Winkler, Liebscher, Aurich:

Smoothers for Discontinuous Signals

Sonderforschungsbereich 386, Paper 146 (1999)

Online unter: http://epub.ub.uni-muenchen.de/

Smoothers for Discontinuous Signals

by G. Winkler, V. Liebscher [1] and V. Aurich[2]

Abstract

First we explain the interplay between robust loss functions, nonlinear filters and Bayes smoothers for edge-preserving image reconstruction. Then we prove the surprising fact that maximum posterior smoothers are nonlinear filters. A (generalized) Potts prior for segmentation and piecewise smoothing of noisy signals and images is adopted. For one-dimensional signals, an exact solution for the maximum posterior mode - based on dynamic programming - is derived. After, some results on the performance of nonlinear filters on jumps and ramps we finally introduce a cascade of nonlinear filters with varying scale parameters and discuss the choice of parameters for segmentation and piecewise smoothing.

Keywords: Image processing, jump preserving smoothing, filters, Potts model

1 Introduction

Spatio-temporal statistics is penetrating into image analysis more and more. This leads to both, new methods and a better understanding of algorithms from apparently different fields like computer science or engineering. Moreover, there is rapidly increasing interest in models and methods which can deal with *discontinuous phenomena*. Focus is on identification of discontinuities in corrupted signal or image data, since relevant basic features like jumps, spikes and boundaries are to be preserved during noise reduction. This is of particular importance in applications our group presently is concerned with, like processing responses to outer stimuli in functional magnetic resonance imaging, detection of microcalcification in X-ray mammography and similar applications in medical imaging and life sciences.

[1]Institute of Biomathematics and Biometrics
GSF - National Research Center for Environment and Health,
Postfach 1129, D-85758 Oberschleißheim, Germany,
liebsche(gwinkler)@GSF.de, http://www.gsf.de/institute/ibb/

[2]Mathematical Institute, Heinrich-Heine University,
Universitätsstr. 1, D-40225 Düsseldorf, Germany,
aurich@cs.uni-duesseldorf.de,
http://www.cs.uni-duesseldorf.de/aurich/index.html

Recent and very recent methods we have in mind were developed by the authors and their groups ([29], [3], [20], [2], [12]), [31], [19]), as well as by others ([4], [8], [9], [21](a), (b), (c), [5]). We started to discuss and compare such methods in the common framework of Bayesian image analysis [30] and robust statistics [15], [14] in the paper [31]. We interpreted them in the language of 'energy functions' and this way worked out common aspects. Their performance was compared by way of application to suitably constructed phantoms which show typical features like smooth regions, sudden changes of intensity, canyons or spikes (for instance from [5]). The reasoning was on an informal level.

Typical examples of such approaches are *(i)* Bayesian methods similar to those in S. and D. Geman(1984), [8], which also comprise those suggested in A. Blake and A. Zisserman (1987), [4], cf. also G. Winkler (1995), [30], *(ii)* (local) M-smoothers with score functions redescending to zero introduced in C.K. Chu, I. Glad, F. Godtliebsen and J.S. Marron (1998), [5], *(iii)* nonlinear filters or σ-filters studied for example in J. Weule (1994), [29] or F. Godtliebsen, E. Spjøtvoll and J.S. Marron (1994), [9], *(iv)* chains of nonlinear filters with varying scale-parameters developed and studied in a series of papers by V. Aurich and his group (1994 – 98), [2], [3], [20], [29], *(v)* adaptive weights smoothing adressed recently by J. Polzehl and V.G. Spokoiny (1998), [21], [22], [23]. *(vi)* local radial-basis-function networks introduced by K. Hahn and Th. Waschulzik (1998), [12].

In this paper we restrict ourselves to two apparently antagonistic approaches: Bayesian segmentation and piecewise smoothing (i) on the one hand and nonlinear filtering (iii), (iv), on the other hand. It turns out that there are close relations between these methods; the missing link inbetween is (local) M-smoothing discussed in [5]. We are interested in such relations since all these methods have obvious merits and shortcomings which - loosely spoken - are opposite to each other. The Bayesian method for example rests on a beautiful and transparent model including a natural quality measure; on the other hand it leads to nearly intractable optimization problems, for instance to compute maximum posterior estimates. Suboptimal solutions are obtained by Markov-Chain-Monte-Carlo algorithms like simulated annealing; but these can be discouragingly slow and, even worse, *they are not exact*. Monitoring the output to check mixing and convergence may be cumbersome. Thus imperfection of convergence frequently cannot be told from imperfection of the model or of model parameters (cf. [10]). As opposed to this (chains of) nonlinear filters converge very fast and give excellent results but the theoretical foundations presently are insufficient. In fact, even i.i.d. noise is transformed into coloured noise in an obscure way even by a single filter step. Thus these filter chains share theoretical shortcomings with other

iterated nonlinear filters (a thorough discussion of these aspects for iterated medians is given in [16] and [27]). Hopefully, relations between the methods can be established by which we might gain more insight into one method from what we learned about the other.

The plan of this paper is as follows: Having introduced basic notions and concepts we briefly sketch relations between some methods on an intuitive level. Then we prove a somewhat surprising result: under mild conditions maximum posterior mode estimations amounts to σ-filtering in a sense to be made precise. Then we turn to the Bayesian approach; focus is on the simple case of a Potts prior distribution suited for segmentation or regression onto piecewise constant signals. A generalization in the spirit of [4] allows for piecewise smoothing. For one-dimensional signals we derive an algorithm which computes *exact (Bayesian) maximum a posterior modes* estimation (MAP) estimates extremely fast and therefore allows to scan the family of estimates over the whole range of hyperparameters. Then we introduce a chain of nonlinear filters with varying scale-parameters. Having derived first properties like the ability to preserve jumps or to sharpen blurred edges we give heuristic arguments for the optimal choice of scale-parameters.

2 Relations between Smoothers

In this section we look at image smoothers from three different points of view: minimizers of *loss functions*, *filters* i.e. convex combinations of data and *Bayes estimators*. Focus is on segmentation and edge-preserving smoothing.

We first try to bring out relations between these seemingly different notions. It turns out that (local) minimizers of M-functions and iteratively reweighted squares algorithms practically are equivalent and thus the former are closely related to W-estimators. Simultaneously nonlinear or σ-filters are embedded into this framework and hence a link to w-estimation is established. A loose connection to MAP-estimation is established as well.

In the second part of this section we study the relation between nonlinear filters and maximum posterior modes in more detail. We give a *rigorous* proof for the somewhat surprising fact, that under mild conditions MAP-estimation is a special case of nonlinear filtering.

Let us first fix some notation. Let S denote a finite set of design points s in Euclidean space $\mathbb{R}^d$. They need not necessarily be equispaced but in most examples they are. The design points will frequently be called *pixels*. A *signal* or 'image' is a family $x = (x_s)_{s \in S}$ of *intensities* or 'grey-values' from a supply G which may be finite or not; if convenient we shall write $x(s)$ instead of x_s. For simplicity we assume real intensities $x_s \in \mathbb{R}$.

2.1 Loss Functions, Sigma-Filters and *MAP*-Smoothers

Nothing in this subsection is really new; some of the observations are scattered over the literature others seem to be folklore. Nevertheless, we found it useful to put different aspects together and comment on their relations. In the course of the following discussion we shall be somewhat sloppy with derivatives. We shall tacitly assume generalized derivatives φ' if for instance a function φ has isolated jumps and is differentiable elsewhere. Thereby we include indicator functions of intervals or trapezoidal functions which frequently arise in imaging.

Suppose that data $(y_t)_{t\in S}$ is observed, i.e. a realization of random variables $(Y_t)_{t\in S}$ (in this paper we do not care about missing data). The aim is to infer signals x from data y under certain regularity conditions or prior expectations. Sometimes these are given in explicit form, for example as prior distributions, regularization terms or penalties. Sometimes they are hidden behind the formalism of the algorithm like in the case of filters.

Robustness aspects enter in a natural way since edges, i.e. abrupt changes in intensity, are important image features. Smoothing out noise in a smooth part of the image - say in a moving window - should not be affected by the contamination caused by intensities beyond an edge gradually entering the window. This requirement is equivalent to edge preservation.

Let us start now from the very beginning. Given real random observations $Y_1, \ldots, Y_n$ the standard form of a loss-function for a location parameter is

$$\Phi(\vartheta) = \sum_{i=1}^{n} \varphi(Y_i - \vartheta).$$

Minimizers of Φ are called *M-estimators*. For local smoothing of an image at pixel s the observed values y_t, $t \in B(s)$, in a window $B(s)$ around s may be plugged in for the Y_i. The idea behind is that locally at least the majority of the Y_t are approximately i.i.d. Hence loss functions of this type are suitable for 'piecewise constant' or locally slowly varying images. Robustness is built in preservation of discontinuities. It is mirrored by 'cup'-shaped function φ with 'derivatives redescending to zero'; this basically means that $|\varphi(u)| \uparrow c$ for some constant c as $|u| \uparrow \infty$. Usually the functions φ are symmetric with minimum at 0 and nondecreasing on the positive half-line. Spacial influence of data frequently is modelled by soft windows rather than hard ones. They are given by functions $v(u) = h(\|u\|)$ where $\|\cdot\|$ is Euclidean norm and h is a kernel function similar in shape to $-\varphi$. They weight the influence of remote pixels down. Throughout this paper we shall generically denote 'cup'-functions by Greek letters like φ or ψ and bell-shaped functions by italic letters like v or w.

For the intensity estimate at pixel s one thereby arrives at a loss-function

$$\Phi_s(\vartheta) = \sum_{t \in S} \varphi(y_t - \vartheta) v(t - s). \tag{1}$$

Example 2.1 (a) Hard windows are given by indicator functions like

$$v(t - s) = \mathbf{1}_{[-a,a]}(\|t - s\|)$$

in the isotropic case. Frequently Gaußian functions

$$v(u) = g(u/\tau)/\tau, \;\; g(u) = \exp(-\|u\|^2/2)$$

are adopted.
(b) For the intensity weight functions negative Gaußians

$$\varphi(u) = -g(u/\sigma)/\sigma$$

are most popular. Indicator functions - corresponding to truncated means - are of interest as well. For computational reasons trapezoidal φ are used for instance in [7]. Functions

$$\varphi(u) = \min\{(\lambda \cdot u)^2, \gamma\}, \;\; \gamma > 0, \tag{2}$$

arise naturally in [4]. We shall argue below that they are intimately connected with edge detection.

Contour lines of the map

$$(s, \vartheta) \longmapsto \Phi_s(\vartheta) \tag{3}$$

are displayed in Fig. 1 for the Gaußian case. Data are simulated from the phantom in [5].

A view at the contour lines of the function $(s, \vartheta) \mapsto \Phi_s(\vartheta)$, displayed in Fig. 1 shows that the global minimizers of (1) will respect boundaries between reasonably large plateaus. A spike in a pixel s, however, in general cannot prevail since the superposition of many terms $\varphi(y_t - \vartheta)v(t - t)$ for t near s and similar values y_t will produce a deeper valley along the cut $\{s\} \times G$ than the single spike term $\varphi(y_s - \vartheta)v(t - s)$ with $y_s \gg y_t$. As a remedy, *local* minimizers of (1) are proposed in [5] and in fact, these *local M-smoothers* show an excellent performance for a large variety of images. More precisely, the authors for each s choose the next local minimum of Φ_s which is downhill from y_s.

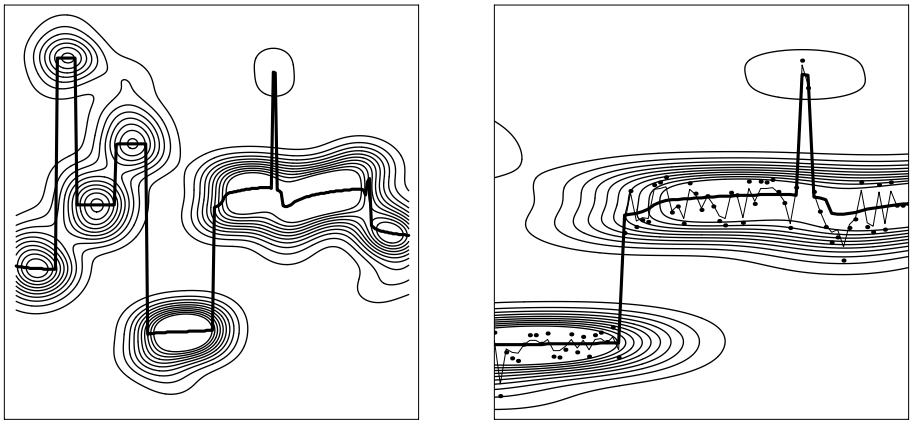


Figure 1: Data y (dots) and contour lines for the map (3), local M-smoothing (solid line) and w- or σ-filter smoothing (wiggly line).

It was observed in [25] and independently by the first author that local M-smoothing is intimately connected to W-estimation presumably introduced by J.W. Tukey, 1970 (cf. [14], Chapter 2.3.d). In the present context W-estimators are defined by the fixed point equation

$$\vartheta^* = \sum_{t\in S} w(y_t - \vartheta^*)v(t-s)y_t \Big/ \sum_{t\in S} w(y_t - \vartheta^*)v(t-s)\,, \tag{4}$$

where w is some kernel function. As a convex combination of observations the W-estimator is of filter type; it is nonlinear since the estimate itself enters the filter weights. One usually resorts to the iterative algorithm

$$\begin{aligned}\vartheta_{k+1} &= \sum_{t\in S} w(y_t - \vartheta_k)v(t-s)y_t \Big/ \sum_{t\in S} w(y_t - \vartheta_k)v(t-s) \\ &= \vartheta_k + \sum_{t\in S} w(y_t - \vartheta_k)v(t-s)(y_t - \vartheta_k) \Big/ \sum_{t\in S} w(y_t - \vartheta_k)v(t-s)\,. \end{aligned} \tag{5}$$

If a function φ is (up to a constant) defined by $\varphi'(u) = u \cdot w(u)$ then (5) reads

$$\vartheta_{k+1} = \vartheta_k - \gamma_k(\vartheta_k) \sum_{t\in S} \varphi'(y_t - \vartheta_k)v(t-s) \tag{6}$$

with adaptive gains γ_k. This reformulation shows that (if we are lucky with convergence) the algorithm initialized with y_s results precisely in the local M-estimate introduced above. The only difference is that the latter is formulated as a general minimization problem whereas the former is given by a fixed point equation (4) together with the special algorithm (5). The simulations in [24] indicate that the algorithm (6) initiated with data (y_s) in practice gives the same result as local M-smoothing with other optimization techniques albeit we are not aware of a rigorous proof.

Next we conclude from (6) that (5) is an iteratively reweighted least squares algorithm. The generic step transforms an input ϑ into an output

$$\tilde{\vartheta} = \sum_{t\in S} w(y_t - \vartheta)v(t-s)y_t \Big/ \sum_{t\in S} w(y_t - \vartheta)v(t-s) \ ,$$

which implies

$$\sum_{t\in S} w(y_t - \vartheta)v(t-s)(y_t - \tilde{\vartheta}) = 0$$

and hence

$$\tilde{\vartheta} = \operatorname*{argmin}_{\lambda} \sum_{t\in S} w(y_t - \vartheta)v(t-s)(y_t - \lambda)^2. \tag{7}$$

Thus each step is an estimator associated to its own loss function (7).

The *first step* $\vartheta_0 \mapsto \vartheta_1$ in (6) is called a *w-estimator* (cf. [14], Chap. 2.3.d). Initiated with data y it has output

$$\mathcal{F}y = \left(\sum_{t\in S} w(y_t - y_s)v(t-s)y_t \Big/ \sum_{t\in S} w(y_t - y_s)v(t-s) \ : s \in S \right). \tag{8}$$

We recognize this nonlinear filter as the *σ-filter*, well-known in imaging for a long time; if v and w are Gaußian then it is called the *nonlinear Gaußian filter* (*NLGF*). The above derivation clearly shows that the σ-filter drives data y_s towards the local M-estimate but in general gets stuck before they are reached. Hence its output lies between the data and the output of the local M-smoother. This explains the observation that σ-filters have small-scale 'wiggliness' (cf. [5]).

Remark 2.2 (a) The original σ-filter by J.S. Lee (1983), [18], used indicator functions $w = \mathbf{1}_{[-\varepsilon,\epsilon]}$ and hard windows. Hence in window around s, for suitable ε it performs a test of significance to decide whether Y_t has mean y_s and takes only the mean of those y_t which pass the test.
(b)The case $w \equiv 1$ gives the Nadaraya-Watson kernel smoother (cf. [6]) which for Gaußian v simply is a *linear Gaußian filter*.

Recall that we established a one-to-one correspondence between local M-smoothers and iteratively reweighted least squares, which turned out to be a sequence of σ-filters. The correspondence between loss functions (1) and filter weights in (8) is given by the identity $\varphi'(u) = u \cdot w(u)$. If for example $\varphi = -g$ is a Gaußian function turned upside down then w is a Gaußian function; if $\varphi(u) = u^2$ then $w \equiv 1$ (the linear Gaußian filter); if $\varphi(u) = \min\{(\lambda u)^2, \gamma\}$, $\gamma > 0$, is a truncated square then $w(u) = \mathbf{1}_{[-\varepsilon,\varepsilon]}(u)$ with $\varepsilon = \sqrt{\gamma}/\lambda$; hence the associated σ-filter is a truncated mean. We shall argue below that this 'sharp cup' function φ corresponds to 'sharp boundaries'.

Finally we consider Bayesian a smoothers. Some notation is needed before. We give the definitions for discrete spaces only; for continuous spaces densities are plugged in. The *prior probabilities* $\Pi(x) > 0$, $\sum_x \Pi(x) = 1$, rate (favourable) regularity properties of the x. For each $x = (x_t)$, data $y = (y_t)$ is observed with probability (density) $\Pi(y|x)$. Given y the prior is modified to the *posterior (distribution)* $\Pi(x|y) = \Pi(x)\Pi(y|x)/\Pi(y)$. A popular estimate of the 'true image' is the *MAP*-estimate $x^* = \operatorname{argmax}_x \Pi(x|y)$. In the *Gibbsian formulation* this reads

$$\begin{aligned} \Pi(x) &\propto \exp(-K(x)),\ \ \Pi(y|x) \propto \exp(-D(x,y)), \\ \Pi(x|y) &\propto \exp(-K(x) - D(x,y)), \\ x^* &= \operatorname*{argmin}_x (K(x) + D(x,y)). \end{aligned}$$

The data term is determined by the observation device; hence the *prior energy* $K(x)$ is the interesting term.

For $s \in S$ consider the conditional prior $\Pi(z_s | x_t : t \neq s)$. The conditional prior energy is

$$-\ln \Pi(z_s | x_t : t \neq s) = K(z_s, x_t : t \neq s) + \text{const.}$$

A common way to construct prior energy functions is to plug in suitable loss functions like (1) whilst replacing the variables y_t by x_t. This results in

$$K(x) = \sum_{t \in S} \varphi(x_s - x_t) v(t - s).$$

Each of the above statements about φ and v holds *mutatis mutandis* for Bayesian models as well. Let now S be endowed with an undirected graph structure and call s and t *neighbours* if they are connected by an edge of the graph. This will be indicated by the symbol $s \sim t$. Assume further that neighbours have distance 1. Then with $v = \mathbf{1}_{\{1\}}(\|\cdot\|)$ we get

$$K(x) + D(x,y) = \sum_{s \sim t} \varphi(x_s - x_t) + D(x,y).$$

We now want to work out the relation between such priors and boundaries. This is easiest explained in the case of sharp cups (2). They can be written in the form

$$\psi(u) = \min\{(\lambda u)^2, \gamma\} = \min\{\lambda^2(u)^2(1-a)) + \gamma a : a \in \{0,1\}\}.$$

Setting

$$d(u) = \begin{cases} 0 \text{ if } u \leq \gamma^{1/2}/\lambda \\ 1 \text{ otherwise} \end{cases} \tag{9}$$

this reads

$$\psi(u) = (\lambda u)^2(1-d(u)) + \gamma d(u).$$

Having introduced binary variables $b = (b_{st} : s,t \in S, s \sim t)$, $b_{st} \in \{0,1\}$, we conclude that the following are equivalent:
(a) x^* minimizes

$$x \longmapsto \sum_{s \sim t} \psi(x_s - x_t) + D(x,y) \tag{10}$$

and $b^*_{st} = 0$ if $|x^*_s - x^*_t| \leq \gamma^{1/2}/\lambda$ and $b^*_{st} = 1$ elsewhere,
(b) (x^*, b^*) minimizes

$$(x,b) \longmapsto \sum_{s \sim t} \left(\lambda^2(x_s - x_t)^2(1-b_{st}) + \gamma b_{st}\right) + D(x,y). \tag{11}$$

Now we arrived at the classical model from [4]. Simultaneously it is a special case of [8]. This allows a new interpretation of this prior: the variables b_{st} are interpreted as active or inactive (micro) edges between neighbouring pixels s and t according to $b_{st} = 1$ or $b_{st} = 0$. Active edges correspond to discontinuities of intensity and 'switch off' smoothing. Thus $\{s \sim t : b_{st} = 1\}$ is a 'contour'. The terms γb_{st} penalize each active edge by $\gamma > 0$. Since their sum is γ times contour length, short and thus 'smooth' contours are favourable. If $b_{st} = 1$ then the quadratic smoothing term is switched off which – in view of the penalty – pays off if $\lambda^2(x_s - x_t)^2 > \gamma$. Small intensity differences are favourable. In summary, the prior favours smooth regions but allows for abrupt changes in intensity where there is evidence for a boundary in the data.

Let us stress that the reformulation in terms of edge elements provides a link between robust priors and edge-preserving smoothing with a conspicuous interpretation.

There still remains another interesting observation. Using the loss function of the σ-filter in (7) instead of (1) we get

$$K(x) = \sum_{s \sim t} (x_t - x_s)^2 w(x_t - x_s). \tag{12}$$

Specializing to the binary case $w = \mathbf{1}_{[-\varepsilon,\varepsilon]}$ with $\varepsilon = \gamma^{1/2}/\lambda$ we find that $(1-d) = w$ and hence MAP-smoothing with (12) is equivalent to the minimization of

$$(x, b) \longmapsto \sum_{s\sim t} \Big(\lambda^2 (x_s - x_t)^2 (1 - b_{st})\Big) + D(x, y).$$

This (11) without the penalty term γb_{st}. Therefore boundaries will be less smooth in accordance with wiggliness of the σ-filter.

We finally compare local M-smoothers, NLG filters and a chain of NLG filters in Fig. 2.

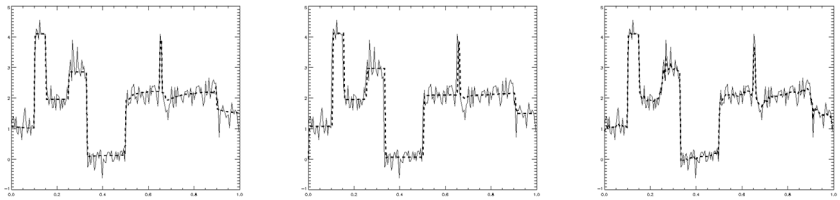


Figure 2: Noisy phantom smoothed by the local M-smoother, the NLG filter chain and a NLG filter.

2.2 Maximum Posterior Modes are Filters

In this section we prove that MAP-estimators are (nonlinear) filter under mild and natural conditions.

Let us first give a precise definition of the latter. The space $\{\ (x_s) : x_s \in \mathbb{R}\ \}$ of signals will be denoted by $\mathbb{X}$. Recall that S is finite; hence we may consider matrices $M = (M_{st})_{s,t\in S}$. If $M_{st} \geq 0$ and $\sum_{t\in S} M_{st} = 1$ then M is called a $(S \times S$-$)$ *stochastic matrix.*

Definition 2.3 *A map $\mathcal{F} : \mathbb{X} \to \mathbb{X}$ is called a* nonlinear filter *if there is a map $W : \mathbb{X} \to \mathbb{R}^{S\times S}$ into the set of stochastic matrices such that*

$$\mathcal{F}x = W(x)x.$$

$\mathcal{F}$ is a nonlinear filter if

$$(\mathcal{F}x)_s = \sum_{t\in S} W_{st}(x)x_t \tag{13}$$

for all $x \in \mathbb{X}$, $s \in S$. We shall use the term *filter* in this section for a nonlinear filter. Obviously, σ-filters introduced above are filters.

We are going to characterize filters by means of convexity. Let $\operatorname{conv} A$ denote the convex hull of A.

Lemma 2.4 *A map $\mathcal{F} : \mathbb{X} \to \mathbb{X}$ is a nonlinear filter if and only if*

$$(\mathcal{F}x)_s \in \operatorname{conv}\{\, x_t : t \in S \,\} \tag{14}$$

for all $x \in \mathbb{X}$ and $s \in S$

Proof. If $\mathcal{F}$ is a filter then $(\mathcal{F}x)_s$ is a convex combination of the values x_t, $t \in S$ by (13) and hence in the convex hull of these points. Conversely, if $(\mathcal{F}x)_s \in \operatorname{conv}\{\, x_t : t \in S \,\}$ then $(\mathcal{F}x)_s$ is a convex combination of $\{\, x_t : t \in S \,\}$. Hence there are nonnegative weights $W_{st}(x)$, $\sum_{t\in S} W_{st}(x) = 1$ with (13). Thus $\mathcal{F}$ is a filter.

Obviously, the function W is far from being unique. One can define an (almost) unique setting all weights to zero exept those for extremal x_t.

We are interested in maps $\mathcal{F}$ induced by MAP-estimates. More precisely, let

$$\mathcal{F}y = \operatorname*{argmin}_x H(x, y) = \operatorname*{argmin}_x (K(x) + D(x, y)). \tag{15}$$

In the following, we assume that the prior energy K only penalizes 'non-smoothness' of signals. This results in shift-invariance conditions like

$$K(x + c) = K(x),$$

where $(x + c)_s = x_s + c$, $c \in \mathbb{R}$. Under this assumption, we can fix a function $\tilde{K}$ with

$$K(x) = \tilde{K}((x_s - x_t)_{s\sim t}).$$

Typically, there are functions φ and ρ such that

$$K(x) = \sum_{s\sim t} \varphi(x_s - x_t), \tag{16}$$

$$D(x, y) = \sum_s \rho(x_s - y_s). \tag{17}$$

To state the main result, we introduce relations $\prec_y$, $y \in \mathbb{X}$, on $\mathbb{X}$ by

$$x \prec_y z \text{ if and only if } z_s \le x_s \le y_s \text{ or } z_s \ge x_s \ge y_s \text{ for all } s \in S;$$

Hence $x \prec_y z$ means that for each single site s the signal x_s is closer to y_s than z_s and that it is on the same side of y_s. Call K and D *monotonous* if

$$x \prec_0 z \;\Rightarrow\; \tilde{K}(x) \le \tilde{K}(z) \tag{18}$$

$$x \prec_y z, x \ne z \;\Rightarrow\; D(x, y) < D(z, y). \tag{19}$$

Theorem 2.5 *If* $K : \mathbb{X} \longmapsto \mathbb{R}$ *and* $D : \mathbb{X} \times \mathbb{X} \longmapsto \mathbb{R}$ *are monotonous then each map* $\mathcal{F} : \mathbb{X} \longmapsto \mathbb{X}$ *fulfilling* (15) *is a nonlinear filter.*

The formulation explicitly takes into account that minimizers of (15) in general are not unique.

Remark 2.6 Conditions (18) and (19) have simple interpretations. The first condition means that a signal x, all jumps of which are smaller than those of another signal z and of the same sign, is smoother than the signal z. In other words, K is a measure of smoothness in a precise sense. The second condition (19) simply means that the term $D(\cdot, y)$ - measuring fidelity to data y - is strictly smaller for x closer to y than z, i.e., D penalizes bias from y. One easily concludes that under these conditions constant signals are fixed points of $\mathcal{F}$. In this special case, this boils down to (14).

It does not matter whether K or D is assumed to be strictly monotonous. We decided on strict monotony of D since it fits better to the robust priors applied in section 3.

Proof. The proof is based on Lemma 2.4. For each $y \in \mathbb{X}$ we define the map $x \mapsto \hat{x}^y$ by

$$\hat{x}^y_s = \begin{cases} \max\{\, y_s : s \in S \,\} & \text{if } x_s > \max\{\, y_s : s \in S \,\} \\ \min\{\, y_s : s \in S \,\} & \text{if } x_s < \min\{\, y_s : s \in S \,\} \\ x_s & \text{otherwise} \end{cases},$$

cf. Fig. 3. It is easy to see that

$$\hat{x}^y \prec_y x, \;\; (x_s - x_t)_{s\sim t} \prec_0 (x_s - x_t)_{s\sim t}.$$

Moreover, if $x_s \notin \operatorname{conv}\{\, y_t : t \in S \,\}$ for some $s \in S$ then $\hat{x}^y \neq x$. As a consequence of the assumptions we find $H(x, y) > H(\hat{x}^y, y)$ for such x. Now we conclude from $\hat{x}^y \in \operatorname{conv}\{\, x_s : s \in S \,\}^S$ that all minimizers x^* of $x \mapsto H(x, y)$ fulfil $x^*_s \in \operatorname{conv}\{\, y_t : t \in S \,\}$ for all $s \in S$. Application of Lemma 2.4 completes the proof.

Corollary 2.7 *Suppose that* D *and* K *are given by* (17) *and* (16) *and* φ *and* ρ *are monotonous w.r.t.* $\prec_0$*; more precisely*

$$(u' < u \leq 0 \quad or \quad u' > u \geq 0) \Rightarrow \begin{cases} \varphi(u) \leq \varphi(u') \\ \rho(u) < \rho(u') \end{cases}$$

Then each MAP*-estimate* $\mathcal{F} : \mathbb{X} \to \mathbb{X}$ *is a filter.*

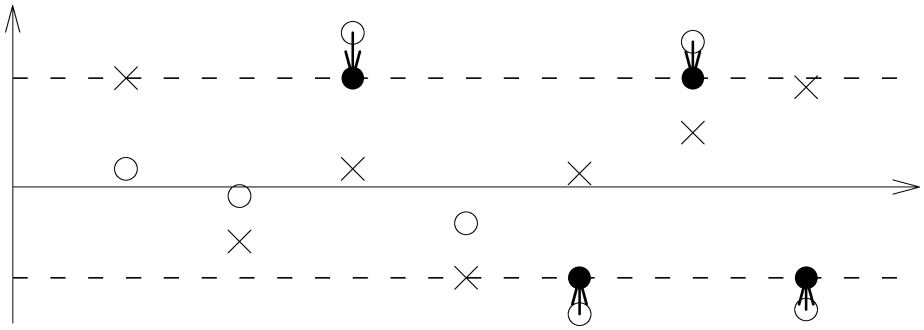


Figure 3: Illustration of the map $x \to \hat{x}^y$. The signal y is represented by crosses, x by circles and the points of $\hat{x}^y$ different from x by bullets.

A similar result holds for priors like

$$K(x) = \sum_{\substack{s \sim t \sim u \\ s \not\sim u}} \tilde{\varphi}(x_s - 2x_t + x_u).$$

These correspond to locally linear smoothing in contrast to locally constant smoothing considered above. For the corresponding MAP-estimate $\mathcal{F}$ not only constant signals but also linear signals are fixed points.

3 Generalized Potts Priors for Segmentation and Smoothing

We continue with MAP-estimation in the models (10). The main problems - not yet overcome - are estimation of hyperparameters and the numerical solution of the minimization problem. Focus is on optimization; but the considerations below will also shed some light on the choice of parameters. Usually such optimization problems are solved by stochastic relaxation techniques like Gibbs- or Metropolis annealing ([30], [11]). In cases like the Potts model specially taylored relaxation algorithms like the Swendson-Wang algorithm [26] may be adopted. But all stochastic relaxation algorithms cause considerable practical problems. They theoretically find minima but do not

realize that they are there (cf. [30], Chapter 7). In particular, there is no stopping criterion. Moreover, we are faced to considerable numerical problems. Due to rounding errors annealing tends to mutate into a greedy algorithm, a problem which is not well understood. Finite time annealing - i.e annealing with a bounded number of steps - usually leads to an optimization problem much harder to the original one (cf. [13]).

Hence each *exact* algorithm is a useful tool to infer models on the one hand and to study convergence of relaxation on the other. Only few exact algorithms for Bayesian image analysis are known; examples are the Ford-Fulkerson approach to binary data in [10] or the GNC-algorithm to solve (10) for the Gaußian case $D(x,y) = \sum_s (y_s - x_s)^2$ (and the special functions ψ) in [4]. Both algorithms are restricted to these very special situations and as far as it is known cannot be generalized.

Below we present an extremely fast exact algorithm for the computation of MAP-estimates for one-dimensional signals based on dynamic programming. It works for the Potts model and any kind of noise for which an explicit estimator of the mean can be computed like the empirical mean if noise is i.i.d. Gaußian or the median for i.i.d. double exponential noise. Implementation for the general model (10) in one dimension is work in progress as well as the extension to the 2-d case. We also keep track of parameters.

Let us first introduce the Potts prior. For $\lambda = \infty$ or $\psi(u) = \gamma(1 - \mathbf{1}_{\{0\}}(u))$ the energy (10) boils down to a Potts model

$$H(x) = \sum_{s \sim t} \psi(x_s - x_t) + D(x,y) = \gamma \cdot |\{s \sim t : x_s \neq x_t\}| + D(x,y) \quad (20)$$

where $|A|$ is the cardinality of the set A. The prior term simply counts neighbours with different intensities.

3.1 Exact *MAP* for the 1-d Potts Prior

We now impose some severe restrictions:

(1) S is a one-dimensional lattice $S = \{1, \ldots, N\}$ with a nearest neighbour structure.

(2) Noise is i.i.d. and hence $D(x,y) = \sum_{s \in S} \rho(x_s - y_s)$ with some function ρ.

A signal x is completely determined by a partition $\mathcal{P}_x$ of S into discrete intervals $I \in \mathcal{P}_x$ and the constant intensities $\mu_{\mathcal{P}} = (\mu_I : I \in \mathcal{P})$ on the

intervals and, conversely, each $(\mathcal{P}, \mu_{\mathcal{P}})$ determines a unique x. Hence we may write $H(x) = H_{\mathcal{P}}(\mu_{\mathcal{P}})$ if convenient. Then (20) boils down to

$$H(x) = H_{\mathcal{P}_x}(\mu_{\mathcal{P}_x}) = \gamma(|\mathcal{P}_x| - 1) + \sum_{I \in \mathcal{P}_x} \sum_{s \in I} \rho(y_s - \mu_I).$$

Given a partition $\mathcal{P}$, the minimization problem reduces to the minimization of each single term

$$H_I(\mu_I) = \gamma + \sum_{s \in I} \rho(y_s - \mu_I).$$

For many functions ρ the minimizers μ_I^* are known. If, for instance, noise is Gaußian and thus $\rho(u) = u^2$ then the means $\mu_I^* = (\sum_{s \in I} y_s)/(|I|)$ will be plugged in; if it is double-exponential and $\rho(u) = |u|$ then the mean is replaced by the median of $\{y_s : s \in I\}$ and so on. One has

$$\min_x H(x) = \min_{\mathcal{P}, \mu_{\mathcal{P}}} H_{\mathcal{P}}(\mu_{\mathcal{P}}). \tag{21}$$

A minimizer of this function minimizes some $H_{\mathcal{P}}$. This observation reduces the minimization problem (21) to the discrete problem

$$\text{minimize } \mathcal{P} \longmapsto H_{\mathcal{P}}(\mu_{\mathcal{P}}^*). \tag{22}$$

Here we incorporate dynamic programming: Define

$$J(n) = \min_{\mathcal{P} \in \mathfrak{P}(N)} H_{\mathcal{P}}(x_{\mathcal{P}}^*)$$

where $\mathfrak{P}(n)$ denotes the set of all partitions of $\{\, 1, \ldots, n \,\}$. Since

$$\begin{aligned} H_{\{\, I_1, \ldots, I_k \,\}}(\mu_{\{\, I_1, \ldots, I_k \,\}}) &= -\gamma + \sum_l H_{I_l}(\mu_{I_l}) \\ = \quad H_{\{\, I_1, \ldots, I_{k-1} \,\}}(\mu_{\{\, I_1, \ldots, I_{k-1} \,\}}) &+ H_{I_k}(\mu_{I_k}) \end{aligned}$$

for all $N \geq n > 1$ the following holds

$$J(n) = \min_{1 \leq r \leq n-1} (J(r) + \min_{\mu \in \mathbb{R}} H_{[r+1,n]}(\mu)). \tag{23}$$

In other words, J is a Bellmann function.

Now we can establish the algorithm for the minimization of (21). It runs as follows:

(1) For all $1 \leq r < s \leq N$ determine $\mu_{r,s}^* = \operatorname{argmin}_{\mu \in \mathbb{R}} H_{[r,s]}(\mu)$ and $H_{[r,s]}(\mu_{[r,s]}^*)$.

(2) Set $J(1) = 0$.

(3) Determine $J(n)$ for all $1 < n \leq N$ in (23), keeping track of (at least one) r_n^* giving the minimum in (23).

(4) Determine recursively from minimizers r^* for $J(N)$, $J(r_N^*)$, ... a partition $\{ I_1, \ldots, I_k \}$ of $\{ 1, \ldots, N \}$. Then x^* is determined by $x_s^* = \mu_{I_l}^*$ for $s \in I_l$.

By a suitable arrangement, the complexity is $O(N^2)$ for (1), $O(1)$ for (2), $O(N^2)$ for (3) and $O(N)$ for (4). Thus the whole algorithm works in $O(N^2)$ complexity. This is of the same order of the generic complexity of nonlinear filters.

Up to now the parameter γ was fixed. Like for the filter chain (26) there remains the crucial problem to determine the best parameter. Of course, this value depends on the model behind the data y and the quality function on the set of approximations.

We adopt a completely different approach. Depending on the value of γ the algorithm determines piecewise constant approximations to the data. If $\gamma \downarrow 0$, data is recovered. If $\gamma \to \infty$, the optimal vector x^* becomes a constant signal. In the range between γ controls the degree of smoothness.

Therefore we should compute the minimizers of (22) for *all* values of γ. Reformulation of the above scheme in terms of $H_I^0(\mu) = H_I(\mu) - \gamma$ and $H_{\mathcal{P}}^0 = H_{\mathcal{P}} - \gamma(|\mathcal{P}| - 1)$ results in

$$\tilde{J}(k, n) = \min_{\mathcal{P} \in \mathfrak{P}(n), |\mathcal{P}| = k} H_{\mathcal{P}}^0(\mu_{\mathcal{P}}^*).$$

Again, for $N \geq n \geq k > 1$ we find a recurrence relation

$$\tilde{J}(k, n) = \min_{1 \leq r \leq n-1} (\tilde{J}(k-1, r) + \min_{\mu \in \mathbb{R}} H_{[r+1,n]}^0(\mu)). \tag{24}$$

Because of

$$\min_x H(x) = \min_{1 \leq k \leq N} (\gamma(k-1) + \tilde{J}(k, N))$$

the global minimum is a continuous piecewise linear function in γ. Since $k \mapsto \tilde{J}(k, N)$ is increasing one finds the points of discontinuity of its first derivative in $O(N)$.

In summary, we adopt the following algorithm

(1) For all $1 \leq r < s \leq N$ determine $\mu_{[r,s]}^* = \operatorname{argmin}_{\mu \in \mathbb{R}} H_{[r,s]}^0(\mu)$ and $H_{[r,s]}^0(\mu_{[r,s]}^*)$.

(2) Set $\tilde{J}(1, n) = H_{[1,n]}^0(\mu_{[1,n]}^*)$.

(3) Determine $\tilde{J}(k,n)$ for all $1 < k \leq n \leq N$ from (24), keeping track of (at least one) $r^*_{k,n}$ giving the minimum in (24).

(4) Determined recursively from the minimizers r^* for $\tilde{J}(k,N)$, $\tilde{J}(k-1, r^*_{k,N})$, ... a partition $\{ I^k_1, \ldots, I^k_p \}$ of $S = \{ 1, \ldots, N \}$. $x^{k,*}$ is determined by $x^{k,*}_s = \mu^*_{I^k_l}$ for $s \in I^k_l$.

(5) Construct the piecewise linear function $\gamma \mapsto \min_x H(x) = \min_k H(x^{k,*})$.

Complexity changes because of step (3) to $O(N^3)$. On the other hand, this gives us now the solution for any value of γ.

Fig. 4 displays segmentation of a noisy phantom by exact *MAP*-segmentation with the Potts model and different parameters γ. Some snapshots are cut out of the movie with decreasing γ. Observe that the number of jumps - and hence the segmentation stay constant over large γ-intervals.

Example 3.1 We started to apply this program to data from human brain mapping assessed by functional magnetic resonance imaging (*fMRI*). The observed time series represents a response in one voxel of the visual cortex to an outer boxcar-shaped visual stimulus [1]. The task is to decide whether there is a response in the voxel or not. The above algorithm transforms data into a 'segmentation'; in particular it gives a series of jumps. These may be used as a decision criterion. Such an approach should work with minimal prior statistical hypothesis. The only relevant features of a signal considered are the jumps. This is work in progress.

Such a time series and its segmentation by this method is displayed in Fig. (5). For simplicity, noise was assumed to be Gaußian. For $N = 69$ the C-implementation on a 133 MHz Pentium PC ran in much less than a second. For visualization we used the language IDL.

3.2 Generalizations

If λ in the definition of ψ is finite then the problem is more involved. For Gaußian noise, i.e. $\rho(u) = u^2$, basically the same procedure as above applies; we simply replace H_I by

$$H^\varphi_I(x_I) = \gamma + \sum_{s \sim t \in I} \lambda^2 (x_s - x_t)^2 + \sum_{s \in I} (x_s - y_s)^2.$$

Now the minimizers of H_I are not constant any more. Thus there is the additional problem to minimize H^φ_I. Let $\langle \cdot, \cdot \rangle$ denote Euclidean inner product. It is easy to see that

$$H^\varphi_I(x_I) = \gamma + \langle x_I, (\lambda^2 \Sigma + I) x_I \rangle - 2\langle x_I, y_I \rangle + \langle y_I, y_I \rangle \qquad (25)$$

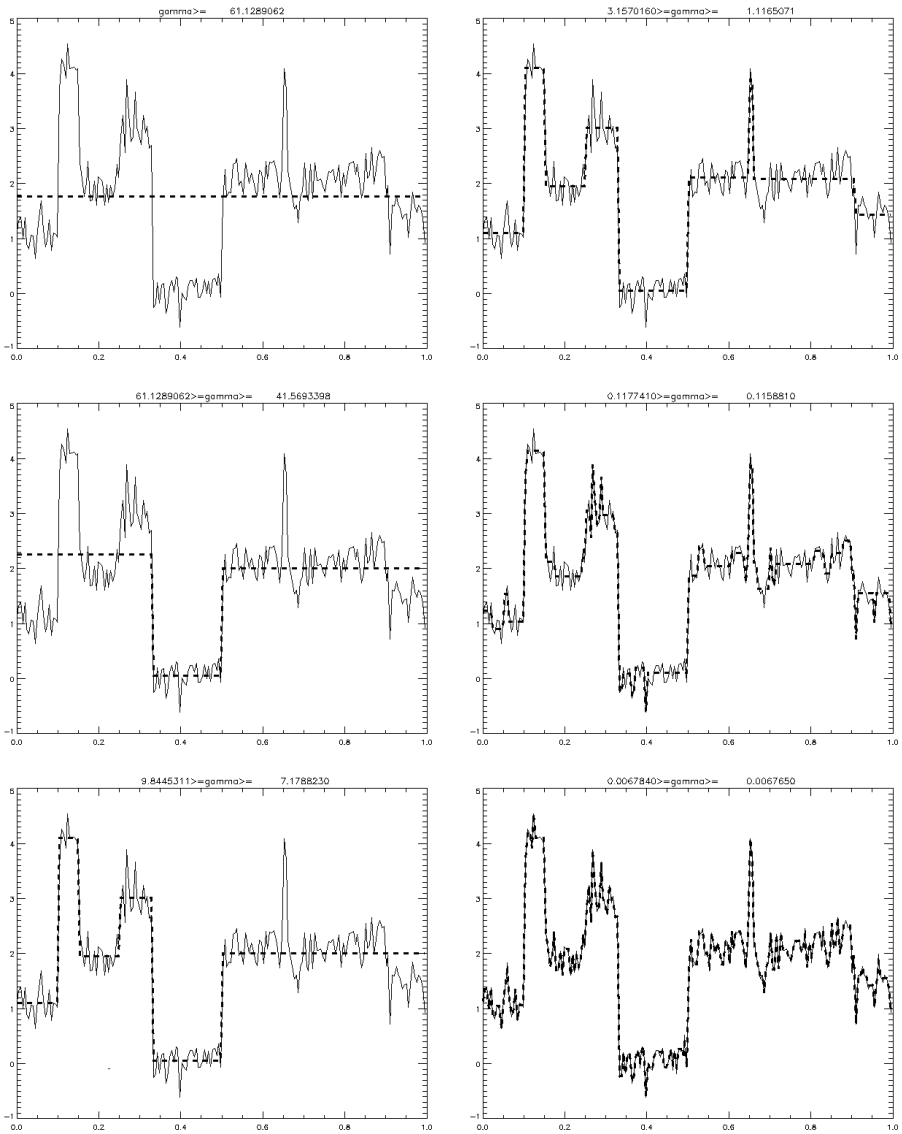


Figure 4: Segmentation of a noisy phantom by exact *MAP*-segmentation with the Potts model (to be viewed from top left to bottom right) for parameters $\gamma \in [61.1, \infty)$, $\gamma \in [41.6, 61.1]$, $\gamma \in [7.18, 9.84]$, $\gamma \in [1.12, 3.16]$, $\gamma \in [0.116, 0.118]$, $\gamma \in [0.00677, 0.00678]$.

where Σ is the $|I| \times |I|$ matrix

$$\Sigma = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & 0 & \\ 0 & & \ddots & & 0 \\ & 0 & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}.$$

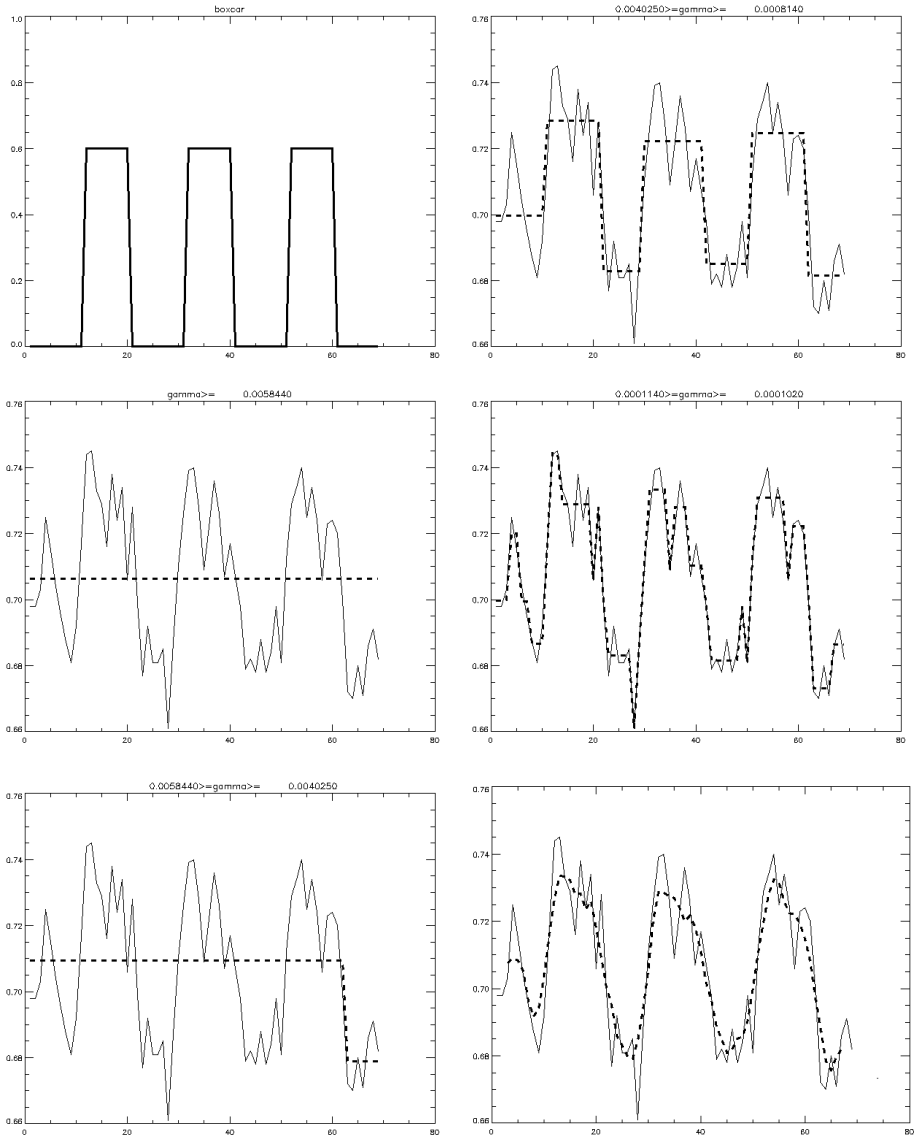


Figure 5: Steps in γ-scanning. Left column: boxcar signal, first step of reconstruction ($\gamma \geq 0.0058$) and data, second step of reconstruction ($\gamma \in [0.0040, 0.0058]$). Right column: third step of reconstruction ($\gamma \in [0.0008, 0.0040]$, 20th step of reconstruction ($\gamma \in [0.00011, 0.00010]$, an output of a radial-basis-function network, [12]. Solid line: (response) signal; dotted line: segmentation.

For the quadratic minimization problem $H_I^{\varphi}(x_I) \mapsto \min$ differentiation of (25) with respect to x_I shows that the unique solution is given by

$$x_I^* = (\lambda^2 \Sigma + Id)^{-1} y_I.$$

To compute $(\lambda^2\Sigma+\mathrm{Id})^{-1}$ it may help that the eigenvalues and eigenvectors of the Σ are well-known, see e.g. [17]. In contrast to the above scheme $H_I^\varphi(x_I^*)$ now depends nonlinearly on λ^2; we have

$$\begin{aligned} H_I^\varphi(x_I^*) &= \gamma + \langle y_I, (\mathrm{Id} - (\lambda^2\Sigma + \mathrm{Id})^{-1}) y_I \rangle \\ &= \gamma + \sum_{k=1}^{|I|} \frac{\lambda^2\lambda_k}{\lambda^2\lambda_k + 1} \langle y_I, \mathrm{Pr}_k y_I \rangle \end{aligned}$$

where λ_k is the k^{th} eigenvalue of Σ and Pr_k is its kth eigen-projection. Due to this nonlinearity in λ^2 it is somewhat harder to implement the scanning of minimizers as a function of both λ^2 and γ. Nevertheless, for constant λ the algorithm has the same computational complexity (quadratic respectively cubic) as computed above.

We conclusion that this dynamic programming approach is flexible and applies to a wide variety of functions ψ and ρ, mainly with modifications in the computation of x_I^*. We conjecture that suitable algorithms should exist for all smooth convex ρ and all $\psi(u) = \min\{\varphi, \gamma\}$ with smooth convex φ.

4 Sigma-Filters and Chains of Sigma-Filters

We now introduce a chain of σ-filters with varying scale parameters (cf. [2], [3], [20], [29]). It is given by

$$\mathcal{F}_{\sigma_n,\tau_n} \circ \ldots \circ \mathcal{F}_{\sigma_1,\tau_1}, \tag{26}$$

where each $\mathcal{F}_{\sigma_k,\tau_k}$ is a nonlinear Gaußian filter with weights w_{σ_k} and v_{τ_k}. It is an edge preserving segmentation and smoothing algorithm which even is able to sharpen blurred edges without any displacement. It first approximates data by (nearly) piecewise constant functions thus providing a segmentation into smooth parts. This is then used as the basis of smoothing in a subsequent processing step. The chain is based on NLG-Filters and thus each filter step reqires only two parameters. Furthermore, only few steps are necessary in practice and there are no practical problems with convergence. In this section we focus on the choice of chain parameters.

In passing we we comment on some aspects of σ-filters we did not meet in the literature.

4.1 Some Basic Properties of Sigma-Filters

An important property of σ-filters is edge preservation. Closely connected is the way it transforms blurred jumps, i.e. ramps and slopes. We give some

elementary arguments that σ-filters even are able to steepen slopes. We restrict ourselves to one dimensional signals. It is convenient to switch from the discrete to the continuous filter. It is given by

$$(\mathcal{F}y)(s) = \int w(y(t) - y(s))v(t-s)y(t)\,dt \Big/ \int w(y(t) - y(s))v(t-s)\,dt\ , \tag{27}$$

where it is assumed that v and w are symmetric around 0 and integration extends from $-\infty$ to ∞. For sake of completeness we assume $v, w \geq 0$, $v \in L^1$, $w \in L^\infty_{loc}$ and $y \in L^\infty$.

The jump-preserving property is best illustrated by application to a pure jump

$$z = C(-\mathbf{1}_{(-\infty,0)} + \mathbf{1}_{[0,\infty)})/2, \tag{28}$$

of height C. Let

$$\begin{aligned} a(s) &= \int_{-\infty}^{-s} v(t)\,dt = \int_{-\infty}^{0} v(t-s)\,dt \\ b(s) &= \int_{-s}^{\infty} v(t)\,dt = \int_{0}^{\infty} v(t-s)\,dt. \end{aligned}$$

One readily computes

$$(\mathcal{F}z)(s) = \frac{C}{2}\frac{-w(0)a(s) + w(C)b(s)}{(w(0)a(s) + w(C)b(s)}, \quad s < 0.$$

Since $a(s), b(s) \to \int v(t)\,dt/2$ as $s \to 0$ the left-hand limit of the output at the jump is

$$(\mathcal{F}z)(0-) = \frac{C}{2} \cdot \frac{w(C) - w(0)}{w(0) + w(C)},$$

and by symmetry the right-hand limit is

$$(\mathcal{F}z)(0+) = \frac{C}{2} \cdot \frac{w(0) - w(C)}{w(0) + w(C)}.$$

After filtering the jump has height

$$\Delta = (\mathcal{F}z)(0+) - (\mathcal{F}z)(0-)$$

and the proportion of jump heights is

$$\frac{\Delta}{C} = \frac{w(0) - w(C)}{w(0) + w(C)}.$$

From this identity one concludes that the output of the filter has a jump where the input has a jump; moreover, the proportion of jump heights can be derived from the shape of w.

Example 4.1 Obviously, one has $\mathcal{F}z = z$ for the pure jump whenever $w(C) = 0$. More generally, by such a filter a jump at 0 of height C is preserved for all signals which increase on the support of v.

All this holds in particular for truncated means $w = \mathbf{1}_{[-\sigma,\sigma]}$, $\sigma < C$. For such w the filter transforms a signal y like a linear filter if $\sigma > \sup_{s,t} |y(s) - y(t)|$. For the nonlinear Gaußian filter and input z the outputs are displayed for $\tau = 1$ and $\sigma = k \cdot 0.5$, $1 \le k \le 10$ in Fig. 6. It also shows a plot of Δ as a function of σ.

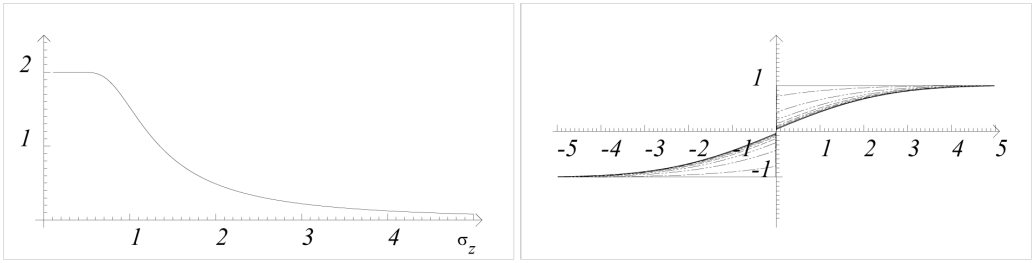


Figure 6: Jump height for input signal (28) filtered by NLGF as a function of σ and filter outputs for $\tau = 1$ and $\sigma = k \cdot 0.5$, $1 \le k \le 10$.

Another remarkable property of σ-filters is their ability to flatten and - even more important - to *steepen* slopes. Plainly, this ability strongly depends on the parameters. If σ is larger than the diameter of the range of intensities then the σ-filter tends to perform like a linear filter and blur edges. If - on the other hand - σ is small and τ is sufficiently large then ramp edges can be sharpened. Let us make these heuristic considerations more precise. For differentiable signals let us consider derivatives as measures of steepness. Then steepening or flattening an increasing slope means that the derivative is in- or decreased, respectively. The following general result provides an explicit formula. Let the (strictly positive) denominator in (27) be denoted by $D(s)$.

Proposition 4.2 *Let y be a continuously differentiable, bounded and odd function on the real line with bounded derivative y'. Let further v and w be continuously differentiable, strictly positive and even, assume that v is integrable and there are $\varepsilon > 0$ and an integrable function u such that $|v'(t - s)| \le u(t)$ for all $s \in (-\varepsilon, \varepsilon)$. Then*

$$D(s)\Big((\mathcal{F}y)' - y'(0)\Big) = \int \Big(y'(t) - y'(0)\Big) v(t) \Big(w(y(t) + w'(y(t))y(t)\Big)\, dt$$

The calculations (carried out in the appendix of [29]) are straightforward but somewhat tricky. For convenience of the reader we give a simpler proof in the appendix.

If we assume y' to be maximal at 0 then $y'(t) - y'(0) < 0$; moreover, $w(y) \geq 0$ and $yw'(y) \leq 0$. The latter holds if y is odd and increasing and w is bell-shaped. After a minute of reflection one concludes that according to the shape of v and w there is steepening or flattening at 0. This behaviour of the σ-filter was already claimed in [24] and discussed there in an informal way.

Example 4.3 This becomes more conspicuous in the case of Gaußian kernels

$$w_\sigma(u) = g(u/\sigma), \;\; v_\tau(t) = g(t/\tau)., \;\; g(t) = \exp(-t^2/2).$$

In this case the identity boils down to

$$D(s)\Big((\mathcal{F}y)'(0) - y'(0)\Big) = \sigma^2 \int \Big(y'(0) - y'(t)\Big) v_\tau(t) w_\sigma(y(t)) \Big((y(t)/\sigma)^2 - 1\Big)\, dt.$$

It is immediately clear that for $\sigma \geq \|y\|_\infty$ there is flattening. But for suitable signals, small σ and large enough τ the slope is steepened. This is illustrated in Fig. 7. The input signal is $y = 2\boldsymbol{G} - 1$ where $\boldsymbol{G}$ denotes the cumulative distribution function of the standard normal distribution; it is filtered by $NLGF$ with parameters $\sigma = 0.5$ and $\tau = 3$. One should keep in mind that $\boldsymbol{G}$ can be thought of as a jump function of type (28) blurred by a linear Gaußian filter with $\tau = 1$. The residual $(\mathcal{F}y)'(0) - y'(0)$ for fixed $\sigma = 0.5$ as

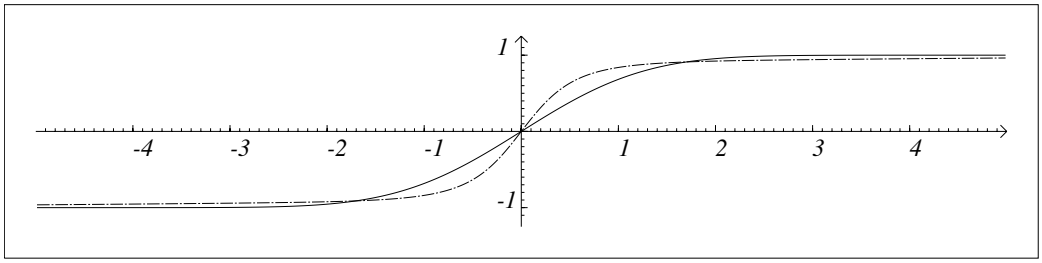


Figure 7: Sharpening of a blurred edge by $NLGF$ with $\sigma = 0.5$ and $\tau = 3$.

a function of τ is plotted in Fig. 8 .

Remark 4.4 The NLG filter is closely related to anisotropic diffusion $\partial u/\partial t = \operatorname{div}(h(u) \cdot \operatorname{grad})u$, cf. [28]. We shall not pursue this aspect here.

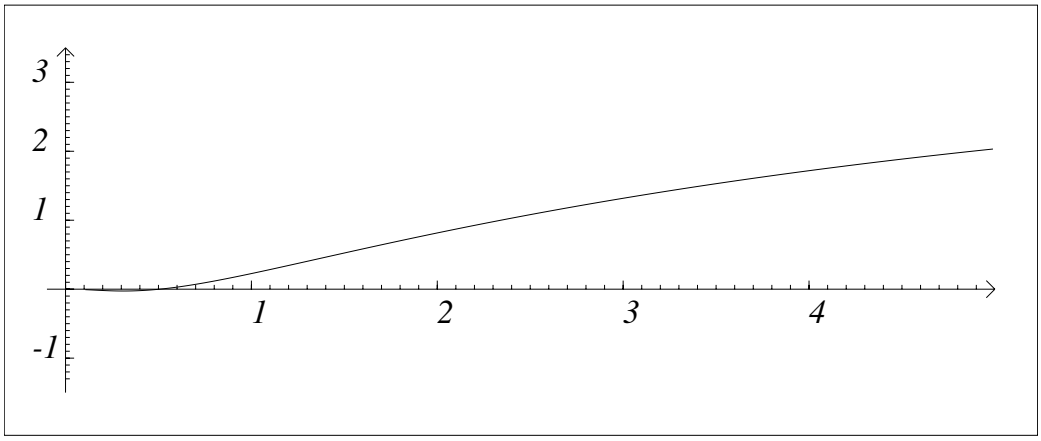


Figure 8: The function $\tau \mapsto \mathcal{F}^{0.5,\tau}(y)'(0) - y'(0)$ for $y = 2\boldsymbol{G} - 1$ with the standard normal c.d.f. $\boldsymbol{G}$.

4.2 Best Parameters for the *NLG* Filter Chain

The above observed parameter-dependent abilities of σ-filters to smooth and sharpen may be combined in a chain of such filters with different parameters in each step. Formally, such a chain is given by (26). The chain (26) gives a segmentation of the signal. After segmentation, the filter weights of the last step may be used in subsequent filtering of raw data to perform smoothing on the segments. Below we only discuss segmentation. For carefully chosen parameters performance is illustrated in Fig. 9. A two-dimensional example is displayed in Fig. 10, where a very dirty radio is cleaned by the chain.

It is difficult to analyze the exact performance of a nonlinear Gaußian filter chain because each filter stage mixes the input in a complicated way. Even if the input (Y_s) is white noise which means that the X_s are i.i.d. random variables with zero mean, the output variables $(\mathcal{F}_{\sigma_1,\tau_1} X)_s$ of the first filter step are correlated in a tortuous way. Hence it is very difficult to obtain rigorous results for the distribution of the outputs this and the following filter steps.

Nevertheless, on a heuristic level plausible arguments can be given for the parameter choice; at least for many practical application the derived strategy proved to be successful and, in fact could not be outperformed in any experiment. We basically follow the arguments of V. Aurich and E. Mühlhaus, cf. [20]. Outputs of nonlinear filters will frequently be compared to those of of linear ones; hence we introduce the *linear Gaußian filter*

$$(\mathcal{G}_\tau y)_s = \sum_{t\in S} v_\tau(t-s) y_t \Big/ \sum_{t\in S} v_\tau(t-s)\ .$$

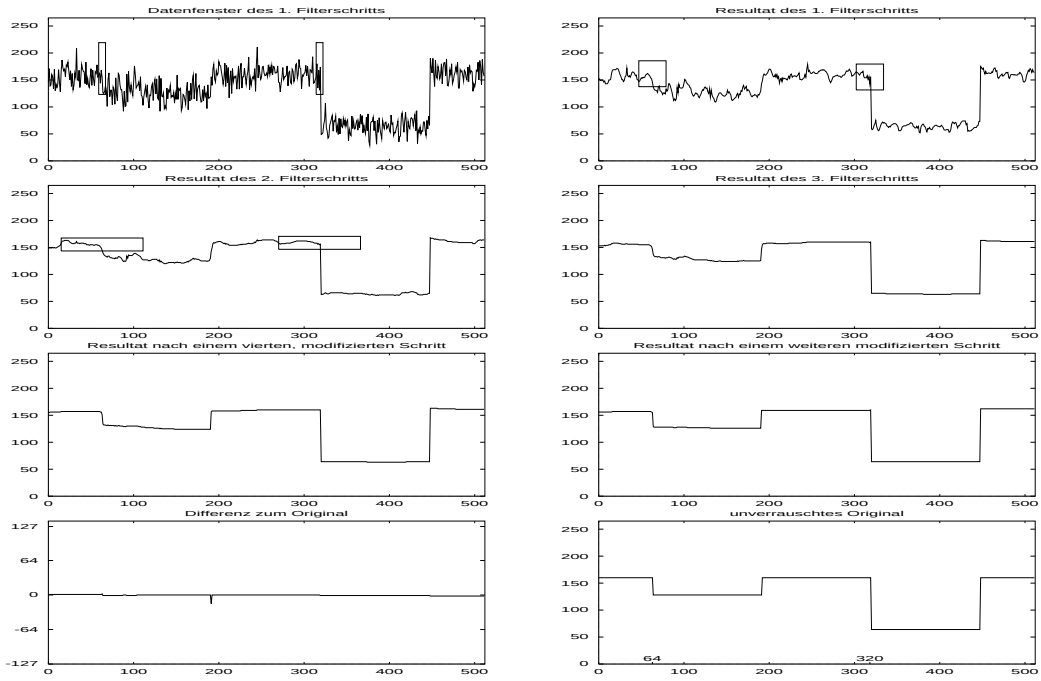


Figure 9: Performance of the NLG filter chain. From first to last row: Data and windows (indicated by boxes) of first step; output of first , second, third step; output of modified fourth, fifth step; residuals; original signal.

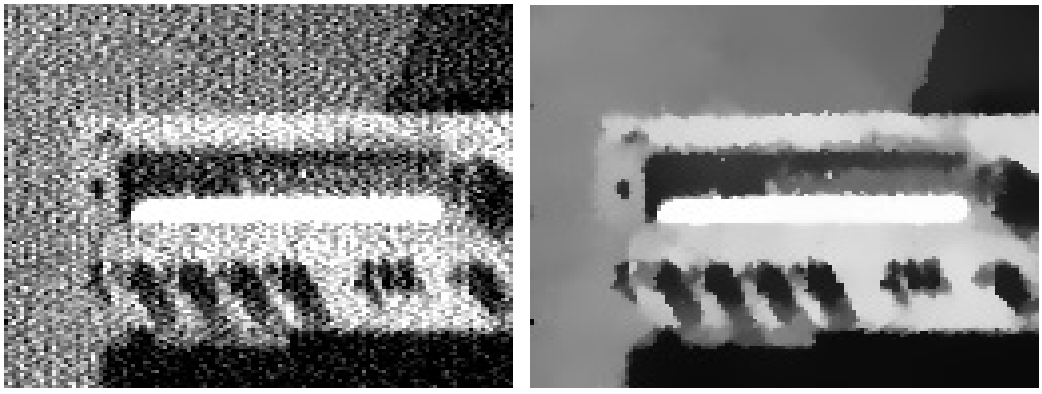


Figure 10: A dirty radio cleaned by the NLG filter chain

The following arguments are based on some simplifying heuristics. Throughout the rest of this section, $Y = (Y_t)$ will denote i.i.d. random variables with normal distribution $\mathcal{N}(\mu, \rho^2)$.

(A1) If $\sigma \geq 3\varrho$ then $(\mathcal{F}_{\sigma,\tau}Y)_s$ and $(\mathcal{G}_\tau Y)_s$ have similar distributions.

(A2) If Z is the output of a linear Gaußian filter G_η fed with Y and if $\sigma \geq 2(\mathrm{var} Z_s)^{1/2}$ then $(\mathcal{F}_{\sigma,\tau} Z)_s$ and $(\mathcal{G}_\tau Z)_s$ have similar distributions.

The Z_s are again Gaußian but dependent. The nearer s and t are to each other the more the joint distribution of Z_s and Z_t is concentrated near the diagonal. Therefore σ can be chosen smaller than in (A1).

(A3) The chain $\mathcal{G}_{\tau_j} \circ \cdots \circ \mathcal{G}_{\tau_1}$ of linear Gaußian filters can be replaced by the single Gaußian filter $\mathcal{G}_\tau$ with $\tau = (\sum_{i=1}^{j} \tau_i^2)^{1/2}$.

In the continuous case the exact equality $\mathcal{G}_{\tau_j} \circ \cdots \circ \mathcal{G}_{\tau_1} = \mathcal{G}_\tau$ holds and such an assumption is not critical.

(A4) The variance of $(\mathcal{G}_\tau Y)_s$ is about $\varrho^2/(2\pi^{1/2}\tau)^d$.

In fact, neglecting the discretization error one computes

$$\begin{aligned} \mathrm{var}\, \mathcal{G}_\tau Y_s &= \frac{1}{N^2} \sum_t g_\tau^2(||t||)\, \varrho^2 = \frac{1}{N^2} \sum_t g_{\tau/\sqrt{2}}(||t||)\, \varrho^2 \\ &\approx \varrho^2 \int g_{\tau/\sqrt{2}}(||t||)\, dt \Big(\int g_\tau(||t||)\, dt \Big)^{-2} \\ &= \varrho^2 \cdot (\pi^{1/2}\tau)^d / ((2\pi)^{1/2}\tau)^{2d} = \varrho^2/(2\pi^{1/2}\tau)^d. \end{aligned}$$

To ensure that $\mathcal{F}_{\sigma,\tau}$ smoothes white noise in a similar way as the linear Gaußian filter $\mathcal{G}_\tau$ does, the parameter σ has to be sufficiently large compared to noise. On the other hand, blur of a jump is negligible if σ is smaller than the height of the jump. Therefore we assume exponentially decreasing parameters with

$$\sigma_{j+1} = \sigma_j/\alpha = \sigma_1/\alpha^j$$

for some $\alpha > 1$. Given the noise variance ϱ^2, we watch out for parameters $\sigma_1, \tau_1, \ldots, \tau_n$ as small as possible on the one hand but on the other hand fulfilling the following property:

(P) The distributions of $\mathcal{F}_{\sigma_j,\tau_j} \circ \ldots \circ \mathcal{F}_{\sigma_1,\tau_1} Y$ and of $\mathcal{G}_{\tau_j} \circ \ldots \circ \mathcal{G}_{\tau_1} Y$ are close to each other.

We proceed by induction:
Set $\sigma_1 = 3\varrho$. According to (A1) $\mathcal{F}_{\sigma_1,\tau_1} Y$ and $\mathcal{G}_{\tau_1} Y$ have similar distributions. The parameter τ_1 has to be chosen such that (A2) applies to the second filter step $\mathcal{F}_{\sigma_2,\tau_2}$. Therefore and by (A4)

$$\sigma_2 \geq 2(\mathrm{var}(\mathcal{G}_{\tau_1} Y)_s)^{1/2} \approx 2\varrho/(2\sqrt{\pi}\tau_1)^{d/2}$$

Because $\sigma_2 = \sigma_1/\alpha = 3\varrho/\alpha$ this implies

$$\tau_1 \geq (2\pi^{1/2})^{-1}(2\alpha/3)^{2/d} =: (2\pi^{1/2})^{-1}\mu.$$

To keep blur of jumps small we set $\tau_1 = \mu$. Then $\mathcal{F}_{\sigma_2,\tau_2} \circ \mathcal{F}_{\sigma_1,\tau_1} Y$ and $\mathcal{G}_{\tau_2} \circ \mathcal{G}_{\tau_1} Y$ have similar distributions.

For $j \geq 2$ we argue as follows. Suppose that $\sigma_1, \tau_1, \ldots, \tau_{j-1}$ are given such that $\mathcal{F}_{\sigma_{j-1},\tau_{j-1}} \circ \cdots \circ \mathcal{F}_{\sigma_1,\tau_1} Y$ and $\mathcal{G}_{\tau_{j-1}} \circ \cdots \circ \mathcal{G}_{\tau_1} Y$ have similar distributions. Because of (A3) $\mathcal{F}_{\sigma_{j-1},\tau_{j-1}} \circ \cdots \circ \mathcal{F}_{\sigma_1,\tau_1} Y$ and $\mathcal{G}_{\eta_i} Y$ with $\eta_i = (\sum_{k=1}^{i} \tau_k^2)^{1/2}$ have similar distributions.

Hence

$$\mathrm{var}(\mathcal{F}_{\sigma_i,\tau_i} \circ \ldots \circ \mathcal{F}_{\sigma_1,\tau_1} Y)_s \approx \mathrm{var}(\mathcal{G}_{\eta_i} Y)_s \approx \frac{\varrho^2}{(2\sqrt{\pi}\eta_i)^d}$$

by (A4). Using (A2), (A3), (A4) one obtains

$$\begin{aligned} \mathrm{var}(\mathcal{F}_{\sigma_{j-1},\tau_{j-1}} \circ \cdots \circ \mathcal{F}_{\sigma_1,\tau_1} Y)_s &\approx \mathrm{var}(\mathcal{G}_{\tau_{j-1}} \circ \cdots \circ \mathcal{G}_{\tau_1} Y)_s \\ &\approx \mathrm{var}(\mathcal{G}^{\eta_{j-1}} Y)_s \approx \frac{\varrho^2}{(2\sqrt{\pi}\eta_{j-1})^d}. \end{aligned}$$

By the same reasoning as above, we find

$$\sigma_j = \sigma_1/\alpha^{j-1} \geq 2\frac{\varrho}{(2\sqrt{\pi}\eta_{j-1})^{d/2}}$$

or

$$4\pi \sum_{k=1}^{j-1} \tau_k^2 \geq (2\alpha/3)^{4/d}.$$

Again, we should choose τ_{j-1} minimal. This yields

$$\tau_{j-1} = (2\sqrt{\pi})^{-1/2}(2\alpha^{j-1}/3)^{2/d}\sqrt{\mu^2 - 1}.$$

Thus

$$\frac{\tau_j}{\tau_{j-1}} = \begin{cases} (2\alpha^{j-1}/3)^{2/d} & \text{if } j > 2 \\ \sqrt{\mu^2 - 1} & \text{if } j = 2 \end{cases}.$$

For practical experiments we chose $\alpha = 2$ and used the following
Strategy: If ϱ^2 is the estimated noise variance then choose

$$\begin{aligned} \sigma_1 &= 3\varrho, \qquad \sigma_j = \frac{1}{2}\sigma_{j-1} \text{ for } j > 1, \\ \tau_1 &= \frac{1}{2\sqrt{\pi}}\sqrt[d]{\frac{16}{9}}, \qquad \tau_j = \sqrt[d]{4}\tau_{j-1} \text{ for } j > 1. \end{aligned}$$

Notice that $(\sqrt[d]{2^4}-1)^{1/2} \approx \sqrt[d]{4}$; hence we use for $j=2$ and $j>2$ the same recursion for σ.

Special cases:

$$\begin{aligned} d=1: &\qquad \tau_j = 4\tau_{j-1} \\ d=2: &\qquad \tau_j = 2\tau_{j-1} \\ d=3: &\qquad \tau_j = \sqrt[3]{4}\tau_{j-1} \approx 1,6 \cdot \tau_{j-1} \end{aligned}$$

In practice τ_1 is chosen between 0.5 and 1.

Numerous experiments in dimensions $1, 2$ and 3 have been performed. The above strategy worked very well if noise was more or less white and bell-shaped distributed. The choice of the parameters σ_1 and τ_1 is usually not critical; small changes have only little influence on the filter result. Implementations which are reasonably fast can be downloaded from `http://www.cs.uni-duesseldorf.de/aurich/nlg`.

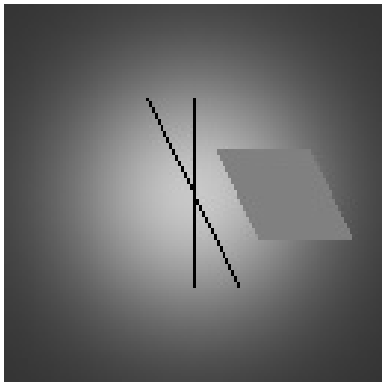
Although the above strategy works quite well it is not satisfying from a theoretical point of view because the decay of the σ_j is fixed in advance. A more flexible choice can possibly diminish the blurring signal jumps without spoiling the noise reduction. For this purpose [Mühlhaus, 47] introduces the notion of total blur of a filter chain and defines a filter chain as optimal if its total blur is minimal. The application of this notion in practice suffers from the fact that the minimization problem is not solved explicitly.

4.3 Non-Horizontal Gaußian Filters

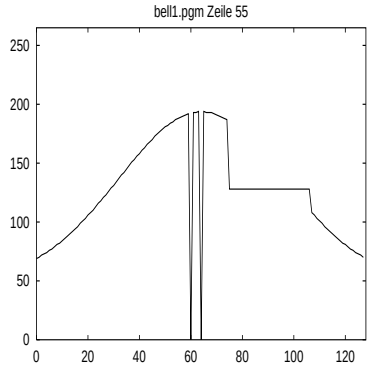
Nonlinear Gaußian filter chains tend to break up ramp-like input signals into steps since they give a kind of segmentation. The reason is that the weight term locally penalizes deviations from constants: it is based on first discrete derivatives. Second discrete derivatives would be a straightforward generalization. Since the input data are noisy we avoid them here and instead plug in the output of a *linear* Gaußian filter. Thus a non-horizontal nonlinear Gaußian filter is defined by

$$\begin{aligned} (\mathcal{H}_{\pi,\sigma,\eta} y)_s &= \frac{1}{N_s} \sum_t g_\tau(||t-s||) g_\sigma(y_t - (\mathcal{G}_\eta y)_t) y_t, \\ N_s &= \sum_t g_\tau(||t-s||) g_\sigma(X_t - (\mathcal{G}_\eta X)_t). \end{aligned}$$

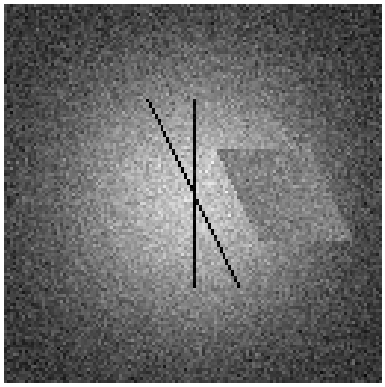
Chains of such filters can eliminate noise without destroying ramps or jumps. We mention this without any further discussion. Performance is illustrated in Fig. 11 where the noisy image is filtered by a non-horizontal Gaußian filter chain with 4 steps.



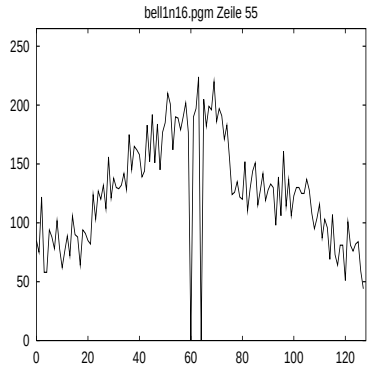
original

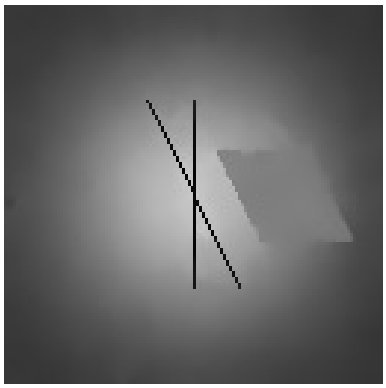


line 55



noise added





filter result

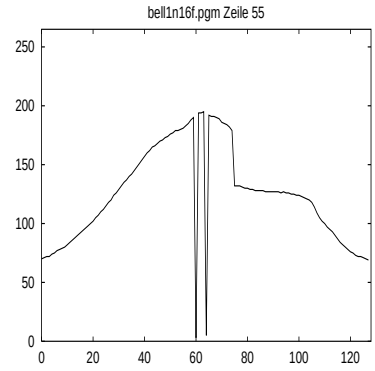


Figure 11: Image data with ramps and edges corrupted by Gaußian noise and processed by the non-horizontal nonlinear Gaußian filter chain

These images can be downloaded from
`http://www.cs.uni-duesseldorf.de/aurich/testimages`. The reader is invited to compare with the performance of other filters.

5 Appendix

In this appendix we reprove a formula from [29], p. 112-113. It is used to explain sharpening of blurred edges in Section 4.1.

Proposition 5.1 *Let y be a continuously differentiable, bounded and odd function on the real line with bounded derivative y'. Let further v and w be continuously differentiable, strictly positive and even, assume that v is integrable and there are $\varepsilon > 0$ and an integrable function u such that $|v'(t-s)| \leq u(t)$ for all $s \in (-\varepsilon, \varepsilon)$. Then*

$$(\mathcal{F}y)' - y'(0) \quad = \quad \frac{\int \Big(y'(t) - y'(0)\Big) v(t) \Big(w(y(t) + w'(y(t))y(t)\Big) dt}{\int w(y(t))v(t)dt}.$$

.

Proof. We have to differentiate the function

$$s \longmapsto (\mathcal{F}y)(s) = \frac{\int (w(y(t) - y(s))v(t-s)y(t)\, dt}{\int w(y(t) - y(s))v(t-s)\, dt} =: \frac{D(y)(s)}{N(y)(s)}.$$

Denominator and numerator $D(y)$ and $N(y)$ are continuously differentiable functions, $N(y)$ is strictly positive and $D(y)(0) = 0$ which implies

$$(\mathcal{F}y)'(0) = \frac{D'(y)(0)N(y)(0) + D(y)(0)N'(y)(0)}{N^2(y)(0)} = \frac{D'(y)(0)}{N(y)(0)}.$$

Interchange of differentiation w.r.t. s and integration w.r.t. t yields

$$\begin{aligned} D'(y)(s) &= \int \frac{d}{ds}(w(y(t) - y(s))v(t-s)y(t))dt \\ = &\int -\frac{d}{dt}(w(y(t) - y(s))v(t-s)y(t))\, dt \\ + &\int v(t-s)\Big\{w\Big(y(t) - y(s)\Big)y'(t) \ + w'\Big(y(t) - y(s)\Big)y(t)\Big(y'(t) - y'(s)\Big)\Big\}\, dt. \end{aligned}$$

The first term vanishes and a rearrangement of terms gives the desired identity.

The last remark concerns box-shaped kernels.

Remark 5.2 Let y be as above, y' strictly positive but for v and w we choose box-functions. Define $v = 1_{[a,b)}$ and $w = 1_{[\alpha,\beta)}$ and assume that $N(y)(0) = \int w(y(t))v(t)dt > 0$. Then we get for s close to 0

$$(\mathcal{F}y)(s) = \int_{y^{-1}(y(s)+\alpha)\vee s+a}^{y^{-1}(y(s)+\beta)\wedge s+b} y(t)\,dt \Big/ N(y)(s)$$

and by Leibniz' rule

$$(\mathcal{F}y)'(0) * N(y)(0) = y(b)\mathbf{1}_{\{y(b)<\beta\}} + \frac{\beta y'(0)}{y'(y^{-1}(\beta))}\mathbf{1}_{\{\beta<y(b)\}}$$
$$-y(a)\mathbf{1}_{\{y(a)>\alpha\}} - \frac{\alpha y'(0)}{y'(y^{-1}(\alpha))}\mathbf{1}_{\{\alpha>y(a)\}}.$$

If $y(b) = \beta$ or $y(a) = \alpha$ the function $(\mathcal{F}y)$ is not differentiable at $s = 0$.

Acknowledgement: We thank A. Martin for generalizing the formula in the appendix and smoothing some arguments. K. Hahn and K. Rodenacker provided technical support and background from applications.

References

[1] D. Auer. fMRI-Data. Personal Communication, 1999.

[2] V. Aurich, E. Mühlhaus, and S. Grundmann. Kantenerhaltende Glättung von Volumendaten bei sehr geringem Signal-Rausch-Verhältnis. In *Zweiter Aachener Workshop über Bildverarbeitung in der Medizin*, 1998.

[3] V. Aurich and J. Weule. Non-linear gaussian filters performing edge preserving diffusion. In *Proceed. 17. DAGM-Symposium, Bielefeld*, pages 538–545. Springer, 1995.

[4] A. Blake and A. Zisserman. *Visual Reconstruction.* The MIT Press Series in Artificial Intelligence. MIT Press, Massachusetts, USA, 1987.

[5] C.K. Chu, I. Glad, F. Godtliebsen, and J.S. Marron. Edge-preserving smoothers for image processing. *JASA*, 93(442):526–541, 1998.

[6] C.K. Chu and J.S. Marron. Choosing a kernel regression estimator. *Statistical Science*, 6:404–436, 1991.

[7] M.M. Fleck. Plus ça change, ... In *Proc. Second European Conference on Computer Vision*, pages 151–159, 1992.

[8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEE Trans. PAMI*, 6:721–741, 1984.

[9] F. Godtliebsen, E. Spjøtvoll, and J.S. Marron. A nonlinear Gaussian filter applied to images with discontinuities. *J. Nonparametr. Statist.*, 8:21–43, 1997.

[10] D.M. Greig, B.T. Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, 51:271–279, 1989.

[11] X. Guyon. *Random Fields on a Network. Modelling, Statistics, and Applications*. Probability and its Applications. Springer Verlag, New York, Berlin, Heidelberg, 1995.

[12] K. Hahn and Th. Waschulzik. On the use of local RBF networks to approximate multivalued functions and relations. In L. Niklasson, M. Bodén, and T. Ziemke, editors, *Proceedings of the 8.th International Conference on Artificial Neural Networks ICANN, Skövde, Sweden, 2-4 September 1998*, volume 2, pages 505–510, Sweden, 1998. University of Skövde, Springer Verlag.

[13] B. Hajek and G. Sasaki. Simulated annealing - to cool or not. *Systems and Control Letters*, 12:443–447, 1889.

[14] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. Wiley & Sons, New York, 1986.

[15] Huber P. J. *Robust Statistics*. Wiley Series in Probability and Mathematical Statistics. Wiley & Sons, New York, 1981.

[16] B.I. Justusson. Median filtering: Statistical properties. In *Two-Dimensional Digital Signal Processing II. Transforms and Median Filters*, volume 43 of *Topics in Applied Physics*, chapter 5, pages 161–196. Springer Verlag, Berlin and Heidelberg and New York, 1981.

[17] H.R. Künsch. Robust priors for smoothing and image restoration. *Ann. Inst. Statist. Math.*, 46(1):1–19, 1994.

[18] J.S. Lee. Digital image smoothing and the sigma-filter. *Computer Vision, Graphics and Image Processing*, 24:255–269, 1983.

[19] V. Liebscher and G. Winkler. A Potts model for segmentation and jump-detection. Technical Report 99-7, Institute of Biomathematics and Biometrics, GSF-National Research Center for Environment and Health, Neuherberg/München, Germany, February 1999. To appear in: Proceedings of the S4G, Prague.

[20] E. Mühlhaus. *Die sprungerhaltende Glättung verrauschter, harmonischer Schwingungen.* PhD thesis, Heinrich-Heine-Universität Düsseldorf, 1998.

[21] J. Polzehl and V.G. Spokoiny. Adaptive image denoising with applications to MRI. May 1998.

[22] J. Polzehl and V.G. Spokoiny. Adaptive weights smoothing with applications to image segmentation. Preprint 405, Weierstraß-Institut für angewandte Analysis und Stochastik, Berlin, April 1998.

[23] J. Polzehl and V.G. Spokoiny. Image Denoising: Pointwise Adaptive Approach. Discussion Paper 38, Humbold-Universität zu Berlin, Sonderforschungsbereich 373: Quantifikation und Simulation ökonomischer Prozesse, Berlin, 1998.

[24] P. Saint-Marc, J.-S. Chen, and S.M. Piter. Adaptive smoothing: A general tool for early vision. *IEEE Trans. PAMI*, 8(2):147–163, 1986.

[25] D.G. Simpson, X. He, and Y.-T. Liu. Comment on: C.K. Chu and I. Glad and F. Godtliebsen and J.S. Marron, edge-preserving smoothers for image processing. *JASA*, 93(442):544–548, 1998.

[26] R.H. Swendsen and J.-S.Wang. Nonuniversal critical dynamics in Monte Carlo simulations. *Phys.Rev.Lett.*, 58:86–88, 1987.

[27] S.G. Tyan. Median filtering: Deterministic properties. In *Two-Dimensional Digital Signal Processing II. Transforms and Median Filters*, volume 43 of *Topics in Applied Physics*, chapter 6, pages 197–218. Springer Verlag, Berlin and Heidelberg and New York, 1981.

[28] J. Weickert. *Anisotropic Diffusion in Image Processing.* B.G. Teubner, Stuttgart, 1998.

[29] J. Weule. *Iteration nichtlinearer Gauß-Filter in der Bildverarbeitung.* PhD thesis, Heinrich-Heine-Universität Düsseldorf, 1994.

[30] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*, volume 27 of *Applications of Mathematics*. Springer Verlag, Berlin, Heidelberg, New York, 1995.

[31] G. Winkler, V. Aurich, K. Hahn, A. Martin, and K. Rodenacker. Noise reduction in images: Some recent edge-preserving methods. Technical Report 98-15, Institute of Biomathematics and Biometrics, GSF-National Research Center for Environment and Health, Neuherberg/München, Germany, December 1998. To appear in: Pattern Recognition and Image Analysis (1999).