Energy-Based Image Simplification with Nonlocal Data and Smoothness Terms

Stephan Didas[1], Pavel Mrázek[2], and Joachim Weickert[1]

[1] Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science
Saarland University, 66041 Saarbrücken, Germany
{didas,weickert}@mia.uni-saarland.de
[2] Upek R&D s.r.o., Husinecka 7, 130 00 Prague 3, Czech Republic
pavel.mrazek@upek.com

Summary. Image simplification and smoothing is a very important basic ingredient of a lot of practical applications. In this paper we compare different numerical approaches to solve this image approximation task within a unifying variational approach presented in [8]. For methods based on fixed point iterations we show the existence of fixed points. To speed up the convergence we also use two approaches involving Newton's method which is only applicable for convex penalisers. The running time in practice is studied with numerical examples in 1-D and 2-D.

1 Introduction

The task of image smoothing, simplification and denoising is the subject of various approaches and applications. An initial image is approximated by filtered versions which are smoother or simpler in some sense. Statistical estimation, median or mode filters, nonlinear diffusion, bilateral filtering or regularisation methods are among the tools helpful to reach this aim. Most of these tools somehow incorporate a neighbourhood of the pixel under consideration and perform some kind of averaging on the grey values. One of the earliest examples for such filters has been presented by Lee [7], followed by a lot of successors like the *SUSAN* filter by Smith and Brady [14]. In the context of statistical methods, Polzehl and Spokoiny presented a technique called *adaptive weights smoothing* [11]. The *W-estimator* by Winkler et al. [17] can be related to a spatially weighted *M-smoother* [5]. A very similar evolution is the *bilateral filter* by Tomasi and Manduchi [16], another prominent example for a weighted averaging filter. In its original form it is interestingly not meant to be iterative. There are approaches to relate it to variational principles [4]. In general there are a lot of approaches to give relations between averaging methods and techniques based on minimisation of energy functionals or on partial differential equations [13, 1].

In [8], an energy-based approach has been proposed which allows to consider a whole spectrum of well-known methods as different facets of the same model. This approach makes use of so-called *Nonlocal Data and Smoothness terms*; thus it will be called *NDS* here. These terms can consider not only information from a small region around a pixel but also make it possible to involve large neighbourhoods. The data term rewards similarity of our filtered image to the given one while the smoothness term penalises high deviations inside a neighbourhood of the evolving image.

The goal of the present paper is to analyse numerical methods for this approach. This paper is organised as follows: Section 2 gives a closer description of the energy functional we deal with and its relations to well-known filtering methods like M-smoothers and the bilateral filter. In Section 3 we discuss different approaches to minimise the NDS functional including a fixed point scheme and Newton's method. Numerical experiments in 1-D and 2-D in Section 4 compare the behaviour and running time of the presented approaches. A summary of the results and an outlook conclude the paper in Section 5.

2 The Filtering Framework

In this section we review the variational model presented in [8] and relate it to other filtering techniques. Let $f, u \in \mathbb{R}^n$ be discrete one- or two-dimensional images. We always denote the initial noisy image of the filtering process with f and the processed one with u. Let $\Omega = \{1, \dots, n\}$ be the index set of all pixels in the images. The pixel positions on the one- or two-dimensional grid will be denotes with $x_i (i \in \Omega)$. That means $|x_i - x_j|^2$ yields the square of the Euclidean distance between the two pixels x_i and x_j in the real line (1-D) or the plane (2-D). This will be referred to as *spatial distance*. The *tonal distance* then is the distance between grey values of two pixels, for example $|u_i - f_j|^2$.

We start with an energy functional involving the tonal distance between u and f:

$$E_D(u) = \sum_{i \in \Omega} \sum_{j \in \Omega} \Psi_D \left(|u_i - f_j|^2 \right) w_D \left(|x_i - x_j|^2 \right) \tag{1}$$

The iterative minimisation of such a scheme leads to the well-known W-estimator

$$u_i^0 := f_i, \quad u_i^{k+1} := \frac{\sum_{j \in \Omega} \Psi_D' \left(|u_i - f_j|^2 \right) w_D \left(|x_i - x_j|^2 \right) f_j}{\sum_{j \in \Omega} \Psi_D' \left(|u_i - f_j|^2 \right) w_D \left(|x_i - x_j|^2 \right)} \tag{2}$$

This scheme is very similar to another well-established filtering technique known in image processing: the bilateral filter presented by Tomasi and Manduchi [16]. The bilateral filter can be obtained by replacing f_j with u_j in (2). Similar to the above reasoning the bilateral filter can be thought of as minimisation scheme for a nonlocal smoothness term:

$$E_S(u) = \sum_{i\in\Omega}\sum_{j\in\Omega} \Psi_S\left(|u_i - u_j|^2\right) w_S\left(|x_i - x_j|^2\right) \quad . \tag{3}$$

We keep in mind that a minimisation of (3) would lead to a constant image with an arbitrary grey value, since the initial image f does not appear in E_S. Nevertheless, the bilateral filter can be seen as the first step of an iterative minimisation procedure for (3).

The functional E of the NDS filter presented in [8] is a linear combination of both data and smoothness terms:

$$\begin{aligned} E(u) = \alpha &\sum_{i\in\Omega}\sum_{j\in\Omega} \Psi_D\left(|u_i - f_j|^2\right) w_D\left(|x_i - x_j|^2\right) \\ &+ (1-\alpha)\sum_{i\in\Omega}\sum_{j\in\Omega} \Psi_S\left(|u_i - u_j|^2\right) w_S\left(|x_i - x_j|^2\right) \quad . \end{aligned} \tag{4}$$

Here we have incorporated a similarity constraint which can lead to non-flat minimisers and a smoothness constraint. The spatial weights w_D and w_S incorporate the spatial distance between pixel positions x_i and x_j while the tonal weights Ψ_D and Ψ_S penalise high deviations between the corresponding grey values.

Table 1 shows some possible choices for the tonal weights Ψ_D and Ψ_S. The NDS functional (4) allows to express a lot of different models, so it is natural that the tonal weights are motivated from different contexts. The list in Table 1 is clearly not meant to be complete since there is a whole variety of possible penalisers at hand. The choice of a special one should be motivated from the type of noise and image, but this is not within the scope of this article.

Table 1. Possible choices for tonal weights Ψ.

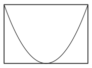
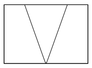
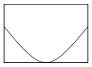
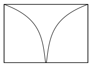
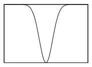
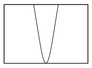
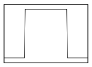
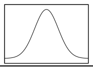
$\Psi(s^2)$		$\Psi'(s^2)$	known in the context of
s^2		1	Tikhonov regularisation [15]
$2\left(\sqrt{s^2+\varepsilon^2}-\varepsilon\right)$		$\left(s^2+\varepsilon^2\right)^{-\frac{1}{2}}$	regularised total variation [12]
$2\lambda^2\left(\sqrt{1+\frac{s^2}{\lambda^2}}-1\right)$		$\left(1+\frac{s^2}{\lambda^2}\right)^{-\frac{1}{2}}$	nonlinear regularisation, Charbonnier *et al.* [2]
$\lambda^2\log\left(1+\frac{s^2}{\lambda^2}\right)$		$\left(1+\frac{s^2}{\lambda^2}\right)^{-1}$	nonlinear diffusion, Perona and Malik [10]
$\lambda^2\left(1-\exp\left(-\frac{s^2}{\lambda^2}\right)\right)$		$\exp\left(-\frac{s^2}{\lambda^2}\right)$	nonlinear diffusion, Perona and Malik [10]
$\min(s^2,\lambda^2)$		$\begin{cases}1 & \lvert s\rvert<\lambda\\ 0 & \text{else}\end{cases}$	segmentation, Mumford and Shah [9]

Table 2. Possible choices for spatial weights w.

$w(s^2)$			known in the context of
$\begin{cases} 1 & \lvert s \rvert < \lambda \\ 0 & \text{else} \end{cases}$		hard window	locally orderless images, Koenderink and van Doorn [6]
$\exp\left(-\frac{s^2}{\lambda^2}\right)$		soft window	Chu *et al.*, [3]

Two simple examples of functions which can lead as spatial weights are displayed in Table 2. They both have in common that they are symmetric. Since in our model (4) we only use $w(s^2)$ we only plug in nonnegative values and this symmetry is obtained automatically. Essentially the same model allows to use nonsymmetric spatial weights, too. We also have chosen spatial weights which are between 0 and 1 and have their maximum in the point 0. This makes sure that the pixel itself is taken into consideration with the highest weight. Centering the spatial weight in the data term around a number different from 0 would perform a shift of the whole image during filtering.

3 Minimisation Methods

After discussing the derivation and the meaning of the NDS functional we now study different methods to minimise it. All numerical minimisation methods are based on conditions on the derivatives of E so we now calculate the first and second partial derivatives of E.

Taking the partial derivatives of the data term (1) yields

$$\frac{\partial E_D}{\partial u_k} = 2 \sum_{j \in \Omega} \Psi_D' \left(|u_k - f_j|^2 \right) (u_k - f_j) w_D \left(|x_k - x_j|^2 \right) \tag{5}$$

$$\frac{\partial^2 E_D}{\partial u_k \partial u_l} = \begin{cases} 2 \displaystyle\sum_{j \in \Omega} \left[2\Psi_D'' \left(|u_l - f_j|^2 \right) (u_l - f_j)^2 \right. & \\ \qquad \left. + \Psi_D' \left(|u_l - f_j|^2 \right) \right] w_D \left(|x_l - x_j|^2 \right) & l = k \\ 0 & l \neq k \end{cases} \tag{6}$$

In a similar way we calculate the derivatives of the smoothness term (3) which leads to

$$\frac{\partial E_S}{\partial u_k} = 4 \sum_{j \in \Omega} \Psi_S' \left(|u_k - u_j|^2 \right) (u_k - u_j) w_S \left(|x_k - x_j|^2 \right) \tag{7}$$

$$\frac{\partial^2 E_S}{\partial u_k \partial u_l} = \begin{cases} 4 \sum\limits_{j\in\Omega} \left[2\Psi_S''\left(|u_l - u_j|^2\right)(u_l - u_j)^2 \right. & \\ \qquad \left. + (1-\delta_{lj})\Psi_S'\left(|u_l - u_j|^2\right)\right] w_s\left(|x_l - x_j|^2\right) & l = k \\ -4\left[2\Psi_S''\left(|u_k - u_l|^2\right)(u_k - u_l)^2\right. & \\ \qquad \left. + \Psi_S'(|u_k - u_l|^2)\right] w_S\left(|x_k - x_l|^2\right) & l \neq k \end{cases} \tag{8}$$

In the second derivatives δ_{lj} denotes the Kronecker symbol $\delta_{lj} = \begin{cases} 1 & l = j \\ 0 & \text{else} \end{cases}$.

It is clear the complete derivatives then have the form

$$\frac{\partial E}{\partial u_i} = \alpha \frac{\partial E_D}{\partial u_i} + (1-\alpha)\frac{\partial E_S}{\partial u_i} \quad , \tag{9}$$

and the corresponding sum for the second derivatives. Having these derivatives at hand we can now study the concrete minimisation algorithms.

3.1 Jacobi Method – Fixed-Point Iteration

For a critical point u of the energy functional E we have

$$\nabla E(u) = 0 \quad \Longleftrightarrow \quad \frac{\partial E}{\partial u_i} = 0 \quad \text{for all } i \in \{1, \ldots, n\} \ . \tag{10}$$

We define the abbreviations

$$d_{i,j} := \Psi_D'\left(|u_i - f_j|^2\right) w_D\left(|x_i - x_j|^2\right) \quad , \tag{11}$$

$$s_{i,j} := \Psi_S'\left(|u_i - u_j|^2\right) w_S\left(|x_i - x_j|^2\right) \tag{12}$$

which help us to rewrite (10) as

$$0 = \alpha \sum_{j\in\Omega} d_{i,j}(u_i - f_j) + 2(1-\alpha)\sum_{j\in\Omega} s_{i,j}(u_i - u_j) \tag{13}$$

where we use the partial derivatives shown in (5) and (7). This can be transformed into fixed point form

$$u_i = \frac{\alpha \sum_{j\in\Omega} d_{i,j} f_j + (1-\alpha)\sum_{j\in\Omega} s_{i,j} u_j}{\alpha \sum_{j\in\Omega} d_{i,j} + (1-\alpha)\sum_{j\in\Omega} s_{i,j}} \quad . \tag{14}$$

To have a positive denominator we assume that $\Psi_{\{S,D\}}'(s^2) > 0$, i. e., the penalisers are monotonically increasing. Furthermore we assume that $w_{\{S,D\}}(s^2) \geq 0$ and $w_{\{S,D\}}(0) > 0$ for the spatial weights. We use this equation to build up a first iterative method to minimise the value of E where an additional index k denotes the iteration number. Note that $d_{i,j}$ and $s_{i,j}$ also depend on the evolving image u^k and thus also get a superscript to denote the iteration level involved. The corresponding fixed point iteration then reads as

$$u_i^0 := f_i \ , \tag{15}$$

$$u_i^{k+1} := \frac{\alpha \sum_{j\in\Omega} d_{i,j}^k f_j + 2(1-\alpha) \sum_{j\in\Omega} s_{i,j}^k u_j^k}{\alpha \sum_{j\in\Omega} d_{i,j}^k + 2(1-\alpha) \sum_{j\in\Omega} s_{i,j}^k} \ . \tag{16}$$

With our assumptions on $\Psi_{\{D,S\}}$ and $w_{\{D,S\}}$ from above we know that $d_{i,j}^k \geq 0$ and $s_{i,j}^k \geq 0$ for all i, j, k. That means in (16), u_i^{k+1} is calculated as a convex combination of grey values of the initial image f_j and of the last iteration step u_j^k. Thus we have

$$\min_{j\in\Omega}\{u_j^k, f_j\} \leq u_i^{k+1} \leq \max_{j\in\Omega}\{u_j^k, f_j\} \quad \text{for all } k \in \mathbb{N} \ . \tag{17}$$

Induction shows that the fixed point scheme (16) satisfies a maximum-minimum principle, i. e.

$$\min_{j\in\Omega}\{f_j\} \leq u_i^k \leq \max_{j\in\Omega}\{f_j\} \quad \text{for all } i \in \Omega, k \in \mathbb{N} \ . \tag{18}$$

Let us now consider the set

$$M := \{u \in \mathbb{R}^n \mid \|u\|_\infty \leq \|f\|_\infty\} \tag{19}$$

with the norm $\|u\|_\infty := \max_{j\in\Omega} |u_j|$. $M \neq \emptyset$ is compact and convex. Writing our scheme (16) in the form $u^{k+1} = F(u^k)$ with $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$, the maximum-minimum stability implies that $F(M) \subseteq M$. With our requirements on $\Psi_{\{D,S\}}$ and $w_{\{D,S\}}$, the denominator in (16) is always larger than zero. This means that each component $F_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ is continuous with respect to the norm $\|\cdot\|_\infty$. Since this holds for all i, we know that $F : (\mathbb{R}^n, \|\cdot\|_\infty) \longrightarrow (\mathbb{R}^n, \|\cdot\|_\infty)$ is continuous. Then Brouwer's fixed point theorem (see for example [18, page 51]) shows that F has a fixed point in M.

In the fixed point iteration scheme (16) we calculate u^{k+1} using only components of the vector u^k of the old iteration level:

$$u_i^{k+1} := F_i(u^k) \quad \text{for all } i \in \Omega, k \in \mathbb{N} \ . \tag{20}$$

Such a method can also be called a nonlinear Jacobi method.

3.2 Newton's Method

We search a zero of the gradient $\nabla E(u) = 0$. To this end we use Newton's method for the function ∇E:

$$u^{k+1} = u^k - H(E, u^k)^{-1} \, \nabla E(u^k) \ , \tag{21}$$

where $H(E, u^k)$ is the Hessian matrix of E at the point u^k. In each step of (21) we have to solve a linear system of equations. This system of equations can only be solved if the Hessian matrix is invertible which is the case for a convex functional E. That means we cannot use Newton's method for all penalisers shown in the last section. If both $\Psi_D(s^2)$ and $\Psi_S(s^2)$ are convex in s, i. e. $2\Psi''(s^2)s^2 + \Psi'(s^2) > 0$, the Hessian matrix $H(E, u^k)$ has positive diagonal entries and is strictly diagonally dominant. This does not only allow us to solve the linear system of equations, but it also gives us the possibility to use a whole variety of iterative solution algorithms like the Gauß-Seidel, successive overrelaxation, or conjugate gradient method. We have chosen to use the Gauß-Seidel method here to solve the linear system of equations.

A practical observation shows that the steps of Newton's method are often too long. Thus we have used a simple line-search strategy:

$$u^{k+1} = u^k - \sigma_k H(E, u^k)^{-1} \nabla E(u^k) \tag{22}$$

with $\sigma_k \in (0, 1]$. We try $\sigma_k = 1, \frac{1}{2}, \frac{1}{4}, \ldots$ until the energy is decreasing in the step: $E(u^{k+1}) < E(u^k)$.

It is clear that one step of Newton's method is much more expensive than one fixed point iteration step. Nevertheless, numerical examples will show that the whole process can still converge faster.

3.3 Gauß-Seidel Method

Instead of the nonlinear Jacobi method (20) one can also use a nonlinear Gauß-Seidel method which involves pixels of the old and the new iteration level. For each pixel $u_i =: x^0$, we perform m steps of a local fixed point iteration

$$x^{l+1} := F_i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, x^l, u_{i+1}^k, \ldots, u_n^k) \quad l = 1, 2, 3, \ldots \tag{23}$$

and set $u_i^{k+1} := x^m$ afterwards. Since these inner steps satisfy a maximum-minimum principle, the whole Gauß-Seidel method does. Thus one can apply the same reasoning as above and gets the existence of fixed points for the equation.

3.4 Gauß-Seidel Newton Method

Here we solve the single component equations with Newton's method. We start with the pixel value $x^0 = u_i^k$ of the last iteration level and set

$$x^{l+1} = x^l - \sigma_l \left(\frac{\partial^2 E}{\partial u_i^2}(\tilde{u}) \right)^{-1} \frac{\partial E}{\partial u_i}(\tilde{u}) \tag{24}$$

with $\tilde{u} = (u_1^{k+1}, \ldots, u_{i-1}^{k+1}, x^l, u_{i+1}^k, \ldots, u_n^k)$. After m steps of this method we set $u_i^{k+1} = x^m$ and proceed with the next pixel. The only difference is that we use the criterion $E_{loc}(x^{l+1}) < E_{loc}(x^l)$ for the choice of the step size σ_l where the local energy is defined as

$$\begin{aligned} E_{loc}(u) = \alpha \sum_{j \in \Omega} \Psi_D \left(|x^l - f_j|^2 \right) w_D \left(|x_i - x_j|^2 \right) \\ + (1-\alpha) \sum_{j \in \Omega} \Psi_S \left(|x^l - \tilde{u}_j|^2 \right) w_S \left(|x_i - x_j|^2 \right) \ . \end{aligned} \tag{25}$$

We should note that besides the number of (outer) iterations, all methods except of the Jacobi method have the number of inner iterations as an additional parameter for the numerics.

4 Numerical Experiments

Now we investigate the practical behaviour of the methods presented in the last section. We use the two stopping criteria

$$\|u^{k+1} - u^k\|_2 < a \quad \text{and} \quad |E(u^{k+1}) - E(u^k)| < b \ . \tag{26}$$

That means we stop the algorithm if the changes of both the evolving image (in terms of the Euclidean norm) and the energy value are smaller than prescribed limits a and b. The results of the 1-D example are displayed in Fig. 1 and Table 3. Here we have Gaussian noise, and we have chosen $\Psi_D(s^2) = s^2$, $\Psi_S(s^2) = 2\left(\sqrt{s^2+\varepsilon^2} - \varepsilon\right)$ with $\varepsilon = 0.01$, and $w_D(s^2) = w_S(s^2) = 1.0$ inside a data term window of size 7 and a smoothness term window of size 11 with $\alpha = 0.5$. The number of inner iterations was optimised to yield a fast convergence for each method. We see that Newton's method is the fastest one in this case.

Fig. 2 and Table 4 contain the results of the 2-D experiments. For the removal of salt-and-pepper noise we chose $\Psi_D(s^2) = 2\left(\sqrt{s^2+\varepsilon^2} - \varepsilon\right)$ with

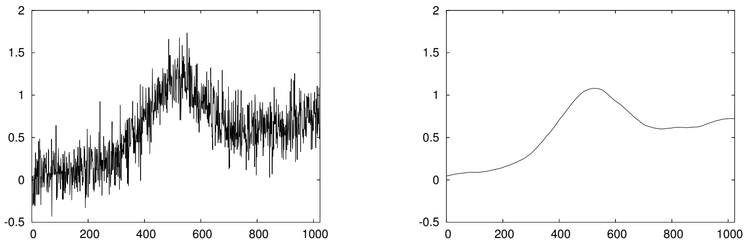


Fig. 1. Denoising experiment in 1-D. Left: Test signal with additive Gaussian noise with zero mean, size 1024 pixels. Right: Denoised version of the signal.

Table 3. Denoising experiment in 1-D with $a = 10^{-2}$ and $b = 10^{-6}$.

method	iterations	inner it.	energy	time [sec]
Fixed point	1309	–	165.70820	3.332
Newton	25	60	165.70807	0.515
Gauß-Seidel	842	1	165.70815	2.193
G.-S. Newton	683	1	165.70813	5.739

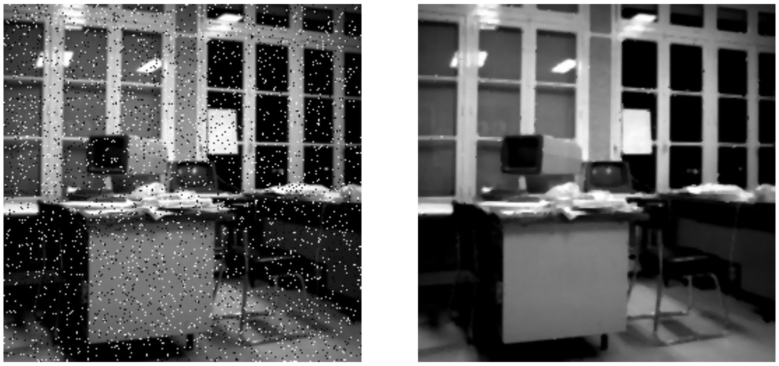


Fig. 2. Denoising experiment in 2-D. Left: Test image with salt-and-pepper noise (256 × 256 pixels). Right: Denoised version of the image.

$\varepsilon = 0.01$, $\Psi_S(s^2) = \left(1 + \frac{s^2}{\lambda^2}\right)^{-\frac{1}{2}}$ with $\lambda = 0.1$. We set $w_D(s^2) = w_S(s^2) = 1.0$ with both windows of size 3 and $\alpha = 0.95$. Here we have the opposite case, and the simple fixed point scheme is faster than Newton's method. We have performed some more experiments indicating that this does not depend on the dimension of the problem but on the choice of penalisers.

5 Conclusions

We have investigated four different algorithmic approaches for the variational image simplification NDS-model presented in [8]. For schemes based on fixed point iterations we have shown the existence of fixed points. Newton's method is only applicable for a certain class of convex penalisers. We have seen with practical examples that in terms of running time we cannot prefer one single method in general. Currently we are considering the question if other numerical approaches based on multigrid ideas could help to reduce the running time especially of the fixed point approaches applicable for all weighting types.

Acknowledgements. We gratefully acknowledge partly funding within the priority programme SPP1114 of the *Deutsche Forschungsgemeinschaft (DFG)*, project WE 2602/2-2.

Table 4. Denoising experiment in 2-D with $a = b = 10^3$.

method	iterations	inner it.	energy	time [sec]
Fixed point	38	–	$1.86 \cdot 10^7$	8.175
Newton	25	5	$2.07 \cdot 10^7$	89.239
Gauß-Seidel	3	25	$1.86 \cdot 10^7$	8.502
G.-S. Newton	6	2	$1.86 \cdot 10^7$	23.317

References

1. D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847, June 2002.
2. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. *Proc. IEEE International Conference on Image Processing (ICIP-94, Austin, Nov. 13-16, 1994)*, 2:168–172, 1994.
3. C. K. Chu, I. K. Glad, F. Godtliebsen, and J. S. Marron. Edge-preserving smoothers for image processing. *Journal of the American Statistical Association*, 93(442):526–541, June 1998.
4. M. Elad. On the origin of the bilateral filter and ways to improve it. *IEEE Transactions on Image Processing*, 11(10):1141–1151, October 2002.
5. L. D. Griffin. Mean, median and mode filtering of images. *Proceedings Royal Society of London A*, 456:2995 – 3004, 2000.
6. J. J. Koenderink and A. L. Van Doorn. The structure of locally orderless images. *International Journal of Computer Vision*, 31(2/3):159–168, 1999.
7. J.-S. Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24:255–269, 1983.
8. P. Mrázek, J. Weickert, and A. Bruhn. On robust estimation and smoothing with spatial and tonal kernels. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, *Geometric Properties for Incomplete Data*, volume 31 of *Computational Imaging and Vision*. Springer, November 2005.
9. D. Mumford and J. Shah. Optimal approximation of piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–685, 1989.
10. P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:629–639, 1990.
11. J. Polzehl and V. Spokoiny. Adaptive weights smoothing with applications to image restoration. *Journal of the Royal Statistical Society, Series B*, 62(2):335 – 354, 2000.
12. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
13. P. Saint-Marc, J.-S. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):514 – 529, June 1991.
14. S. M. Smith and J. M. Brady. SUSAN – A new approach to low lewel image processing. *International Journal of Computer Vision*, 23(1):43–78, 1997.
15. A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4(2):1035–1038, 1963.
16. C. Tomasi and R. Manduchi. Bilateral filtering for gray and colour images. In *Proc. of the 1998 IEEE International Conference on Computer Vision*, pages 839–846, Bombay, India, January 1998. Narosa Publishing House.
17. G. Winkler, V. Aurich, K. Hahn, and A. Martin. Noise reduction in images: Some recent edge-preserving methods. *Pattern Recognition and Image Analysis*, 9(4):749–766, 1999.
18. E. Zeidler. *Nonlinear Functional Analysis and Applications I: Fixed-Point Theorems*. Springer-Verlag New York, 1986.