Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Patch-wise adaptive weights smoothing

Jörg Polzehl, Kostas Papafitsoros, Karsten Tabelow

submitted: July 16, 2018

Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: joerg.polzehl@wias-berlin.de
kostas.papafitsoros@wias-berlin.de
karsten.tabelow@wias-berlin.de

No. 2520
Berlin 2018

2010 *Mathematics Subject Classification.* 62G05, 62G99, 62P10, 65K10.

Key words and phrases. Image denoising, patch wise structural adaptive smoothing, total variation, non-local means.

Jörg Polzehl would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme Variational methods and effective algorithms for imaging and vision when work on this paper was undertaken. This work was supported by EPSRC grant numbers EP/K032208/1 and EP/R014604/1. Kostas Papafitsoros acknowledges the support of the Einstein Foundation Berlin within the ECMath project CH12.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Patch-wise adaptive weights smoothing

Jörg Polzehl, Kostas Papafitsoros, Karsten Tabelow

ABSTRACT. Image reconstruction from noisy data has a long history of methodological development and is based on a variety of ideas. In this paper we introduce a new method called patch-wise adaptive smoothing, that extends the Propagation-Separation approach by using comparisons of local patches of image intensities to define local adaptive weighting schemes for an improved balance of reduced variability and bias in the reconstruction result. We present the implementation of the new method in an R-package **aws** and demonstrate its properties on a number of examples in comparison with other state-of-the art image reconstruction methods.

1. INTRODUCTION

One of the problems in image processing is the reduction of noise, that is often inherently connected with the image acquisition process, deteriorates the image quality and hinders image analysis. While the notion of images in general refers to two-dimensional data, the problem of noise reduction also occurs in connection with data in higher dimensions, especially in the context of medical imaging problems.

There is a vast literature on different noise reduction techniques. They typically employ an assumption on the spatial structure of the imaging data. A common and simple assumption is, e.g., that the data is characterized by spatially extended regions of homogeneity that are separated by discontinuities. A more sophisticated assumption replaces local homogeneity by local smoothness (Polzehl and Spokoiny 2008; Polzehl and Tabelow 2012). Alternatively, geometric characterizations using orientation spaces or channels in feature space are discussed, e.g., in Felsberg 2012; Felsberg, Forssén, and Scharr 2006; Florack 2012; Duits, Fuehr, and Janssen 2012; Franken 2008.

Typically, noise reduction methods must balance the variability reduction and the bias of the reconstruction results, realizing some kind of edge preservation. They are based on a large variety of basic methodology, achieving this goal to a different degree. They range from kernel smoothing and local polynomials (Wand and Jones 1995; Fan and Gijbels 1996; Simonoff 1996; Bowman and Azzalini 1997), the filter proposed by Lee 1980, bilateral filtering (Tomasi and Manduchi 1998), scale space methods (Chaudhuri and Marron 2000), to non-linear diffusion methods (Perona and Malik 1990; Scharr and Krajsek 2012; Weickert 1998) in a rather incomplete list.

In this paper we consider a class of noise reduction methods that has been introduced under the name *adaptive weights smoothing* (AWS, Polzehl and Spokoiny 2000) and later refined as Propagation-Separation approach (PS, Polzehl and Spokoiny 2006), which generalizes several of the concepts above in non-parametric regression. PS has been extended to cover locally smooth images (Polzehl and Spokoiny 2008) or color images (Polzehl and Tabelow 2007). Furthermore, it has been successfully applied to a number of imaging problems in neuroimaging, e.g., in functional Magnetic Resonance Imaging (fMRI, Tabelow et al. 2006; Polzehl, Voss, and Tabelow 2010) or in Diffusion-weighted Magnetic Resonance Imaging (dMRI, Tabelow et al. 2008; Becker et al. 2012; Becker et al. 2014). PS combines local comparisons of image intensities to define adaptive weighting schemes with a multiscale approach which iteratively inspects scale space from very local to large scales. In this paper, we will present the implementation of PS within the R-package **aws**.

Buades, Coll, and Morel 2005 introduced an adaptive denoising method also extending some of the filtering concepts above. Instead of comparing pairs of single local image intensities to define adaptive weighting schemes, it uses non-local comparisons of larger patches of intensities. Denoted as non-local means (NLM) it has been successfully applied to many imaging problems, e.g., denoising MRI data (Manjón et al. 2009). The purpose of this paper is to combine the strength of the multiscale

DOI 10.20347/WIAS.PREPRINT.2520 Berlin 2018

approach of PS and the patch-wise comparison of image intensities of NLM to present a new algorithm which we call *patch-wise adaptive smoothing* (PAWS). We will demonstrate its performance on a series of two- and three-dimensional images. This will show that PAWS overcomes two of the major drawbacks of AWS (or PS), the occurrence of artificial structure within areas of smooth intensity changes and non-smooth borders between neighboring regions with significantly different intensity.

Furthermore, we will compare the results with those obtained considering the image reconstruction as an energy minimizing problem with penalization, i.e., the total variation (TV, Rudin, Osher, and Fatemi 1992) and the total generalized variation (TGV, Bredies, Kunisch, and Pock 2010) approach. Even though these two approaches belong to a different family of denoising methods than those based on Propagation-Separation, the comparison is nevertheless very relevant. Indeed, there is a clear analogy of the relationship between TV and TGV on the one hand and AWS and PAWS on the other. As we will discuss in the next Section, AWS employs a structural assumption of piecewise constant image intensity for the noise-free data. This is very similar to TV regularization where the minimization of the ℓ_1 norm of the gradient leads to its sparsity and thus to piecewise constant reconstructions. Hence, TV and AWS suffer from the same type of artifacts, i.e., cartoon-like structures, which are undesirable when the underlying ground truth image contains also piecewise smooth parts. TGV is a higher order extension of TV which was introduced with the aim to eliminate these artifacts. This is done by incorporating an infimal convolution type combination of the ℓ_1 norms of first and second order derivatives resulting in piecewise affine reconstructions. As we see in the following sections, lpAWS and the new method PAWS improve AWS in a very similar way by imposing a locally smooth structure on the denoised image. Thus, they improve AWS in the same way that TGV improves TV, leading to more naturally looking denoised images.

The outline of the paper is as follows: We first review the basic principles of the PS approach and the NLM method. We then introduce the new PAWS algorithm. Furthermore, we shortly review the TV and TGV approaches before presenting extensive examples to demonstrate properties of the different methods in various situations. The new algorithm is implemented within the R-package **aws**, we will explain the corresponding functions.

2. Adaptive weights smoothing

Adaptive weights smoothing for the restoration of images from noisy data was originally introduced in Polzehl and Spokoiny 2000 and refined under the term Propagation-Separation approach in Polzehl and Spokoiny 2006. It employs a structural assumption on the data, more specifically, a local constant parameter model. It is designed as an iterative multi-scale approach that inspects scale space from local to global and simultaneously infers on both the parameter value and its spatial structure.

2.1. Local constant adaptive weights smoothing (AWS).

Let us assume that data $Y_i \in \mathcal{Y}$ is observed at positions $x_i = (x_{i_1}, \ldots, x_{i_d})$ in a bounded subset $\mathcal{X}$ of a d-dimensional metric space. We assume Y_i to be distributed as $Y_i \sim P_{\theta_i}$, where P_{θ_i}, with density $p(y, \theta_i)$, depends on some local parameter θ_i (typically from $\mathbb{R}^p$) and is a probability distribution with support in $\mathcal{Y}$ from some parametric (typically exponential) family $P_{\theta_i} \in \mathcal{P}_\Theta$.

The structural assumption is formulated such that there exists a partitioning $\mathcal{X} = \bigcup_{n=1,\ldots,N} \mathcal{X}_n$ into N subsets with $\mathcal{X}_n \cap \mathcal{X}_l = \emptyset$ if $n \neq l$ and $\theta_i \equiv \theta_j$ if $x_i \in \mathcal{X}_n$ and $x_j \in \mathcal{X}_n$ for some n. Literally speaking we assume that within any subset $\mathcal{X}_n$ the parameter θ as a function of x is constant.

The method employs both a distance $\delta(x_i, x_j)$ in design space $\mathcal{X}$ as well as a distance $\eta(\theta_i, \theta_j)$ in parameter space $\mathcal{Y}$. A common choice in case of $\mathcal{X} \subset \mathbb{R}^d$ is the Euclidean distance

$$\delta(x_i, x_j) = ||x_i - x_j||_2$$

Data: Observations $Y_i \in \mathcal{Y}$ at locations $x_i \in \mathcal{X}_G$.
Initialization: Set $k = 0$, $h^{(0)} = 1$, $w_{ij}^{(0)} = K_{\mathrm{loc}}\big(\delta(x_i, x_j)\big)$ and initialize $\hat{\theta}_i^{(0)}$ as weighted likelihood or least squares estimate;
while $k \leq k^\star$ **do**
For all locations i and j define

$$w_{ij}^{(k)} = K_{\mathrm{loc}}\big(l_{ij}^{(k)}\big) K_{\mathrm{st}}\big(s_{ij}^{(k)}\big)$$

with

$$l_{ij}^{(k)} = \big(\delta(x_i, x_j)/h^{(k)}\big)^2$$

and

$$s_{ij}^{(k)} = N_i^{(k-1)} \cdot \eta\big(\hat{\theta}_i^{(k-1)}, \hat{\theta}_j^{(k-1)}\big)/\lambda.$$

For all i define estimates

$$\hat{\theta}_i^{(k)} = \arg\max_\theta l(Y, W_i^{(k)}; \theta)$$

calculate

$$N_i^{(k)} = \sum_{j \in \mathcal{X}_G} w_{ij}^{(k)}$$

end
Result: Adaptively denoised parameters $\hat{\theta}_i^{(k^\star)}$.

Algorithm 1: Formal outline of adaptive weights smoothing (AWS).

in design space and

$$\eta(\theta_i, \theta_j) = \mathcal{KL}(P_{\theta_i}, P_{\theta_j}) = \int_{\mathcal{Y}} p(y, \theta_i) \log \frac{p(y, \theta_i)}{p(y, \theta_j)} dy$$

in parameter space which is the Kullback Leibler divergence between the probability distributions with parameters θ_i and θ_j at locations x_i and x_j, respectively. Henceforth, we abbreviate the locations with indices i and j.

In general, adaptive weights smoothing can be defined for regular as well as irregular designs $\mathcal{X}$. However, the patch-wise smoothing algorithm that we propose in this paper requires $\mathcal{X}$ to be a one, two or three dimensional grid. Common examples include 2D or 3D images, where $\mathcal{X} \subset \mathbb{R}^d$ $(d = 2, 3)$ is a cube and image intensities Y_i are sampled at rectangular grid points x_i. Let $\mathcal{X}_G$ denote the set of grid points in $\mathcal{X}$.

Adaptive weights smoothing employs an iterative scheme with a sequence of increasing bandwidths $h^{(k)}$ for steps $k = 0, \ldots, k^\star$ alternating the computation of weighted maximum likelihood estimates

$$\hat{\theta}_i^{(k)} = \arg\max_\theta l(Y, W_i^{(k)}; \theta) = \arg\max_\theta \sum_{j \in \mathcal{X}_G} w_{ij}^{(k)} \log(p(Y_j, \theta))$$

and the determination of adaptive weighting schemes $W_i^{(k)} = \{w_{ij}^{(k)}, j \in \mathcal{X}_G\}$. Specifically, the weights $w_{ij}^{(k)}$ at iteration step k are given as the product of two terms: a kernel weight $K_{\mathrm{loc}}\big(l_{ij}^{(k)}\big)$ with $l_{ij}^{(k)} = \big(\delta(x_i, x_j)/h^{(k)}\big)^2$ and a component $K_{\mathrm{st}}\big(s_{ij}^{(k)}\big)$ depending on

$$s_{ij}^{(k)} = N_i^{(k-1)} \cdot \eta\big(\hat{\theta}_i^{(k-1)}, \hat{\theta}_j^{(k-1)}\big)/\lambda.$$

denoted as statistical penalty for two kernel functions K_{loc} and K_{st}. The term $N_i^{(k-1)} = \sum_j w_{ij}^{(k-1)}$ serves as a proxy for the variance reduction achieved for $\hat{\theta}_i^{(k-1)}$. Note, that the noise variance typically enters the function η and needs to be known or has to be estimated. The adaptive weights smoothing (AWS) is summarized by Algorithm 1.

The sequence of bandwidths $h^{(k)}$ is chosen such that for $\lambda = \infty$ the variance of the estimate $\hat{\theta}_i^{(k)}$ is reduced by a factor of c_h compared to $\hat{\theta}_i^{(k-1)}$. The specific value for c_h is not very important, cf. Li et al. 2012. However, $c_h = 1.25$ turned out to be a good compromise between sufficient increase of variance reduction between steps and careful increase of $h^{(k)}$ in order to obtain sufficiently neat coverage of the scale space. The kernels $K_{\rm loc}$ and $K_{\rm st}$ are monotone non increasing functions on $\mathbb{R}^+ \mapsto \mathbb{R}^+$ preferably with compact support. Our default choice is

$$K_{\rm loc}(x) = \max(0, 1 - x^2), \quad \text{and} \quad K_{\rm st}(x) = \max(0, \min(1, 4/3(1 - x))).$$

The main parameters of the procedure are the number k^* of iterations and the scale parameter λ of the statistical penalty. λ can be chosen independently from the data at hand by checking a so-called propagation condition for simulated data, see Becker and Mathé 2013; Becker 2014. For $\lambda = 0$ the data is not changed during the iteration, the choice $\lambda = \infty$ corresponds to a non-adaptive kernel estimate with kernel function K_{loc} and bandwidth $h^{(k^*)}$. An optimal λ will lie between these two extremes and provides a nearly nonadaptive kernel estimate in case of a globally constant parameter θ.

If the structural assumption is valid AWS possesses interesting properties (Polzehl and Spokoiny 2006): Within the interior of any homogeneous region $\mathcal{X}_n$ the final estimate $\hat{\theta}_i^{(k^\star)}$ is similar to a non-adaptive kernel smoother with a bandwidth $h^{(k^*)}$ as specified by the propagation condition (Becker and Mathé 2013; Becker 2014). On the other hand, two different regions $\mathcal{X}_k$ and $\mathcal{X}_l$ of the partition are separated, i.e., $w_{ij}^{(k^\star)} \simeq 0$ if $x_i \in \mathcal{X}_n$, $x_j \in \mathcal{X}_l$, and the contrast $\eta(\theta_i, \theta_j)$ exceeds some critical value that depends on the size of the two regions. For details, see Polzehl and Spokoiny 2006.

2.2. **Local polynomial adaptive weights smoothing (lpAWS).** The final estimate $\hat{\theta}_i^{(k^\star)}$ stabilizes for $k^* \to \infty$. If the structural assumption of a local constant parameter function $\theta(x)$ is not valid the algorithm nonetheless enforces a local constant parameter map, which leads to a cartoon-like appearance for the final estimate. In order to overcome this drawback and to relax the structural assumption on the data the Propagation-Separation approach has been generalized for locally smooth functions $\theta(x)$ (Polzehl and Spokoiny 2008; Polzehl and Tabelow 2012). Due to the increasing complexity of the algorithm we restrict ourselves to the case $d = 2$ of a two-dimensional design space, i.e., imaging data.

Specifically, we extend the structural assumption such that within a homogeneous region $x_i, x_j \in \mathcal{X}_n$, the data Y_j can be modeled with Gaussian additive errors ε_j as

$$Y_j = \theta_i^\top \cdot \Psi(x_{j_1} - x_{i_1}, x_{j_2} - x_{i_2}) + \varepsilon_j,$$

where the components of $\Psi(\xi_1, \xi_2)$ contain values of basis functions

$$\psi_{m_1,m_2}(\xi_1, \xi_2) = \xi_1^{m_1} \cdot \xi_2^{m_2}$$

for integers $m_1, m_2 \geq 0$, $m_1 + m_2 \leq p$ and some polynomial order p.

For a given local model $W_i^{(k)}$ at iteration step k estimates $\hat{\theta}_i^{(k)}$ of θ_i are obtained by local Least Squares as

$$\hat{\theta}_i^{(k)} = \left(B_i^{(k)}\right)^{-1} \sum_{j \in \mathcal{X}_G} w_{ij}^{(k)} \cdot \Psi(x_{j_1} - x_{i_1}, x_{j_2} - x_{i_2}) \cdot Y_j,$$

with

$$B_i^{(k)} = \sum_{j \in \mathcal{X}_G} w_{ij}^{(k)} \cdot \Psi(x_{j_1} - x_{i_1}, x_{j_2} - x_{i_2}) \cdot \Psi(x_{j_1} - x_{i_1}, x_{j_2} - x_{i_2})^\top.$$

At each position x_i (and x_j) the estimates $\hat{\theta}_i$ (and $\hat{\theta}_j$) are given in terms of a local set of basis functions ψ. In order to make $\hat{\theta}_j$ and $\hat{\theta}_i$ comparable we perform a simple linear (coordinate) transformation for

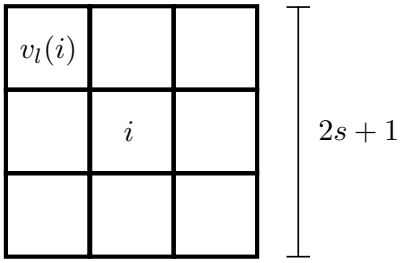

FIGURE 1. Schematic example of a patch V_i of size s of a location i. It contains all locations $v_l(i)$ with a maximum l_1-distance of s from i.

$\hat\theta_j$ using the local model and denote the result by $\hat\theta_{ji}$. At iteration step k the statistical penalty in the Gaussian model above can then be defined as

$$s_{ij}^{(k)} = \frac{1}{\lambda 2\sigma_i^2}\big(\hat\theta_i^{(k-1)} - \hat\theta_{ji}^{(k-1)}\big)^\top B_i^{(k-1)}\big(\hat\theta_i^{(k-1)} - \hat\theta_{ji}^{(k-1)}\big),$$

where σ_i^2 is the (local) variance of ε_i. We refer to Polzehl and Spokoiny 2008 or Polzehl and Tabelow 2012 for a more detailed description and examples using image data.

In general, local polynomial AWS improves the reconstruction results compared to the original AWS procedure in case of locally smooth images. As by lpAWS the cartoon-like appearance of the result is avoided, it enables larger variance reduction, i.e., the use of a larger k^*. However, it also increases model flexibility and therefore requires more extended homogeneous regions $\mathcal{X}_n$ to adapt to small contrasts in θ, i.e., to separate two regions $\mathcal{X}_n$ and $\mathcal{X}_l$ in terms of the adaptive weights $w_{ij}^{(k^\star)} \simeq 0$.

2.3. **Non-local means (NLM).** The non-local means filter has been introduced in Buades, Coll, and Morel 2005 and is related to the adaptive weights smoothing described above. It requires a regular grid $\mathcal{X}_G$ as design, we will assume an isotropic grid. Instead of using only the data at locations i and j to define adaptive weights w_{ij} it uses vectorized values in vicinities of the locations for comparison. In order to formalize the basic idea we introduce a local patch, or vicinity, $V_i = \{v_l(i)|\ ||v_l(i)-i||_1 \le s\}$ of a design point i, see Fig. 1. It contains all $n_s = (2s+1)^d$ grid points $x_{v_l(i)}$ within a d-dimensional cube of side length $2s$. The index $l = 1,\dots,n_s$ varies over the locations $v_l(i)$ in the patch V_i. We denote by Y_{V_i} the vectorized data $(Y_{v_1(i)},\dots,Y_{v_{n_s}(i)})$.

The method then defines local adaptive weights

$$w_{ij} = K_{\rm st}\left(||Y_{V_j} - Y_{V_i}||_2/h\right)$$

where $K_{\rm st}$ is typically chosen as a Gaussian kernel and h is some bandwidth. Thus, NLM is a single step AWS method ($k^\star = 1$) comparing Y_{V_j} and Y_{V_i} instead of Y_j and Y_i with an adaptation bandwidth $\lambda = h$, a uniform location kernel $K_{\rm loc}$, and $h^{(1)} = \infty$, giving rise to the notion "non-local". Estimation is performed by a weighted mean.

Several extensions and refinements of the basic method with a large number of applications in medical imaging context have been proposed and utilized. We refer the reader to the extensive literature on the topic. For the comparisons in this paper we will rely on the efficient Optimized Blockwise NonLocal Means Denoising Filter (ONLM) (Coupé et al. 2008) and the adaptive multi resolution non-local means filter (MRONLM) (Coupé et al. 2012). For both reference implementations by the authors are available.

Data: Observations $Y_i \in \mathcal{Y}$ at locations $x_i \in \mathcal{X}_G$.
Initialization: Set $k = 0$, $h^{(0)} = 1$, $w_{ij}^{(0)} = K_{\mathrm{loc}}\big(\delta(x_i, x_j)\big)$ and initialize $\hat\theta_i^{(0)}$ as weighted likelihood or least squares estimate;
while $k \le k^\star$ **do**
For all locations i and j define

$$w_{ij}^{(k)} = K_{\mathrm{loc}}\big(l_{ij}^{(k)}\big) K_{\mathrm{st}}\big(s_{ij}^{(k)}\big)$$

with

$$l_{ij}^{(k)} = \big(\delta(x_i, x_j)/h^{(k)}\big)^2$$

and

$$s_{ij}^{(k)} = \max_{l=1,\dots,n_s} N_{v_l(i)}^{(k-1)} \cdot \eta\big(\hat\theta_{v_l(i)}^{(k-1)}, \hat\theta_{v_l(j)}^{(k-1)}\big)/\lambda.$$

For all i define estimates

$$\hat\theta_i^{(k)} = \arg\max_\theta l(Y, W_i^{(k)}; \theta)$$

calculate

$$N_i^{(k)} = \sum_{j \in \mathcal{X}_G} w_{ij}^{(k)}$$

end
Result: Adaptively denoised parameters $\hat\theta_i^{(k^\star)}$.

Algorithm 2: Formal outline of patch-wise adaptive weights smoothing (PAWS).

3. Patch-wise adaptive weights smoothing (PAWS)

The major drawback of the original and the local polynomial AWS is that their final estimates for large $k^\star$ reflect the strict structural assumption, but do not incorporate any information on the smoothness of boundaries between homogeneous regions. Due to the construction of the method, large noise realizations in some location of the data can be mistaken for structure and lead to a speckled appearance of the final estimate. Here, we thus develop a new extension of AWS by combining its multi-scale approach with the use of information about the local spatial structure in terms of the local patches V_i as defined for NLM.

This patch-wise adaptive weights smoothing (PAWS) procedure will employ a new form of the statistical penalty $s_{ij}^{(k)}$ based on patches V_i. The variability of the estimates $\hat\theta_i$ at iteration step k depends on the (local) weighting schemes $W_i^{(k)}$. This is taken into account in the definition of the statistical penalty $s_{ij}^{(k)}$ by the use of the sum of weights $N_i^{(k)}$. Depending on the unknown underlying structure $\mathcal{X}_n$ the variability of the estimates $\hat\theta_{v_l(i)}$ may vary considerably over grid points $v_l(i) \in V_i$. Thus, when we extend the definition $s_{ij}^{(k)}$ to comparisons between patches, it should consider accuracy of the parameter estimates reflected by $N_{v_l(i)}$ as achieved in former iteration steps. We thus define a suitable statistical penalty for PAWS by

$$s_{ij}^{(k)} = \max_{l=1,\dots,n_s} N_{v_l(i)}^{(k-1)} \cdot \eta\big(\hat\theta_{v_l(i)}^{(k-1)}, \hat\theta_{v_l(j)}^{(k-1)}\big)/\lambda.$$

Taking the maximum over all locations $l = 1, \dots, n_s$ in the patch enables to balances spatial differences in the variance of the estimates.

As for AWS the adaptation bandwidth λ in $s_{ij}^{(k)}$ depends only on the parametric family $P_{\theta_i} \in \mathcal{P}_\Theta$, the dimension of the design space d and, additionally, on the patch size s. We choose it by a propagation condition, see Becker 2014. The algorithm is summarized in Algorithm 2.

We illustrate the propagation of the weights $W_i^{(k)}$ with k for several ground truth situations of the data. The example image is composed of four quadrants with a constant function, two linear functions with

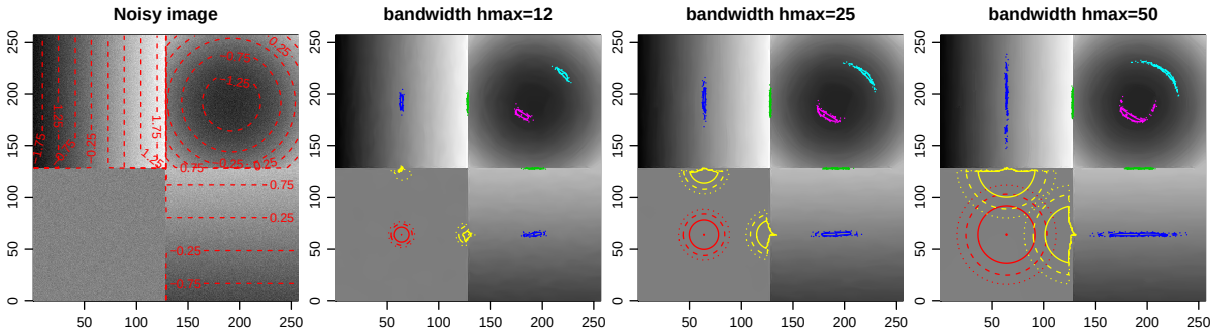

FIGURE 2. Propagation of weights for PAWS. Left: Noisy image with contour lines of the original structure added. 2nd, 3rd and 4th image (from left): reconstruction results for $k^* = 26, 32$, and 38 ($h_{max} = 12, 25$, and 50) and patch size $s = 2$ overlayed with contour lines, levels $.1$ (dotted), $.4$ (dashed) and $.7$ (solid), of weighting schemes in nine selected locations. Color labeling corresponds to varying scenarios, see text.

different gradient orientation and strength and a quadratic function, plus additive Gaussian noise. In the left of Figure 2 we plotted the noisy image and contour lines of the noise-free image for guidance. The other three plots provide the reconstruction results $\theta^{(k)}$ after $k = 26, 32$ and 38 iteration steps of the PAWS algorithm with a patch size $s = 2$. For these iteration steps and for nine selected locations i we overlay contour lines of the weights $w_{ij}^{(k)}$ corresponding to weights of $.1$ (dotted), $.4$ (dashed) and $.7$ (solid). We use different colors to indicate different scenarios: red for a location within a local constant region, blue in a region with a constant gradient, yellow for locations without intensity contrast at the border of two quadrants, green for locations at discontinuities and cyan/magenta for two locations with quadratic intensity profile but differing distance from the intensity minimum.

It can be seen that within a local constant intensity region the weights propagate isotropically in all directions as for a non-adaptive kernel smoother (red). For locations within a region with constant gradient the weighting schemes are more concentrated in gradient direction while freely extending in the orthogonal direction (blue). The behavior at discontinuities depends on the image contrast and the distance to the discontinuity. We observe either propagation within the homogeneous region (yellow) or only along the discontinuity if the distance is less than s (green). In case of a nonlinear intensity map the weighting schemes extend and concentrate along level sets with propagation restricted by their curvature.

Figure 3 provides the corresponding illustration for the original AWS procedure. Within the local smooth regions we observe, with increasing bandwidth $h^{(k)}$, the emergence a local constant image reconstruction and the propagation of weights within its constant segments.

4. ADAPTIVE SMOOTHING BASED ON REGULARIZATION

For data Y on regular Cartesian grids in two dimension, i.e., classical images, approaches based on the numerical solution of optimization problems have been established. They often employ an ℓ_1-regularization term that enforces sparsity of structures where the qualitative local assumptions are violated. The penalties account for deviations from a constant image intensity or constant gradient image. The optimization problem considered below codes similar structural assumptions as the AWS procedures considered in the last section.

4.1. **Total variation regularization (TV).** Total variation is a classical energy minimizing method (Rudin, Osher, and Fatemi 1992), where the denoised image U is obtained as a minimizer of the

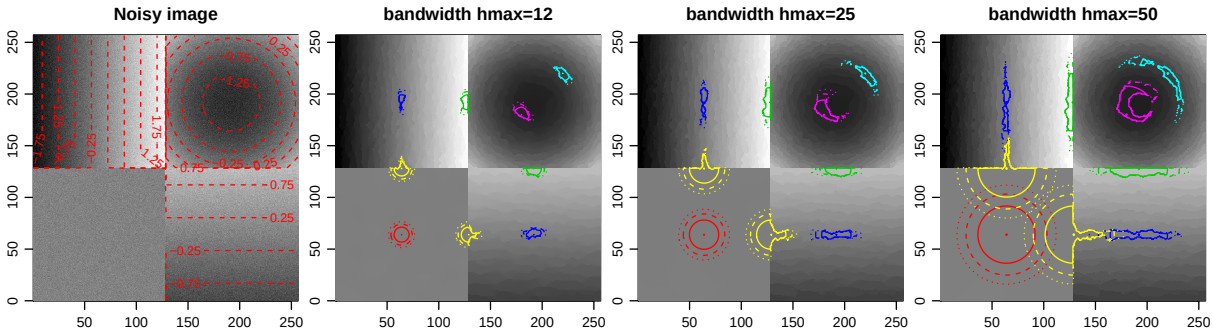

FIGURE 3. Propagation of weights for AWS. Left: Noisy image with contour lines of the original structure added. 2nd, 3rd and 4th image (from left): reconstruction results for $k^* = 26, 32$, and 38 ($h_{max} = 12, 25$, and 50) overlayed with contour lines, levels $.1$ (dotted), $.4$ (dashed) and $.7$ (solid), of weighting schemes in nine selected locations. Color labeling corresponds to varying scenarios, see text.

energy

$$\min_U \frac{1}{2}\|U - Y\|_2^2 + \mathrm{TV}_\alpha(U). \tag{1}$$

Here $\|U - Y\|_2^2$ denotes the square of the Euclidean distance between U and Y (discrepancy term), which corresponds to the fact that the data Y is assumed to be corrupted by Gaussian noise. For an image U defined on a Cartesian grid $d_1 \times d_2$, the term $\mathrm{TV}_\alpha(U)$ is the total variation of U which in this discrete formulation reads

$$\mathrm{TV}_\alpha(U) = \alpha\|\nabla U\|_1 = \alpha \sum_{i_1=1}^{d_1}\sum_{i_2=1}^{d_2}((U_{i_1+1,i_2} - U_{i_1,i_2})^2 + (U_{1,i_2+1} - U_{i_1,i_2})^2)^{1/2}.$$

The scalar constant $\alpha > 0$ balances the two terms in (1) and determines the amount of filtering. Minimizing the ℓ_1 norm of the gradient results in sparsity of the gradient of the denoised image. As a result, total variation, like adaptive weights smoothing with a local constant structural assumption, promotes piecewise constant reconstructions which on one hand leads to edge preservation and sharp images. On the other hand it also leads to blocky artifacts (staircasing effect) which is often undesirable in natural images. TV reconstructions are typically accompanied by a certain loss of contrast. There is a vast literature concerning TV minimisation. Here we refer the reader to Caselles, Chambolle, and Novaga 2007 and Ring 2000 for analytical properties, and to Hintermüller et al. 2017 for parameter selection as well as to the references therein.

4.2. **Total generalized variation regularization (TGV).** Total generalized variation (Bredies, Kunisch, and Pock 2010) has been proposed as a higher order extension of TV, aiming to avoid the staircasing effect and on the same time to retain the ability of TV to preserve edges. In the case of Gaussian noise, one minimizes the functional

$$\min_U \frac{1}{2}\|U - Y\|_2^2 + \mathrm{TGV}_{\alpha,\beta}(U),$$

where

$$\mathrm{TGV}_{\alpha,\beta}(U) = \min_W \ \alpha\|\nabla U - W\|_1 + \beta\|\mathcal{E}W\|_1.$$

Here, $\mathcal{E}W$ is the symmetrized gradient of W, i.e., $\mathcal{E}W = \frac{1}{2}(\nabla W + \nabla W^\perp)$, and $\alpha, \beta > 0$. If $W = 0$ then $\mathrm{TGV}_{\alpha,\beta}(U) = \mathrm{TV}\alpha(U)$. If $W = \nabla U$ then $\mathrm{TGV}_{\alpha,\beta}(U) = \beta\|\nabla^2 U\|_1$. Thus, the TGV functional can be interpreted as an optimal balance between first and second order ℓ_1-type regularization. TGV minimization promotes piecewise affine reconstructions rather than piecewise constant, typically

resulting in more visually pleasing results than TV. One drawback of TGV minimization is the presence of two regularizations parameters that need to be balanced, see Papafitsoros and Bredies 2015; Reyes, Schönlieb, and Valkonen 2017, as well as the increased computational cost in comparison to TV.

5. SOFTWARE

The three adaptive smoothing algorithms AWS, lpAWS and PAWS were implemented in the R-package **aws**. The package is available under GPL (>=2) from `https://cran.r-project.org/package=aws`. The version used in this paper is 2.0-4. The three methods are implemented within the corresponding functions `aws`, `lpaws`, and `paws`:

```
R> aws(nimg, hmax, sigma2 = sig^2)
R> lpaws(nimg, degree = 1, hmax, sigma2 = sig^2)
R> paws(nimg, hmax, patchsize = 1, sigma2 = sig^2)
```

For all three functions `nimg` is the array containing the noisy data. While `aws` and `paws` allow for one-, two-, and three-dimensional data, `lpaws` can be applied to two-dimensional data, only. `hmax` is the value for the maximum bandwidth, the function will automatically create a suitable sequence $h^{(k)}$ of bandwidths with $c_h = 1.25$ such that $h^{(k^\star)}$ does not exceed `hmax`. The `degree` argument of `lpaws` denote the degree of the local polynomials. `sigma2` is a global estimate for the variance of the noise. If omitted, a robust estimate based on the inter-quartile range of first order intensity differences in coordinate directions is generated. In our simulation study we use the known variance for simplicity. The adaptation bandwidth λ is predefined by a propagation condition for the utilized noise model and dimensionality of the data. Adjustments for more or less adaptation in the reconstruction can be achieved by the parameter `ladjust` for all three functions. For more information see the man-pages of the package and the scripts accompanying this paper which have been used to do the simulation study below.

The regularization methods TV and TGV were also implemented in the R-package **aws**:

```
R> TV_denoising(nimg, alpha, iter)
R> TGV_denoising(nimg, alpha, beta, iter)
```

where `alpha` (and `beta`) are the regularization parameters for TV and TGV as described above. Iterations of the optimization algorithm are terminated if either the ℓ_1-norm or ℓ_∞-norm of the difference of two consecutive image reconstructions are less than their specified tolerance values, or the maximum number `iter` of iterations is reached.

For the ONLM and MRONLM algorithms we used the original Matlab implementations by Pierrick Coupé which are available at `https://sites.google.com/site/pierrickcoupe/softwares/denoising-for-medical-imaging/mri-denoising`.

The example images and the scripts to reproduce the results and figures in this paper are available in the online supplement.

6. EXAMPLES

We illustrate and discuss the performance of our patch-wise adaptive weights smoothing (PAWS) algorithm in comparison to the other adaptive smoothing procedures (AWS, lpAWS, ONLM, MRONLM, TV, TGV) in three examples with artificial Gaussian noise. For the evaluation of the reconstruction $\hat{U}$ in comparison to the ground truth image U we use the following common criteria

FIGURE 4. Example 2D images: grayscale image and color (sRGB) image

- Peak Signal to Noise Ratio (PSNR):

$$\mathrm{PSNR}(\hat{U}, U) = 20 \log_{10}(\max(U) - \min(U)) - 10 \log_{10}(\mathrm{var}(\hat{U} - U))$$

- Mean Absolute Error (MAE):

$$\mathrm{MAE}(\hat{U}, U) = \mathrm{mean}(|\hat{U} - U|)$$

- Structural Similarity (SSIM), see Wang et al. 2004:

$$\mathrm{SSIM}(\hat{U}, U) = \frac{(2 * \mathrm{mean}(\hat{U})\,\mathrm{mean}(U) + c_1)(2 * \mathrm{cov}(\hat{U}, U) + c_2)}{(\mathrm{mean}(\hat{U})^2\,\mathrm{mean}(U)^2 + c_1)(\mathrm{var}(\hat{U})^2\,\mathrm{var}(I)^2 + c_2)}$$

  with $c_1 = 10^{-4}(\max(U) - \min(U))^2$ and $c_2 = 9c_1$.
- Mean Absolute Gradient Error (MAGE):

$$\mathrm{MAGE}(\hat{U}, U) = \mathrm{mean}(|\nabla\hat{U} - \nabla U|)$$

  using a numeric approximation of the gradient.
- Root Mean Squared Gradient Error (RMSGE):

$$\mathrm{RMSGE}(\hat{U}, U) = \sqrt{\mathrm{var}(\nabla\hat{U} - \nabla U)}.$$

6.1. **Example 1 — Grayscale parrot image.** The first example uses a greyscale image (resolution 256×256) extracted from kodim23.png (`http://r0k.us/graphics/kodak/`), see the left image in Fig. 4. Spatially independent Gaussian noise with three different standard deviations, $\sigma = .04, \sigma = .08$ and $\sigma = .16$ was added after standardizing the image to a range $[0, 1]$.

For each noise level we computed reconstructions using AWS, with k^* optimized with respect to PSNR; lpAWS, with polynomial degree $p = 1$; TV and TGV, both with parameters α (and β) optimized with respect to PSNR, and our new PAWS method with patch sizes $s = 1, 2$ and 3. For PAWS and lpAWS the number of iterations k^* used was 18, 22 and 24, corresponding to a maximum bandwidth $h^{(k^*)}$ of 4.9, 7.6 and 9.5, for the three noise levels. Default values for λ was used. These values were determined by simulation to obey a propagation condition (Becker and Mathé 2013) and are hard-coded in the functions `aws`, `lpaws` and `paws`.

Numerical results with respect to PSNR, MAE and SSIM are provided in Tables 1- 3. They suggest a superior behavior of PAWS especially in case of low SNR. Figure 5 provides the reconstructions achieved by AWS, PAWS with $s = 2$, lpAWS with a degree $p = 1$, TV and TGV together with the noisy source image. The reconstructed images are projected into the range of the original for display in order to improve comparability.

	original	TV	TGV	AWS	PAWS_1	PAWS_2	PAWS_3	lpAWS
$\sigma = 0.04$	27.4	33.4	33.7	32.2	33.7	33.8	33.7	33.1
$\sigma = 0.08$	21.3	29.8	30.1	28.1	30.5	30.7	30.6	29.7
$\sigma = 0.16$	15.3	26.4	26.7	25.8	27.4	27.8	27.8	25.9

TABLE 1. PSNR of parrot reconstructions for the different noise levels. The index for PAWS corresponds to the patch size ($s = 1, 2, 3$).

	original	TV	TGV	AWS	PAWS_1	PAWS_2	PAWS_3	lpAWS
$\sigma = 0.04$	0.0320	0.0141	0.0135	0.0166	0.0132	0.0130	0.0132	0.0138
$\sigma = 0.08$	0.0641	0.0201	0.0191	0.0239	0.0181	0.0177	0.0179	0.0191
$\sigma = 0.16$	0.1281	0.0287	0.0272	0.0295	0.0252	0.0241	0.0242	0.0279

TABLE 2. As Table 1 but reporting MAE.

	original	TV	TGV	AWS	PAWS_1	PAWS_2	PAWS_3	lpAWS
$\sigma = 0.04$	0.9843	0.9960	0.9962	0.9948	0.9963	0.9964	0.9963	0.9957
$\sigma = 0.08$	0.9400	0.9908	0.9913	0.9866	0.9922	0.9926	0.9923	0.9907
$\sigma = 0.16$	0.7966	0.9795	0.9807	0.9768	0.9840	0.9855	0.9852	0.9772

TABLE 3. As Table 1 but reporting SSIM.

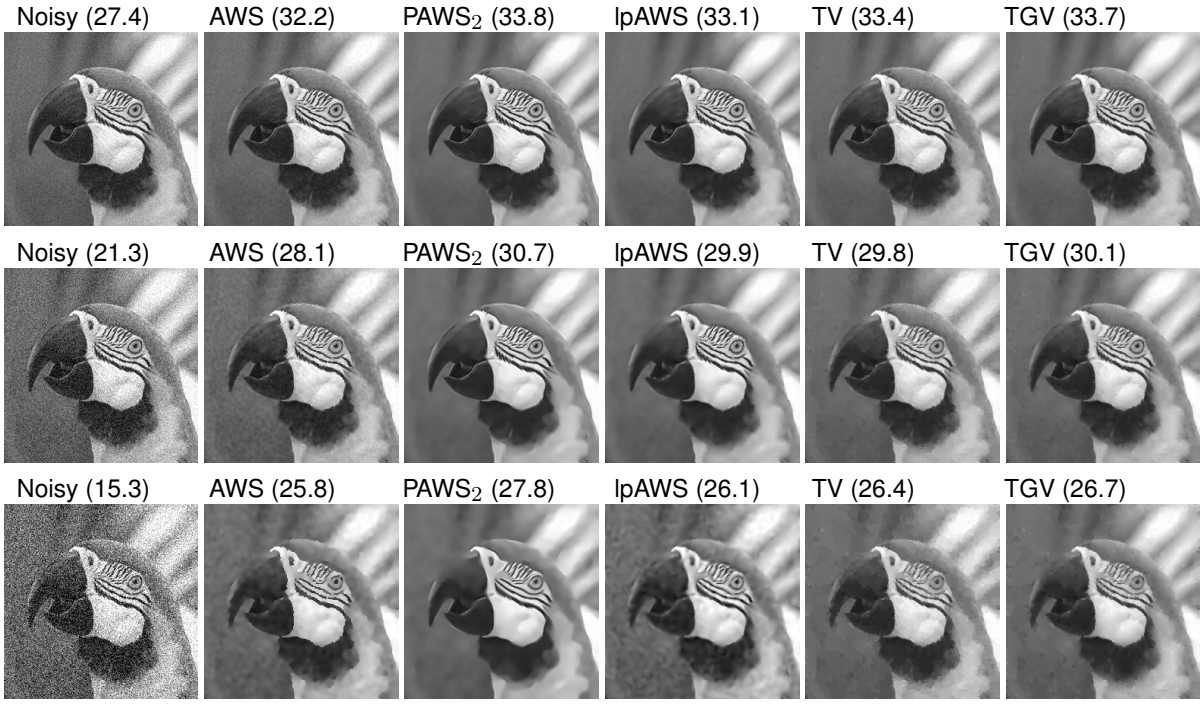

FIGURE 5. Resulting optimal reconstruction for parrot image. The numbers is parentheses are the PSNR values.

6.2. **Example 2 — Color image in 2D.** The AWS and PAWS algorithms can be easily extended to handle color images using $\eta(\theta_i, \theta_j) = (\theta_i - \theta_j)^\top \Sigma_i^{-1} (\theta_i - \theta_j)$ where $\theta \in \mathbb{R}^3$ is a vector of intensities in the RGB channels and the corresponding Σ_i the error covariance matrix, see Polzehl and Tabelow 2007. The extension of TV and TGV to color images is straightforward as well.

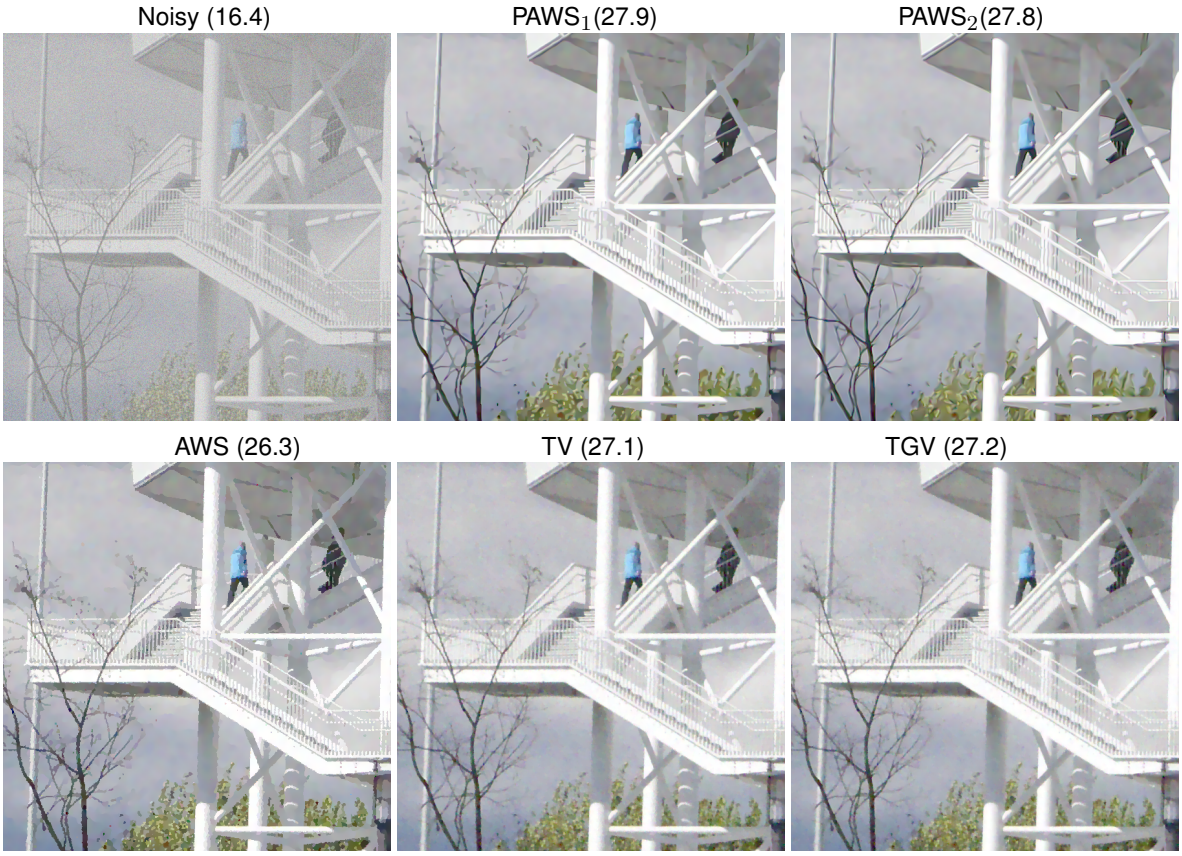

FIGURE 6. Reconstruction results for the color image. For more details a zoomed image is show. PSNR values are in parentheses, see also Table 4.

For an color image example we used the right image in Fig. 4. We added Gaussian noise with standard deviation equal to 15% of the intensity range in each of the RGB channels of the image. The reconstructions using TV and TGV were obtained using optimized parameters α (and β) with respect to PSNR. For AWS and PAWS we used $k^* = 24$ (corresponding to $h^{(k^*)} = 9.5$) and adjusted the parameter λ to maximize PSNR. Figure 6 shows a zoom of the noisy image and its reconstructions. The intensity values have again been projected into the range $[0, 1]$ in each RGB channel for better comparability. PSNR and MAE results reported in the headings of each image correspond to the full image size of 1700×1400 pixel. Note, that the assumption of a local constant model is enforced in the AWS reconstruction in regions with smoothly changing intensities. This effect is by far less prominent with both the PAWS and TV/TGV reconstructions.

	noisy	AWS	PAWS$_1$	PAWS$_2$	TV	TGV
PSNR	16.4	26.3	27.9	27.8	27.1	27.2
MAE	7840	1920	1660	1680	1900	1880

TABLE 4. PSNR and MAE of the reconstruction results for the color image example.

6.3. **Example 3 — 3D brain image.** Our third example uses a 3D T1-weighted image volume with a 1mm isotropic voxel resolution from BrainWeb `http://brainweb.bic.mni.mcgill.ca/cgi/brainweb1` , see Figure 7. The image dimension is $181 \times 217 \times 181$, image intensity ranges between 0 and 4095 (12 Bit).

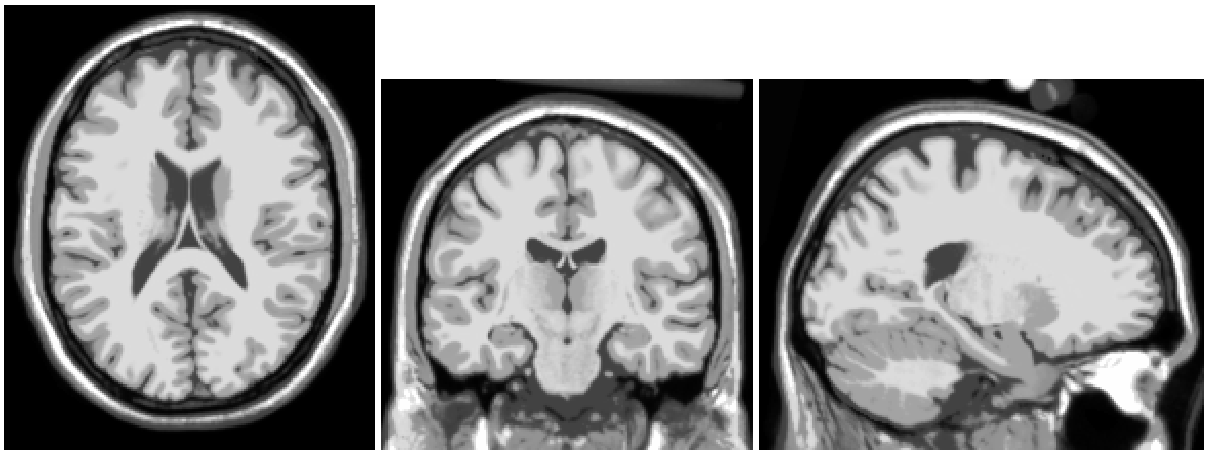

FIGURE 7. Example 3D brain images (T1w) from BrainWeb: axial (slice #91), coronal (slice #109) and sagittal (slice #70) view.

Images corrupted with noise were obtained by adding spatially independent Gaussian noise with a standard deviation 200 (high SNR), 400 (medium SNR) and 800 (low SNR) to each voxel intensity. For reconstruction we used AWS and PAWS with default values of λ chosen according to a propagation condition (Becker and Mathé 2013). The value of k^* for AWS was selected to provide best results in terms of PSNR (specifically 11, 13 and 20 for the three situations). For PAWS we used $k^* = 23 (h^{(last)} = 3.85)$ for high SNR, $k^* = 26 (h^{(last)} = 4.81)$ for medium SNR and $k^* = 28 (h^{(last)} = 5.6)$ in case of low SNR. For comparison with the non-local means methods we used the implementations for ONLM and MRONLM as provided by the authors.

Figure 8 illustrates the quality of reconstruction for the various methods for the central axial slice. The values of PSNR and MAE are reported in Tables 5 and 6 and refer to all voxel within a 3D brain mask obtained by thresholding the original 3D BrainWeb image.

	noisy	AWS	PAWS_1	PAWS_2	ONLM	MRONLM
$\sigma = 200$	26.2	31.4	35.4	34.3	35.6	35.7
$\sigma = 400$	20.2	28.5	32.5	32.2	31.8	32.1
$\sigma = 800$	14.2	26.3	29.3	29.6	28.1	28.6

TABLE 5. PSNR of brain reconstructions for the different noise levels.

	noisy	AWS	PAWS_1	PAWS_2	ONLM	MRONLM
$\sigma = 200$	160	80.2	49.1	56.4	48.8	47.7
$\sigma = 400$	319	113	68.5	72	78.4	74.6
$\sigma = 800$	639	139	99.8	98.7	124	115

TABLE 6. MAE of brain reconstructions for the different noise levels.

In Figure 9 we illustrate the accuracy of edge estimation. We show the central axial slice of a 3D image that contains the norm of the standard numerical gradient approximation as voxel intensity.

7. DISCUSSION & CONCLUSION

In this paper, we presented a new noise reduction algorithm *patch-wise adaptive weights smoothing* based on the Propagation-Separation approach that combines the multiscale approach of PS

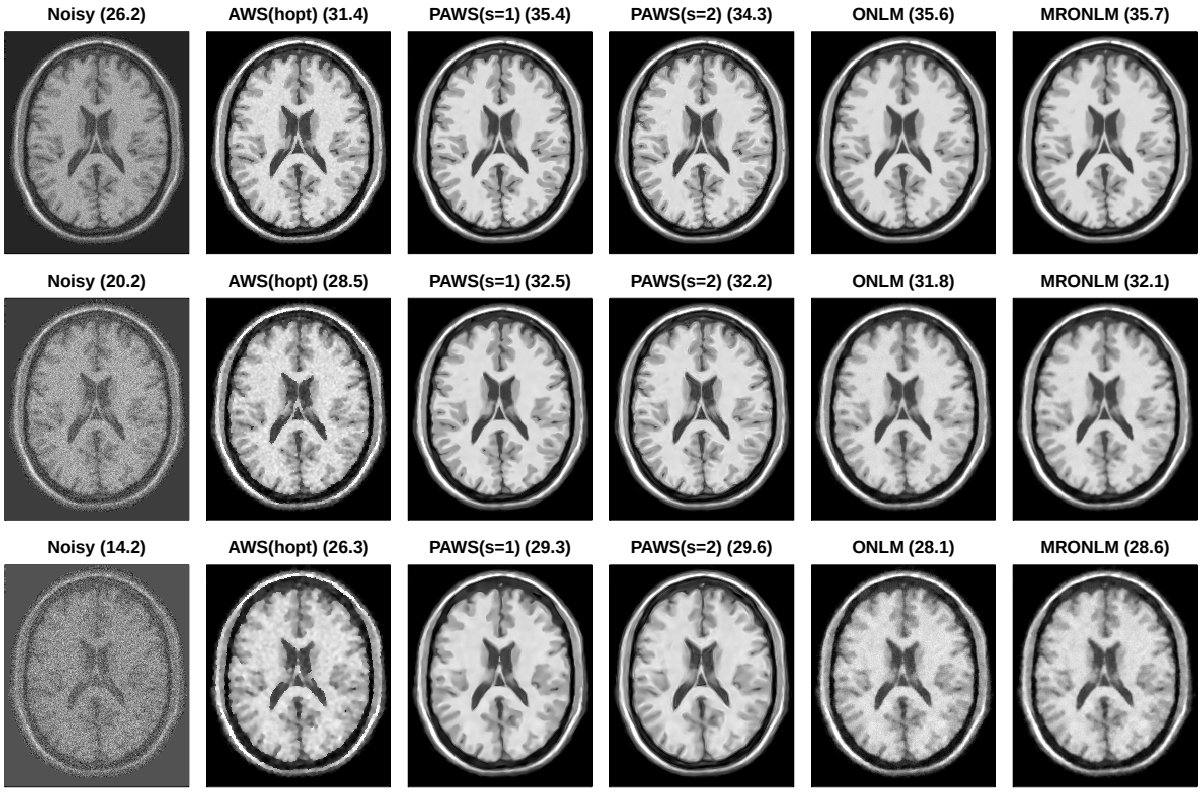

FIGURE 8. Example 3D brain images (T1w) from BrainWeb: axial view (slice 91) of noisy image, AWS, PAWS(s=1,2), ONLM and MRONLM reconstructions. Rows show results for differing noise levels. The numbers is parentheses refer to the PSNR of the 3D reconstruction within the brain mask. Numerical values MAE over all voxel within a brain mask are reported in Tables 5 and 6.

	noisy	AWS	PAWS_1	PAWS_2	ONLM	MRONLM
$\sigma = 200$	170	62.9	30.1	33.3	34	32.2
$\sigma = 400$	362	92	41.1	43	56	50.8
$\sigma = 800$	779	96.7	58.7	58.1	89.3	78.6

TABLE 7. MAGE of brain reconstructions for the different noise levels.

	noisy	AWS	PAWS_1	PAWS_2	ONLM	MRONLM
$\sigma = 200$	200	78.5	45.3	33.3	48.6	48.5
$\sigma = 400$	418	114	64.6	65.7	75.9	73.5
$\sigma = 800$	879	138	91.9	88.7	112	104

TABLE 8. RMSGE of brain reconstructions for the different noise levels.

with the definition of adaptive weighting schemes based on comparisons of patches of image intensities. It can be easily applied for data in any dimension d. We also described an implementation of PAWS as well as AWS and lpAWS within the R-package **aws** which is freely available under GPL from `https://cran.r-project.org/`. We demonstrated, that the combination of both ideas, the multiscale approach and the patch-wise comparison, leads to improved reconstruction results in comparison to methods based on a single ingredient. We also demonstrated how the method compares with regularization-based methods like TV and TGV, which in general show similar properties and

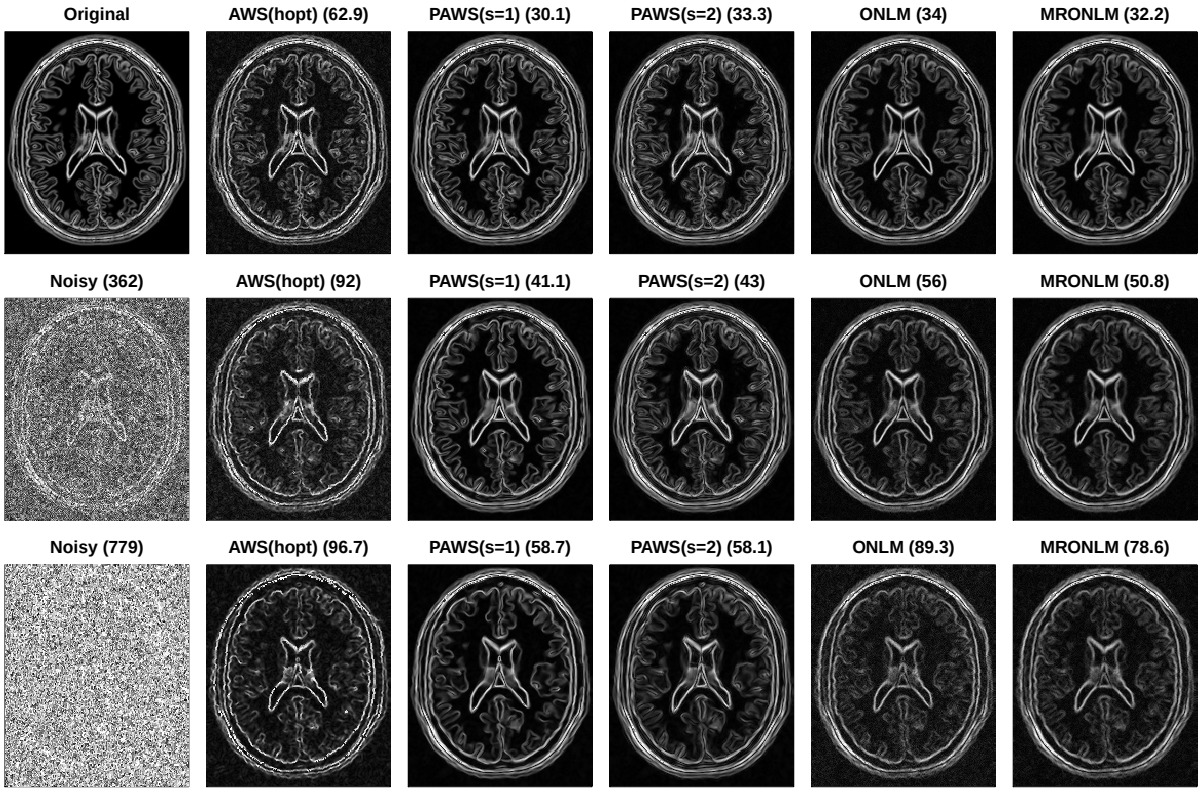

FIGURE 9. Example 3D brain images (T1w) from BrainWeb: axial view (slice 91) of 3D edge indicators for (from left to right) noisy image (2nd and 3rd row, original is used in the first row for comparison), AWS, PAWS(s=1,2), ONLM and MRONLM reconstructions. Rows show results for differing noise levels. Numerical values in parenthesis refer to MAGE. Numerical values for MAGE and RMSGE are reported in Tables 7 and 8.

performance to the PS methods. The R-package **aws** now also contains implementations for both methods, which where also described in this paper.

The new PAWS method overcomes the problem of singular locations with extreme image intensity in the reconstruction and the roughness of the boundaries of regions with homogeneous image intensity. The usage of the maximum statistics in the definition of the statistical penalty automatically takes the different variability of the estimates from the previous iteration step into account. PAWS shows also improved performance if the image data is characterized by local smooth instead of local constant intensity regions. It outperforms lpAWS in these cases and is, in contrast to lpAWS, easily applicable for d-dimensional data with $d > 2$.

REFERENCES

Becker, S. (2014). "The Propagation-Separation Approach - theoretical study and application to magnetic resonance imaging". Doctoral dissertation. Humboldt-University Berlin.

Becker, S. and P. Mathé (2013). "A different perspective on the Propagation-Separation Approach". *Electron. J. Statist.* 7, pp. 2702–2736.

Becker, S., K. Tabelow, H. U. Voss, A. Anwander, R. M. Heidemann, and J. Polzehl (2012). "Position-orientation adaptive smoothing of diffusion weighted magnetic resonance data (POAS)". *Med. Image Anal.* 16.6, pp. 1142–1155.

Becker, S., K. Tabelow, S. Mohammadi, N. Weiskopf, and J. Polzehl (2014). “Adaptive smoothing of multi-shell diffusion-weighted magnetic resonance data by msPOAS”. *NeuroImage* 95, pp. 90–105.

Bowman, A. and A. Azzalini (1997). *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

Bredies, K., K. Kunisch, and T. Pock (2010). “Total Generalized Variation”. *SIAM J. Imaging Sci.* 3.3, pp. 492–526.

Buades, A., B. Coll, and J. M. Morel (2005). “A review of image denoising algorithms, with a new one”. *Simulation* 4, pp. 490–530.

Caselles, V., A. Chambolle, and M. Novaga (2007). “The discontinuity set of solutions of the TV denoising problem and some extensions”. *Multiscale Model. Simul.* 6.3, pp. 879–894.

Chaudhuri, P. and J. S. Marron (2000). “Scale space view of curve estimation”. *Ann. Statist.* 28, pp. 408–428.

Coupé, P., P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot (2008). “An Optimized Blockwise NonLocal Means Denoising Filter for 3-D Magnetic Resonance Images”. *IEEE Trans. Med. Imaging* 27.4, pp. 425–441.

Coupé, P., J. V Manjon, M. Robles, and D. L. Collins (2012). “Adaptive multiresolution non-local means filter for three-dimensional magnetic resonance image denoising”. *IET Image Process.* 6(5).5, pp. 558–568.

Duits, R., H. Fuehr, and B. Janssen (2012). “Left Invariant Evolutions on Gabor Transforms”. In: *Mathematical Methods for Signal and Image Analysis and Representation*. Ed. by L. Florack, M.-C. van Lieshout, R. Duits, L. Davies, and G. Jongbloed. Computational Imaging and Vision. Springer. Chap. 8, pp. 137–158.

Fan, J. and I. Gijbels (1996). *Local Polynomial Modelling and its Applications*. Chapman & Hall, London.

Felsberg, M., P.-E. Forssén, and H. Scharr (2006). “Channel Smoothing: Efficient Robust Smoothing of Low-Level Signal Features”. *IEEE Trans. Pattern Anal. Mach. Intell.* 28, pp. 209–222.

Felsberg, M. (2012). “Adaptive Filtering using Channel Representations”. In: *Mathematical Methods for Signal and Image Analysis and Representation*. Ed. by L. Florack, M.-C. van Lieshout, R. Duits, L. Davies, and G. Jongbloed. Computational Imaging and Vision. Springer. Chap. 2, pp. 31–48.

Florack, L. (2012). “Scale Space Representations Locally Adapted to the Geometry of Base and Target Manifold”. In: *Mathematical Methods for Signal and Image Analysis and Representation*. Ed. by L. Florack, M.-C. van Lieshout, R. Duits, L. Davies, and G. Jongbloed. Computational Imaging and Vision. Springer. Chap. 9, pp. 159–172.

Franken, E. M. (2008). “Enhancement of Crossing Elongated Structure in Images”. Doctoral dissertation. Eindhoven University of Technology, Dept. of Biomedical Engineering.

Hintermüller, M., C. Rautenberg, T. Wu, and A. Langer (2017). “Optimal Selection of the Regularization Function in a Weighted Total Variation Model. Part II: Algorithm, Its Analysis and Numerical Tests”. *J. Math. Imaging Vis.* 59.3, pp. 515–533.

Lee, J. (1980). “Digital image enhancement and noise filtering by use of local statistics”. *IEEE Trans. Pattern Anal. Mach. Intell.* pp. 165–168.

Li, Y., J. H. Gilmore, J. Wang, M. Styner, W. Lin, and H. Zhu (2012). “TwinMARM: Two-Stage Multiscale Adaptive Regression Methods for Twin Neuroimaging Data”. *IEEE Trans. Med. Imaging* 31.5, pp. 1100–1112.

Manjón, J. V., P. Coupé, L. Martí-Bonmatí, D. L. Collins, and M. Robles (2009). “Adaptive non-local means denoising of MR images with spatially varying noise levels”. *J. Magn. Res. Imaging* 31.1, pp. 192–203.

Papafitsoros, K. and K. Bredies (2015). “A study of the one dimensional total generalised variation regularisation problem”. *Inverse Probl. Imag.* 9.2, pp. 511–550.

Perona, P. and J. Malik (1990). “Scale-space and edge detection using anisotropic diffusion”. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, pp. 629–639.

Polzehl, J. and V. Spokoiny (2000). “Adaptive Weights Smoothing with Applications to Image Restoration”. *J. R. Stat. Soc. Series B* 62, pp. 335–354.

– (2006). “Propagation-separation approach for local likelihood estimation”. *Probab. Theory and Relat. Fields* 135, pp. 335–362.

– (2008). “Structural adaptive smoothing by propagation-separation methods”. In: *Handbook of Data Visualization*. Ed. by C. Chen, W. Härdle, and A. Unwin. Series Handbooks of Computational Statistics. Springer-Verlag. Chap. Structural adaptive smoothing by propagation-separation methods, pp. 471–492.

Polzehl, J. and K. Tabelow (2007). “Adaptive smoothing of digital images: The R package adimpro”. *J. Statist. Software* 19, pp. 1–17.

– (2012). “Structural Adaptive Smoothing: Principles and Applications in Imaging”. In: *Mathematical Methods for Signal and Image Analysis and Representation*. Ed. by L. Florack, M.-C. van Lieshout, R. Duits, L. Davies, and G. Jongbloed. Computational Imaging and Vision. Springer. Chap. 4, pp. 65–81.

Polzehl, J., H. U. Voss, and K. Tabelow (2010). “Structural adaptive segmentation for statistical parametric mapping”. *NeuroImage* 52.2, pp. 515–523.

Reyes, J. C. De los, C.-B. Schönlieb, and T. Valkonen (2017). “Bilevel Parameter Learning for Higher-Order Total Variation Regularisation Models”. *J. Math. Imaging Vis.* 57.1, pp. 1–25.

Ring, W. (2000). “Structural properties of solutions to total variation regularization problems”. *ESAIM: Math. Model. Numer. Anal.* 34.4, pp. 799–810.

Rudin, L., S. Osher, and E. Fatemi (1992). “Nonlinear total variation based noise removal algorithms”. *Phys. D* 60.1-4, pp. 259–268.

Scharr, H. and K. Krajsek (2012). “A Short Introduction to Diffusion-like Methods”. In: *Mathematical Methods for Signal and Image Analysis and Representation*. Ed. by L. Florack, M.-C. van Lieshout, R. Duits, L. Davies, and G. Jongbloed. Computational Imaging and Vision. Springer. Chap. 1, pp. 1–30.

Simonoff, J. (1996). *Smoothing Methods in Statistics*. Springer, New York.

Tabelow, K., J. Polzehl, H. Voss, and V. Spokoiny (2006). “Analyzing fMRI experiments with structural adaptive smoothing procedures”. *NeuroImage* 33.1, pp. 55–62.

Tabelow, K., J. Polzehl, V. Spokoiny, and H. Voss (2008). “Diffusion Tensor Imaging: Structural adaptive smoothing”. *NeuroImage* 39, pp. 1763–1773.

Tomasi, C. and R. Manduchi (1998). “Bilateral Filtering for Gray and Color Images”. In: *Proceedings of the 1998 IEEE International Conference on Computer Vision*, pp. 839–846.

Wand, M. and M. Jones (1995). *Kernel Smoothing*. Chapman & Hall, London.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004). “Image quality assessment: from error visibility to structural similarity”. *IEEE Trans. Image Process.* 13.4, pp. 600–612.

Weickert, J. (1998). *Anisotropic Diffusion in Image Processing*. ECMI. Teubner-Verlag, Stuttgart.