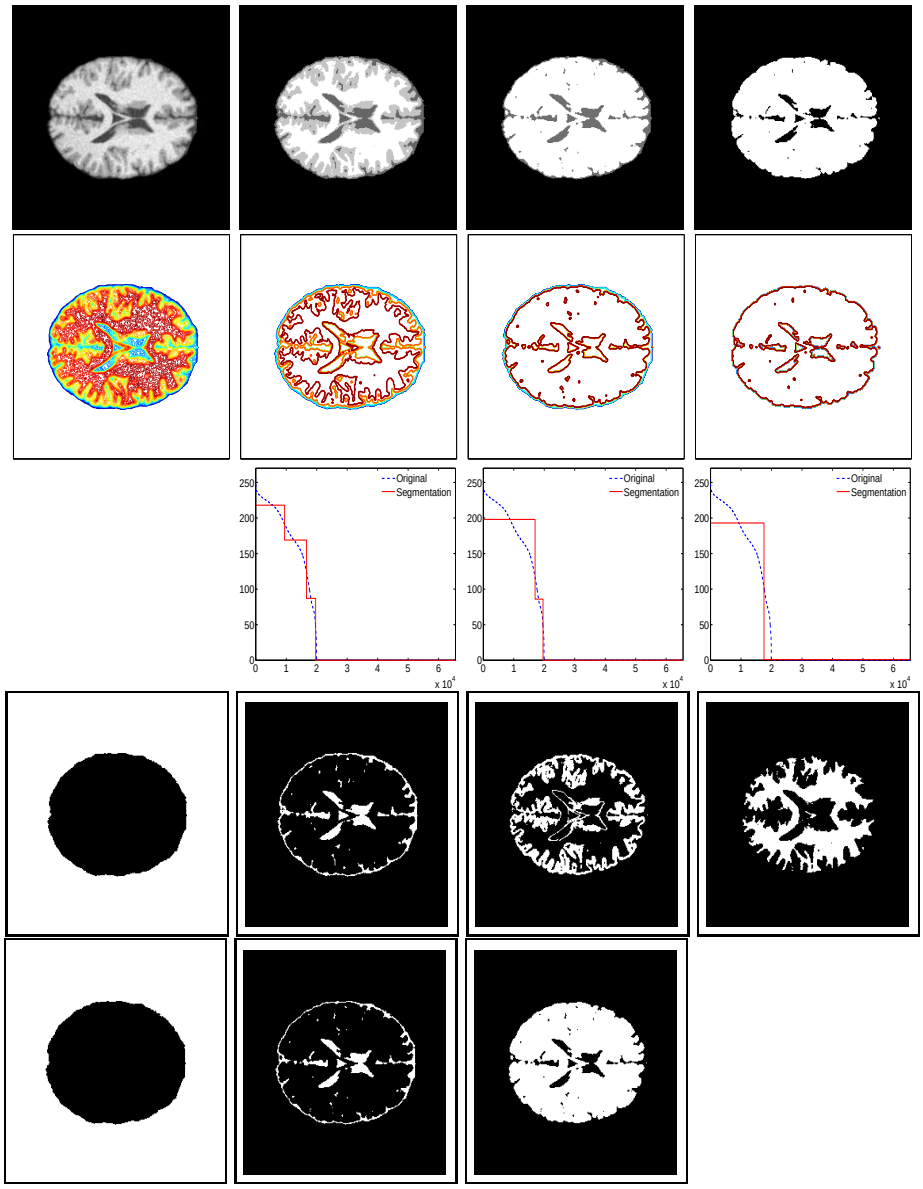


Fig. 2. Segmentation experiment. Results of applying the Neighborhood filter (NF) with several values of the window size h. Rows 1-3: Image, level curves and decreasing rearrangement both of the original image and of the final result, showing the number of segmented regions (flat regions). Columns for rows 1-3: Results of applying the NF with $h = 17$, $h = 20$, and $h = 50$, respectively. Rows 4-5: Masks of the segmented regions. Row 4: $h = 17$. The NF produces four regions, corresponding to background, dura-mater and ventricles, grey matter and white matter. Row 5: $h = 20$. The NF produces three regions, corresponding to background, duramatter and ventricles, grey plus white matter.